Interoperable Simulation Gaming for Strategic Infrastructure Systems Design

by
Paul Thomas Grogan
B.S., University of Wisconsin–Madison (2008)
S.M., Massachusetts Institute of Technology (2010)

Submitted to the Engineering Systems Division
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Engineering Systems
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2014

Author . . . .
Engineering Systems Division
May 8, 2014

Certified by . . . .
Olivier L. de Weck
Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Supervisor

Certified by . . . .
Daniel D. Frey
Professor of Mechanical Engineering and Engineering Systems
Committee Member

Certified by . . . .
Donna H. Rhodes
Senior Lecturer in Engineering Systems
Committee Member

Certified by . . . .
Joseph M. Sussman
J.R. East Professor of Civil and Environmental Engineering and Engineering Systems
Committee Member

Certified by . . . .
John R. Williams
Professor of Civil and Environmental Engineering and Engineering Systems
Committee Member

Accepted by . . . .
Richard C. Larson
Mitsui Professor of Engineering Systems
Chair, Engineering Systems Division Education Committee

This page is intentionally left blank.

Interoperable Simulation Gaming for Strategic Infrastructure Systems Design

by

Paul Thomas Grogan

Submitted to the Engineering Systems Division
on May 8, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Engineering Systems

Abstract

Infrastructure systems are large physical networks of interrelated components which produce and distribute resources to meet societal needs. Meeting future sustainability objectives may require more complex systems with stronger integration across sectors and improved collaboration among constituent organizations. This dissertation introduces and demonstrates a method of interoperable simulation gaming to combine elements from concurrent engineering, wargaming, and serious gaming to support strategic design activities. First, a controlled human subjects experiment quantifies the relative impacts of technical and social sources of complexity using a simple surrogate design task, finding collaboration with communication barriers greatly increases the time and cost of design. Next, a modeling framework identifies common graph-theoretic structures and formal behavior definitions believed to be generalizable to all infrastructure systems. An interoperability interface defines interactions between system models to enable resource exchanges. Next, the High Level Architecture (HLA) standard is applied to the modeling framework to enable distributed, time-synchronized simulation with decentralized authority over constituent system models. A federation object model and agreement define data structures and processes to participate in a federated simulation execution. A prototype application case implements the modeling framework and simulation architecture using the context of Saudi Arabia. Infrastructure system models are developed for agriculture, water, petroleum, and electricity sectors. A baseline scenario develops system and element instantiations using historical estimates of resource flows and fictional costs. A software implementation provides a graphical user interface to modify design scenarios and visualize outcomes. Finally, a game formulation uses the prototype model as the basis of a simulation game with individual and collective objectives among water, energy, and agriculture ministry roles. Players collaboratively propose new infrastructure projects over a planning horizon to maximize objectives within time and budgetary constraints. A second controlled human subjects experiment studies the effect of three tool variants on outcome design quality, finding the number of data exchanges is positively correlated with outcome quality and an integrated simulation variant using the HLA results in more data exchanges compared to an asynchronous file-based variant.

Thesis Supervisor: Olivier L. de Weck
Title: Professor of Aeronautics and Astronautics and Engineering Systems

This page is intentionally left blank.

Acknowledgments

First and foremost, I thank my advisor Oli de Weck for his generous academic, professional, and personal support on a variety of stimulating projects throughout my time at MIT. He has been a constant source of inspiration and optimism and provided great flexibility in my pursuit of novel research topics. I also thank my committee members for their contributions. Dan Frey inspired the multi-actor design study and provided valuable advice on conducting experiments. Donna Rhodes contributed a wealth of experience in systems engineering theory and practice. Joe Sussman offered perspective on computer-supported design methods and helped anchor my abstract contributions. John Williams emphasized new software development methods for modeling and simulation.

This work was made possible with generous support from the U.S. Department of Defense through the National Defense Science and Engineering Graduate Fellowship (NDSEG) program. I also received support for necessary equipment from the Center for Complex Engineering Systems (CCES) at the King Abdulaziz City for Science and Technology in Saudi Arabia and the Massachusetts Institute of Technology under the Sustainable Infrastructure Planning System (SIPS) project. Several application cases are based on past participation in the SpaceNet project funded by the Strategic University Research Partnerships (SURP) program at the Jet Propulsion Laboratory and the City.Net project funded by the Masdar Institute for Science and Technology.

The Simulation Exploration Experience (formerly Simulation Smackdown) hosted by SISO and SCS introduced me to the High Level Architecture with technical guidance provided by Zack Crues and Dan Dexter. Gwen Kolfschoten hosted me at the TU Delft Faculty of Technology Management and Policy in June 2012 where I was introduced to collaborative engineering and met several faculty members who contributed to my understanding and use of gaming methods. At MIT I participated in several research communities including the Strategic Engineering Research Group (SERG), Systems Engineering Advancement Research Initiative (SEAri), and ESD Student Society (ESS), all providing valuable feedback and discussion. Several individuals deserve particular mention. Kaushik Sinha influenced my ideas through his work on complexity quantification. Sydney Do emphasized common features of Earth and space infrastructure as habitation systems. Network methods developed by Thomas Coffee and Takuto Ishimatsu influenced my description of infrastructure as graph edges. Ellen Czaika and Matt Fitzgerald helped explore literature on collaborative design methods. Roi Guinto assisted in developing, testing, and conducting experiments and I owe a debt of gratitude to the 70+ participants who volunteered their time as subjects. I also acknowledge my CCES colleagues who helped frame the Saudi Arabia application case including Ahmad Abdulkareem, Anas Alfaris, Adnan Alsaati, Jennifer Green, Vivek Sakhrani, Abel Sanchez, and Noelle Selin. Other colleagues providing key insights along the way include Jason Bartolomei, Adam Ross, Afreen Siddiqi, and Zoe Szajnfarber. There are many others deserving of mention for day-to-day conversations and feedback but who must forgive me for their omission here.

I thank those who managed the administrative side of research including Laura Bowen, Barb DeLaBarre, Julie Finn, and Beth Milnes. My student life greatly benefited from the camaraderie of my past roommates Allie Anderson, Aaron Johnson, and Greg O'Neill. I could not have completed this work without the nurturing environment from my friends and family. Natasha Lewis provided constant support and patience throughout my studies and I thank her family for considering me as one of their own. Finally, I owe everything to my parents Tom Grogan and Kathy Mulcahy for bringing me into this world and ensuring I had every possible opportunity for success.

Cambridge, Mass., May 2014. P. T. GROGAN.

This page is intentionally left blank.

Contents

This page is intentionally left blank.

List of Figures

List of Tables

This page is intentionally left blank.

Chapter 1

Infrastructure Systems

> "Lacking a national vision or strategy for critical infrastructure renewal and concentrating on single projects, technologies, financing mechanisms, or narrowly defined objectives, ad hoc efforts run the risk of underutilizing or wasting scarce resources and increasing the probability of serious, unintended consequences."
>
> Preface of *Sustainable Critical Infrastructure Systems: A Framework for Meeting 21st Century Imperatives* (NRC 2009)

Societies rely on widespread infrastructure to enable day-to-day activities. The U.S. Department of Homeland Security identifies 18 critical infrastructure sectors (DHS 2009), of which this dissertation focuses on a subset relating to physical resource management such as agriculture and food, energy, water and wastewater, communication, and transportation. These "hard" infrastructure sectors consist of large physical networks of interrelated components which produce and distribute resources. Infrastructure also plays an important role in sustainable development—a long-term perspective on planning commonly described as "[meeting] the needs of the present without compromising the ability of future generations to meet their own needs" (UN 1987). There is present concern of societies consuming natural resources and producing waste products at unsustainable rates, "run[ning] the risk of wasting increasingly scarce resources and of creating new problems for future generations" (NRC 2009).

Interest in infrastructure management and planning at the national level has grown over the past decade. In the U.S., six of the fourteen grand engineering challenges issued by National Academies relate to hard infrastructure (NAE 2008). These infrastructure-related challenges, listed in Table 1.1, span energy, agriculture, water, and transportation sectors. The energy sector seeks to lessen its impact on the natural environment by developing renewable sources of energy such as solar and fusion while reducing emissions of existing carbon-based energy sources through sequestration methods. The agriculture sector seeks to reduce energy use and emissions arising from disruptions to the natural nitrogen cycle with fertilizers. The water sector seeks to provide continued access to water by maintaining natural sources and producing new sources with technology such as desalination. Finally, the urban sector comprising transportation, water, and energy sectors seeks to manage growing populations, protect against natural or targeted disturbances, and accommodate new technologies to continue providing services far into the future.

Two key points arise through inspection of these six infrastructure-related grand challenges. First, there are interrelations between several challenges arising from resource interdependencies between sectors. For example, the agriculture sector is typically the largest consumer of water

Table 1.1: NAE Grand Challenges for Hard Infrastructure

Grand Challenge	Infrastructure Sector Focus
Make solar energy economical	Energy
Provide energy from fusion	Energy
Develop carbon sequestration methods	Energy
Manage the nitrogen cycle	Agriculture and Food
Provide access to clean water	Water
Restore and improve urban infrastructure	Transportation, Energy, and Water

resources and innovations in energy production such as solar power have large impacts on urban infrastructure as a change from large centralized power plants to small distributed generation facilities. Second, many of the challenges are associated with preventing or reversing unintended consequences of past human activity on the natural environment. Some of the greatest achievements of the 20th century include electrification, air conditioning and refrigeration, agricultural mechanization, petroleum and petrochemicals, automobiles, airplanes, and highways (Constable and Somerville 2003), all of which have impacts on the natural environment at large scales.[1] This chapter argues a causal link between these factors: interdependencies among infrastructure sectors may contribute to unexpected, undesired effects.

Study of the unintended consequences of design activities has been discussed in literature for several decades. Reflecting on the expanded role of professionals between 1963 and 1981, Schön writes:

> A series of announced national crises—the deteriorating cities, poverty, the pollution of the environment, the shortage of energy—seemed to have roots in the very practices of science, technology, and public policy that were being called upon to alleviate them.
>
> Government-sponsored "wars" against such crises seemed not to produce the expected results; indeed, they often seemed to exacerbate the crises. The success of the space program seemed not to be replicable when the problems to be solved were the tangled socio-techno-politico-economic predicaments of public life. The concept of the "technological fix" came into bad odor. Indeed, some of the solutions advocated by professional experts were seen as having created problems as bad or worse than those they had been designed to solve. (Schön 1983, pp. 9–10)

In part, past actions relied on "fragile and incomplete" theories which were ineffective to capture the complete nature of social reality. Expanding on these shortfalls, Klabbers (2006) argues the underlying factors of "complexity, uncertainty, and value adjustments" in the context-dependent applications "are not resolved by transforming ill-structured social problems into well-articulated problem-solving tasks." The observations of Schön and Klabbers imply infrastructure design is not solely technical in nature and there is a context dependency not presently captured by analytical methods based on observation and experimentation alone. This perspective aligns with that of de Weck et al. (2012, p. 46) which describes engineering systems research as "(re)thinking about systems" in a way which "goes beyond 'normal' ways of analyzing problems" and "reflects the iterative nature of dealing with systems problems." In particular, this approach may result in a revision of a previous system boundary as new and important interconnections are discovered.

[1] The link between NAE's greatest achievements of the 20th century and grand challenges of the 21st century was first discussed in a presentation by Dr. Richard Miller, President of Olin College.

To reduce the likelihood of the "serious and unintended consequences" identified in the National Research Council workshop committee excerpt opening this chapter, design methods must consider strategic (long-duration) and integrative (multi-sector) implications of infrastructure systems. This dissertation argues existing design methods in isolation are insufficient to tackle the socio-technical sources complexity in infrastructure systems. To frame this discussion, this chapter introduces hard infrastructure as a system of interdependent components. Section 1.1 discusses Masdar City and the International Space Station as two examples of infrastructure systems relying on highly-integrated components to meet sustainability objectives. Section 1.2 expands the discussion to larger-scale infrastructure systems existing in most urban contexts, highlighting the key design challenges in these contexts. Finally, Section 1.3 concludes with the approach and structure of this dissertation to address limitations in infrastructure systems design methods.

1.1 Infrastructure as a System

Infrastructure plays an important role in the transition from labor-intensive societies with large flows of resources from (raw materials) and to (emissions/waste) the natural environment to societies leveraging automation and economies of scale with internal resource flows. This section describes two advanced human societies relying on infrastructure to reduce dependence on external resource flows. First, Masdar City is a community under development striving to be the first large-scale demonstration of technologies supporting a low-waste, low-emission society. Second, the International Space Station is an operational orbiting habitat and laboratory supporting six astronauts. These systems are discussed to identify major components and summarize common design processes.

1.1.1 Masdar City, Abu Dhabi, United Arab Emirates

Masdar City is a new community under development near Abu Dhabi in the United Arab Emirates illustrated in Figure 1-1. It is designed for 40,000 residents and an additional 50,000 commuting workers with ambitions to be a waste-free and low- or zero-carbon economy (Nader 2009). Intended as a testbed for large-scale renewable and sustainable technologies, its initial master plan called for advanced infrastructure components such as a personal rapid transport (PRT) system to efficiently shuttle people and goods about the six square kilometer area, renewable energy from wind and solar photovoltaics (PV), gray water recycling, waste-to-energy conversion, and carbon capture and sequestration (CCS) methods to offset emissions.[2]

Considering a system boundary around the city and nearby infrastructure, Masdar City includes several critical subsystems in Figure 1-2. The centerpiece of the community is the urban system which provides residential, commercial, and light industrial facilities for residents and commuters. Special emphasis is placed on reducing resource consumption through a traditional Arabic architecture with narrow, shaded streets, rooftop solar PV collectors, and dense wall materials to reduce cooling demands. The urban system also governs land use which affects other infrastructure placement. With the exception of rooftop PV panels and water recycling components, all electricity generation, water desalination, and waste transformation components are located outside the immediate city walls.

[2] Economic crises starting in 2008 limited investments and ultimately scrapped the costly undercroft required for the PRT and slipped the scheduled completion date for Masdar City from 2015 to 2025 or later (Cugurullo 2013).

Figure 1-1: Artist's rendering of Masdar City. Large solar photo-voltaic arrays and other infrastructure are visible surrounding the proposed walled city near Abu Dhabi. Credit: Nigel Young / Foster and Partners.

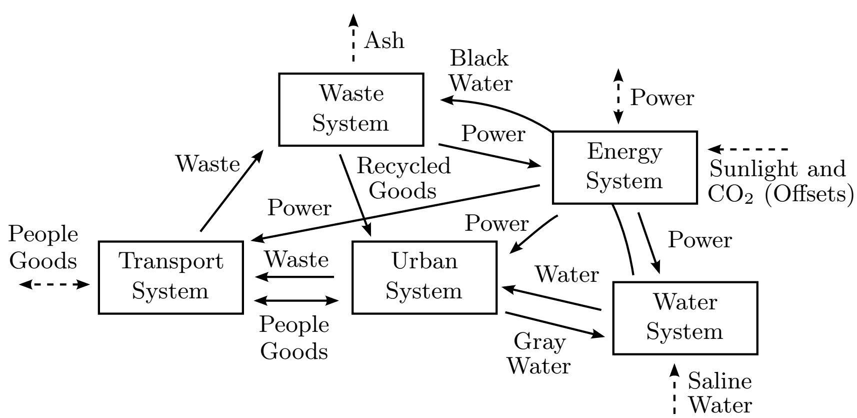

Figure 1-2: System and component dependencies in Masdar City. Solid lines are contained within the city and dashed lines are outside the immediate city.

The transport system includes multiple modes to move people, goods, and waste internally and externally in Masdar City. To meet the objectives of zero/low-emissions, no fossil fuel sources are allowed within the city boundary, instead using electrical power supplied by the energy system. The original master plan calls for internal mobility via a PRT system with stations within 150 meters of any origin and destination, although recent revisions substitute standard electric vehicles at a lower cost. Car parks on the boundary of the city support external mobility of commuters and residents with traditional automobiles forbidden within the city. In addition to personal transport, public transit modes include a proposed metro line to central Abu Dhabi and light rail transport (LRT) connecting to Al Raha Beach and the airport.

The energy system provides a renewable source of electricity, implementing several methods of generation including rooftop solar PV panels, concentrated solar power (CSP), and a wind farm. Each mode provides benefits and drawbacks: PV is light and easy to install but only generates power during the day, CSP creates thermal energy which can be stored for use into the night but requires a large facility with high land use, and wind operates at night but generates variable power dependent on the wind speed. The energy system also interfaces with the external distribution network of the Abu Dhabi Water and Energy Company to mitigate supply and demand fluctuations. Carbon dioxide offsets such as CCS, planting, and greenhouses are proposed to neutralize the emission contributions of external power.

The waste system emphasizes reuse and recycling to minimize net waste. Dry recyclable materials (plastics, glass, metal, paper, etc.) are sorted for material recovery. Non-recyclable waste and gases produced from anaerobic digestion of wet recyclable waste are converted to energy via pyrolysis and gasification processes. Compost and non-hazardous byproducts of the waste-to-energy process including ash are used for other purposes such as fertilizer and building materials.

As rain water is limited in the arid climate, the water system includes desalination infrastructure as an additional source of water to meet the demands of the residents and landscaping. Two potential sources of saline water include the sea at Al Raha (approximately two kilometers away) or ground water pumped via borehole wells. Desalination, however, is an energy-intensive process and additional efforts to treat gray water reduce total energy consumption with initial plans at decentralized facilities within each city block. Non-potable water from treatment processes can also be used for irrigation purposes and black water enters the waste system for disposal.

While Masdar City seeks to minimize waste and emissions as undesirable outputs from human societies, its porous system boundary reduces its ability to do so. Electricity, in particular, is interconnected with the existing Abu Dhabi Water and Energy Company network which still relies on fossil fuels for generation. There is also little consideration for products consumed within the city which, even without waste from packaging, must be produced elsewhere with specialized facilities. Finally, when considering full lifecycle as compared to steady-state operation, Abbasi et al. (2012) note the significance of the "massive carbon debt in the form of greenhouse gas (GHG) emissions entailed in planning, designing and commissioning of the city."

1.1.2 International Space Station, Low Earth Orbit

The International Space Station (ISS) is a space laboratory in low Earth orbit developed as a joint effort between the U.S. National Aeronautical and Space Administration (NASA), the Russian Federal Space Agency (Roscosmos), the European Space Agency (ESA), the Japanese Space Agency (JAXA), and the Canadian Space Agency (CSA). Assembly started in 1998 with the launch of the

Zarya module and was nominally completed in 2011 with the final flight of the *Endeavour* orbiter. Its final configuration includes pressurized and unpressurized modules, docking ports, radiators, and solar panels shown in Figure 1-3. On-orbit operations are scheduled to continue through 2020 with resupply flights from Soyuz and Progress (Roscosmos), HTV (JAXA), ATV (ESA), and commercial vehicles including Dragon (SpaceX) and Cygnus (Orbital).

The ISS provides a habitat for up to six astronauts from partner agencies. Unlike most terrestrial infrastructure systems, its inhabitants are completely dependent on the ISS for all resources required for survival in the vacuum of space. Drawing a system boundary around the orbiting element, several interrelated subsystems in Figure 1-4 perform critical resource supplying and transforming functions for electricity, food, water, air, and other supply items. As the U.S. and Russian segments are partially isolated and redundant, only the U.S. segment subsystems are discussed in detail here.

The human system, comprising the crew and their laboratory, is a critical component of the ISS. The crew perform operational tasks such as scientific experiments and outreach while maintaining and supporting other systems with spares and maintenance procedures. The human system contributes large resource demands by transforming food, water, and gases to waste products.

The power system performs electricity generating and storing functions to supply all onboard components. It includes four large solar photo-voltaic panel modules totaling 70-90 kilowatts of power generation (Gietl et al. 2000). The power system also stores energy in rechargeable nickel-hydrogen batteries for use during regular periods of eclipse lasting about 35 minutes of every 90-minute orbit. The power system works closely with the cooling system which pumps fluid to large radiators for heat rejection to space.

The environmental control and life support system (ECLSS) performs water and gas storing and transforming functions. Its detailed processes include atmosphere control and supply, temperature and humidity control, atmosphere revitalization, water recovery and management, and waste management (Wieland 1998). Many ECLSS subsystems consume power for heating, pumping, or processing fluids. Several ECLSS components also include internal resource reuse loops. For example, air revitalizing and carbon dioxide reducing processes using a Sabatier reactor produce water products, which, along with other wastewater, can be processed into potable water. Electrolysis can also transform water to produce oxygen usable for metabolic air content and hydrogen used in a Sabatier reactor.

Other important ISS components include communications and guidance, navigation and control (GNC) systems. The communications system uses power to transmit messages via the Tracking and Data Relay Satellite System (TDRSS) and direct to the White Sands Ground Station. Data links are used for command and telemetry, crew communication, scientific data, and to guide visiting spacecraft docking operations. Finally, the GNC system maintains the ISS orbit, burning propellant when necessary for orbital corrections.

While the ISS system can operate in isolation for up to several months, it relies on regular interactions with external systems over longer periods. For example, the power system batteries have an expected lifetime of 6.5 years, requiring several replacements by the crew during the ISS life-cycle (Gietl et al. 2000). Similarly, the ECLSS relies on periodic resupply of fluids and other consumable supply items to maintain the necessary physicochemical processes. Finally, crew members operate on rotations of up to 180 days to limit radiation exposure and skeletal-muscular atrophy. Reducing these external dependencies is crucial for enabling future space exploration missions over longer durations and at more distant locations.

Advanced concepts may strengthen dependencies between system components or add new sys-

Figure 1-3: The International Space Station as viewed on April 6, 2009 by the crew of STS-119. It provides all resources required to sustain a crew of six astronauts in low Earth orbit for periods of up to several months. Credit: NASA.

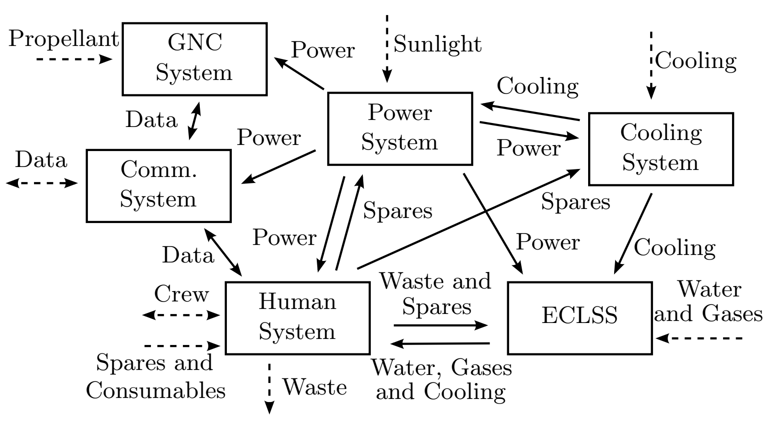

Figure 1-4: System and component dependencies on the International Space Station. Solid lines are contained within the ISS system and dashed lines are outside the immediate ISS system.

Table 1.2: Comparison of ISS and Masdar City Design Approaches

System	Sustainability Objective	Major Interdependencies	Design Approach
Masdar City	Minimize carbon emissions and waste	Urban-Water-Waste-Energy, Urban-Transport-Waste-Energy	Master-planned
International Space Station	Minimize resupply mass from Earth	Human-ECLSS-Power-Cooling	Systems-engineered

tem components. For example, biogenerative technologies interconnect waste and consumable resources (Drysdale et al. 2003). Similarly, repair and scavenging of components rely on crew time to use smaller parts and tools rather than large replacement units (Pettegrew et al. 2007; Oeftering 2010). Finally, future surface exploration may rely upon in-situ resource utilization to produce propellant or life support consumables at remote locations (Sridhar et al. 2000; Sanders et al. 2008).

1.1.3 Design of Infrastructure as a System

Masdar City and the ISS rely on interdependent infrastructure systems to achieve sustainability objectives, summarized in Table 1.2. Constraining resource out-flows, Masdar City minimizes waste and carbon emissions with infrastructure to generate renewable energy and recycle water and waste. Emphasizing resource in-flows, the ISS minimizes resupply mass with infrastructure to recycle water and atmospheric gases. Both applications purposefully create interdependencies between infrastructure systems to meet these objectives. The water and waste recycling processes in Masdar City consume power for transportation and produce power via waste-to-energy conversion. The ECLSS aboard the ISS requires power, cooling, to enable the crew to supply spare parts to other systems and advanced concepts may strengthen dependence between existing systems. Both Masdar City and the ISS adopt a centralized authority in design to account for such interdependencies and work towards the overall sustainability objective.

The design of Masdar City follows a master-planned eco-city approach using the international architectural firm Foster and Partners (F+P) as a central designer. F+P produces a master plan of major infrastructure systems and their integration to achieve the overall objective of low- or zero-carbon emissions within constraints such as land use or capital investment. Unlike many other communities, Masdar City has a wide latitude of design freedom as a greenfield project to grow a society out of existing unused land.

Similar to other space systems, ISS design follows a systems engineering approach to "[develop] an operable system capable of meeting requirements within often opposed constraints" (NASA 2007). NASA maintains design authority over its segments by specifying requirements for component design. Design alternatives may be evaluated using cost metrics such as equivalent system mass (ESM) which quantifies the trade-off in costs between mass, volume, power, cooling, and crewtime (Levri et al. 2003). As launch mass is a major driver of program cost, ESM is a less-contentious system-oriented metric compared to financial cost. Even though various components are contributed by separate designers, the central systems engineer can use metrics such as ESM to evaluate trades across sectors.

The ISS and, on paper, Masdar City can achieve sustainability objectives by virtue of their relatively small scale controllable by a single design authority. While the systems-engineered approach for the ISS has successfully entered operations, the design of Masdar City still boasts significant challenges in "scaling up, and systematically integrating, advanced technologies" (Nader 2009). In a wider frame, however, such centralized control is not possible for infrastructure operating over a much larger scale. The next section discusses these infrastructure systems as a system-of-systems having decentralized control over partially-independent components.

1.2 Infrastructure as a System-of-systems

Large-scale infrastructure systems cannot be designed in the same fashion as Masdar City and the ISS. Rather than a greenfield design managed by a central authority, infrastructure systems are encumbered by past decisions and present operations and have a design process distributed among multiple actors and stakeholders. In this framing, Rinaldi et al. (2001) view infrastructure as a complex adaptive system (CAS), or "complex collections of interacting components in which change often occurs as a result of learning processes." Spanning everything from immune systems to economies, CAS are a class of systems characterized as:

> [having] no single governing equation, or rule, that controls the system. Instead, it has many distributed, interacting parts, with little or nothing in the way of a central control. Each of the parts is governed by its own rules. Each of these rules may participate in influencing an outcome, and each may influence the action of other parts. The resulting rule-based structure becomes grist for the evolutionary procedures that enable the system to adapt. (Holland 1992)

U.S. efforts to study infrastructure systems launched in 1997 with a report by the President's Commission on Critical Infrastructure Protection to address the "growing complexity and interdependence, especially in the energy and communications infrastructures" (PCCIP 1997). Subsequent research efforts studied the emergent behavior arising in infrastructure systems to improve resilience, especially where coupled structure and behaviors may cause widespread failures far beyond the initial scope of a disruption. Recent examples of this phenomenon in the U.S. include the 1994 Northridge earthquake, September 2001 World Trade Center disaster, and August 2003 Northeast blackout (Pederson et al. 2006; O'Rourke 2007; McDaniels et al. 2007). Such "cascading" effects have larger impacts with increasing integration between infrastructure sectors. For example, the 2003 Northeast blackout produced major disruptions to a large percentage of the affected population for nuclear and water utilities, air, roads, and mass transit transportation, hospitals, government offices, and manufacturing and restaurant commerce (McDaniels et al. 2007).

Interdependencies also contribute to coupled effects over longer time-scales. Coupling between water and energy sectors, popularized as the "water-energy" nexus, is particularly important in many regions around the world (Gleick 1994; DOE 2006). Power systems generally require water for cooling, processing or extracting fuels, or electricity generation (e.g. hydropower) and water systems require energy for abstraction (e.g. groundwater pumping), processing (e.g. desalination or recycling), and distribution; however the degree of coupling depends on regional context. For example in the Middle East and North Africa (MENA) region, power systems require little water but sourcing water from desalination and deep aquifers consumes significant amounts of electricity (Siddiqi and Anadon 2011). Other research highlights the coupling of the food and agriculture system which is typically the largest consumer of water and a major consumer of energy though mechanization and fertilization (Bazilian et al. 2011).

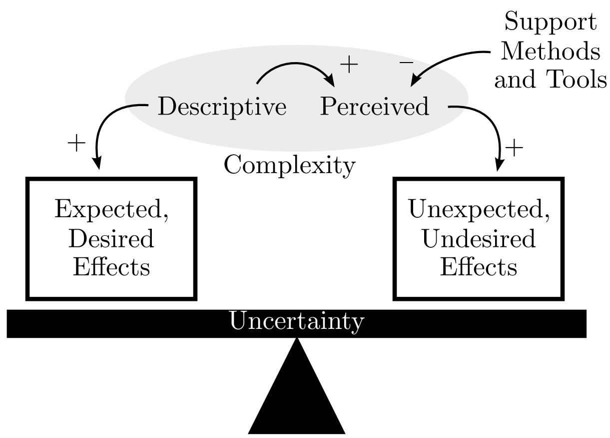

Figure 1-5: Complexity as a balance between expected, desired effects and unexpected, undesired effects.

1.2.1 Complexity in Infrastructure Systems

Interdependency and complexity are terms used to describe challenges in working with infrastructure systems. The abstract property of complexity impedes understanding while interdependency is a particular type or dimension of complexity. There are many attempts to define complexity from various perspectives (e.g. see Sussman 2000). Limited to the context of design, however, Suh (1999) describes complexity as "a measure of uncertainty in achieving the specified [functional requirements]." As complexity increases, so does the uncertainty that particular design will achieve its objectives, possibly leading to unexpected or unintended outcomes.

This framing of complexity is illustrated in Figure 1-5 as an uncertain balance between achieving expected, desired effects and suffering unexpected, undesired effects. It uses the distinction of Schlindwein and Ison (2004) between descriptive (objective) complexity intrinsic to the system and perceived (subjective) complexity dependent on the observer. Expected, desired effects are attributed to the descriptive complexity of the system which has been shown to contribute to improved performance in manufacturing systems provided optimal operation by Deshmukh et al. (1998) and in the process management–simplification tradeoff in supply chains by Frizelle and Woodcock (1995). Unexpected, undesired effects are attributed to limited capacity for understanding of the system as perceived by the designer. Perception, however, is a function of context including methods and tools which may reduce perceived complexity and, correspondingly, the unexpected, undesired effects while maintaining the expected, desired effects.

There are many sources of descriptive complexity as the underlying factor of both desired and undesired effects. Sheard and Mostashari (2010) describe structural, dynamic, and socio-political complexity types in systems engineering. Structural complexity encompasses the size (quantity and diversity) of components, connectivity (number and type of interactions) between components, and architecture (centrality or organization) of connected components.[3] Dynamic complexity includes short-term effects such as nonlinearities, feedback, and stochasticity and long-term effects including context changes and system evolution. Finally, socio-political complexity encompasses

[3]Sinha and de Weck (2012) further quantify architectural complexity as a function of system dependency graph energy. This formulation is lowest for centralized (star-like topology) architectures and increases for more distributed architectures.

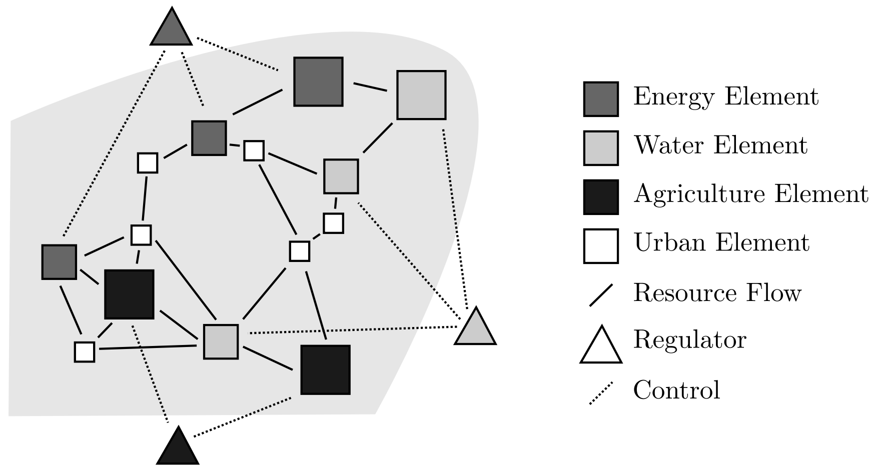

Figure 1-6: A notional infrastructure system with energy, water, agriculture, and urban elements controlled by associated regulators.

the human entities interacting with a technical system. Dodder et al. (2004) further identify evaluative complexity as emerging from differing views and competing objectives of stakeholders and nested complexity arising from interrelationships between the stakeholders and their respective institutions.

Infrastructure systems exhibit high levels of all three types of complexity. Consider the notional example in Figure 1-6 including energy, water, agriculture, and urban infrastructure. Even in an aggregated representation, there are many infrastructure elements in diverse sectors over large geographic areas. Individual elements are mutually interconnected to exchange resource flows and are organized as a distributed architecture without a central hub, even if some sectors are organized with a central hub or layered hierarchy to benefit from economies of scale (e.g. large power or desalination plants). Short-term dynamics respond to fluctuating demands (e.g. daily and seasonal variation) and disturbances from individual element failures. Long-term dynamics accommodate context shifts including technology innovation, quantities and types of resources demanded from populations, and a changing natural environment. Finally, public utilities, private companies, and public-private partnerships control infrastructure elements, which are in turn regulated by agencies or other institutions.

Whereas major infrastructure sectors may have operated more-or-less in isolation in the past, new efforts strengthen the coupling between sectors to increase resource efficiency and meet sustainability objectives. Consider, for example, the effects of wastewater recycling, combined cycle desalination power plants, and electric vehicles. Compared to individual sectors, the total infrastructure system has larger size and connectivity, a more distributed architecture, more complex short- and long-term dynamics, and greater socio-political implications. Returning to the examples of ISS and Masdar City, the higher levels of integration among systems results in more complexity through increased connectivity and short-term dynamics. In these cases, however, the net effect is moderated by centralizing the infrastructure architecture and reducing socio-political impacts (e.g. empowering a master-planner or aligning objectives with ESM as a cost metric). These approaches, however, are not directly applicable to larger-scale infrastructure with distributed authority among interrelated systems. As an alternative, the emerging field of "systems-of-systems" studies the process of influencing design among independent component systems.

1.2.2 Design of Infrastructure as a System-of-systems

Distributed systems lacking an absolute central authority are described in literature as a system-of-systems (SoS). Identifying independence as a key property, Maier defines a SoS as:

> an assemblage of components which individually may be regarded as systems, and which possesses two additional properties:
>
> > Operational Independence of the Components: If the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. That is, the components fulfill customer-operator purposes on their own.
> >
> > Managerial Independence of the Components: The component systems not only *can* operate independently, they *do* operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system-of-systems. (Maier 1998)

Furthermore, Maier (1998) describes three types of SoS depending on the degree of partial control by a central actor. Directed SoS (e.g. integrated air defense networks) are built and managed to fulfill specific purposes and, while component systems maintain an ability to operate independently, a central management guides nominal operations. Collaborative SoS (e.g. the Internet) also have a central management, but without coercive power. Rather, component systems must voluntarily collaborate to fulfill the common purpose. Virtual SoS (e.g. national economies) lack a central management authority or purpose and rely on emergent behavior of component systems. Infrastructure systems may fall into any one of the three types of SoS depending on the perspective and context. For example, a strong central government may operate infrastructure as a directed SoS, while a privatized or market-oriented approach would resemble a collaborative or virtual SoS.

Interest in SoS has grown in recent years in part due to changing acquisition strategies in military domains. The U.S. Department of Defense describes a SoS as "a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities" (DoD 2013). Military actors see SoS as integrated systems relying on new technologies for intelligence, surveillance, and reconnaissance (ISR), command, control, communications, computer applications, and intelligence processing (C4I), and precision force (Owens 1996). An information-oriented SoS requires greater interoperability between component systems and presents a challenge for traditional systems engineering processes (Manthorpe 1996).

Whereby systems engineering is the process of designing a single complex system, system-of-systems engineering (SoSE) is emerging as the process of purposefully shaping a SoS. Keating et al. define SoSE as:

> The design, deployment, operation, and transformation of metasystems that must function as an integrated complex system to produce desirable results. These metasystems are themselves comprised of multiple autonomous embedded complex systems that can be diverse in technology, context, operation, and geography, and conceptual frame. (Keating et al. 2003)

Design activities at the SoS level are quite distinct from those at the system level. For example, Keating et al. (2003) write "SoSE must focus on methodology as primary and process as secondary" because "existing processes for systems engineering can be much too restrictive for SoSE ... [which] must remain flexible to adjust to shifting problem context and conditions." Sage and Cuppan (2001) compare SoSE to federalist approaches of "making things big by keeping them small, encouraging autonomy but within the appropriate bounds of process and architecture standards, and combining variety and shared purpose, individually, and partnerships at national and global levels."

Maier (1998) identifies four architectural principles to support the development of SoS. First, *designing to stable intermediate forms* allow a system to operate in a self-supporting manner, expecting an evolving path to meet stated purposes. Second, *policy triage* recognizes the inability to control all component systems, focusing attention on specific elements such as technical standards and relinquishing control over autonomous components. Third, *leverage at the interfaces identifies* the SoS architecture, i.e. the component interfaces, as the primary design task rather than the components themselves. Finally, *ensuring cooperation* emphasizes mechanism design to promote participation in a SoS and encourage desired behaviors contributing to global, rather than local objectives. To summarize, SoSE activities are somewhat indirect – only the interfaces between component systems are controllable at a SoS level, relying on other mechanisms to encourage the constituent system actors to work towards a common objective.

1.2.3 Key Issues in Infrastructure Systems Design

This section introduced some of the challenges to planning infrastructure systems on large scales beyond the ISS and Masdar City examples presented in the previous section. Infrastructure systems are viewed as both complex adaptive systems (CAS) having distributed components and emergent behaviors and as system-of-systems (SoS) with decentralized authority over constituent systems. This framing leads to two key issues facing design activities for large-scale infrastructure systems.

The first key issue deals with *integration.* Infrastructure systems exhibit interdependencies between sectors, either through dependent resource exchanges (e.g. those described in the water-energy-food nexus) or through internal resource loops required to achieve sustainability objectives. These interdependencies increase system complexity with higher levels of connectivity and a more distributed architecture, potentially leading to unexpected and unplanned outcomes from a proposed design if not well understood. Integrative methods are necessary to consider the system-wide emergent impacts of design decisions and increase the likelihood of achieving desired outcomes.

The second key issue deals with *collaboration.* The decentralized authority in infrastructure systems introduces significant socio-political complexity through differing views, objectives, and interrelationships between various actors and stakeholders. Unlike the processes of systems engineering applied to develop complex systems, no centralized authority exists to coerce constituent systems to meet a global objective. Rather, a SoS requires collaborative methods to align the objectives of independent system actors and solicit the participation of constituent systems.

The dual challenges of integration and collaboration span technical and social fields of study and align with the call from a National Research Council workshop committee for "collaborative, systems-based approaches to leverage available resources and provide for cost-effective solutions across institutional and jurisdictional boundaries" (NRC 2009). This dissertation seeks to answer this call by introducing new SoS-oriented methods to address both integration and collaboration in support of strategic infrastructure system design with the overall goal of reducing perceived complexity and avoiding unexpected, undesired effects.

1.3 Dissertation Approach and Structure

While the larger goal of this work is to promote the efficient and effective use of scarce resources in the design of infrastructure systems to meet long-term objectives, this dissertation focuses on smaller contributing topics oriented around developing and evaluating design methods and tools capable of producing results on shorter time-scales, yet still linked to the overall objective.

1.3.1 Dissertation Structure

This dissertation is structured in eight chapters including this introductory chapter on infrastructure systems. Chapter 2 reviews literature on the topics of design methods for infrastructure systems and related areas in concurrent and collaborative engineering. Based on gaps in the literature, it proposes specific questions to address and outlines a research methodology using both design and analytical science methods to develop and evaluate new artifacts for integrated and collaborative design of infrastructure systems. The proposed artifacts combine interoperable modeling and simulation techniques with gaming concepts to span both technical and social dimensions.

As an initial foray into gaming methods, Chapter 3 frames the challenge of collaborative design by quantifying the added complexity of multi-designer tasks as compared to individual designer tasks. It formulates and executes a human design experiment using simple, context-free design tasks with purposeful barriers to collaboration as a proxy to context-rich infrastructure system design. The results show significant costs for solving tasks as a team as compared to as an individual, suggesting large improvements could be possible with effective collaboration support tools.

Returning to the domain of infrastructure systems, Chapter 4 presents the infrastructure system-of-systems (ISoS) modeling framework for interoperable simulation. It defines core constructs using a graph-theoretic framing of resource flows and identifies structural and behavioral templates for building simulation models. Generality of the ISoS framework is demonstrated with four descriptive use cases for diverse infrastructure systems.

Moving from constructs and models to methods and instantiations, Chapter 5 discusses a software implementation of the ISoS modeling framework using the IEEE Standard 1516 High Level Architecture (HLA). Mirroring the structure of infrastructure systems, the HLA provides decentralized authority over simulation models which also constitute a SoS. This chapter describes required methods to participate in an interoperable simulation using ISoS constructs and models and outlines a sample implementation using the Java programming language.

Building on the general implementation, Chapter 6 introduces a prototype interoperable simulation instantiation using the context of national infrastructure planning in the Kingdom of Saudi Arabia. This application defines system models for agriculture, water, oil and gas, and electricity sectors from a strategic (aggregated) perspective. A graphical user interface (GUI) allows human players to input decisions and view outputs from a simulation execution. The resulting *Sustainable Infrastructure Planning Simulation Game* (SIPS-G) includes roles for three players representing agriculture, water, and energy ministries seeking to develop sustainable infrastructure system plans.

The SIPS-G application is used in Chapter 7 to study collaborative design in a second set of human design experiments. Groups of three players develop a 30-year infrastructure plan to meet individual and team objectives within budgetary constraints. Several individual objectives conflict with team objectives, requiring collaborative approaches among all players. This study evaluates hypotheses that quantitative collective objective metrics and integrated simulation tools result in more effective designs at meeting those objectives. The results show data exchange count is positively correlated with outcome design quality and an integrated simulation tool enables more data exchanges as compared to an asynchronous file-based tool.

Finally, Chapter 8 concludes this dissertation with a summary of research, key contributions to literature, and a critical review leading to future work. Appendices follow with supplemental materials including raw and normalized datasets from human design experiments and additional documentation.

1.3.2 Intended Audiences

This dissertation is intended for audiences in two different communities described by Klabbers (2006). Communities of practice include design scientists who design and evaluate artifacts to solve contemporary problems. On the other hand, communities of observers include analytical scientists who develop and test theories through controlled experiments. Engineering in general acts at the interface of design and analysis, requiring understanding of natural phenomena through controlled experiments as well as creation of new tools and approaches to address practical needs. This dissertation formalizes the associated research methodologies to bridge the practical and observational communities in complementary topics.

Chapters 1, 2, 3, 7, and 8 are directed towards academic audiences in engineering or the applied sciences. These chapters emphasize an integrated discussion of existing literature and develop, execute, and analyze a series of design experiments to evaluate behavioral hypotheses. Chapters 1, 2, and 8 discuss the broader research agenda contributing to infrastructure system design including topics in systems engineering, design research methodologies, and socio-technical systems. Chapters 3 and 7 present the framing and results of human design experiments relevant to fields such as collaborative engineering, management, and social psychology.

Chapters 4, 5, and 6 are directed towards practitioner audiences focusing on the development and use of simulation model artifacts for infrastructure system design tasks. Chapter 4 is of general interest for individuals developing quantitative simulation models for one or more infrastructure systems. Chapters 5 and 6 are of particular interest for developers of interoperable or distributed simulation models with an emphasis on the software architecture and implementation. Finally, Chapter 6 is of interest to individuals studying infrastructure planning in Saudi Arabia as it demonstrates multi-sector modeling in that context.

References for Chapter 1

Abbasi, T., M. Premalatha, and S. Abbasi (2012). "Masdar City: a zero carbon, zero waste myth". In: *Current Science* 102.1, p. 12.

Bazilian, M., H. Roger, M. Howells, S. Hermann, D. Arent, D. Gielen, P. Steduto, A. Mueller, P. Komor, R. S. Tol, and K. K. Yumkella (2011). "Considering the energy, water and food nexus: Towards an integrated modeling approach". In: *Energy Policy* 39.12, pp. 7896–7906. DOI: 10.1016/j.enpol.2011.09.039.

Constable, G. and B. Somerville (2003). *A Century of Innovation: Twenty Engineering Achievements that Transformed our Lives*. Washington, DC: The National Academies Press.

Cugurullo, F. (2013). “How to Build a Sandcastle: An Analysis of the Genesis and Development of Masdar City”. In: *Journal of Urban Technology* 20.1, pp. 23–37. DOI: 10.1080/10630732.2012.735105.

de Weck, O. L., D. Roos, and C. Magee (2012). *Engineering Systems: Meeting Human Needs in a Complex Technological World*. Cambridge, Massachusetts: MIT Press.

Deshmukh, A. V., J. J. Talavage, and M. M. Barash (1998). “Complexity in Manufacturing Systems Part 1: Analysis of Static Complexity”. In: *IIE Transactions* 30.7, pp. 645–655. DOI: 10.1023/A:1007542328011.

Dodder, R., J. Sussman, and J. McConnell (2004). “The Concept of the ‘CLIOS Process:’ Integrating the Study of Physical and Policy Systems using Mexico City as an Example”. In: *Engineering Systems Symposium*. Cambridge, Massachusetts.

Drysdale, A. E., M. K. Ewert, and A. J. Hanford (2003). “Life Support Approaches for Mars Missions”. In: *Advances in Space Research* 31.1, pp. 51–61. DOI: 10.1016/S0273-1177(02)00658-0.

Frizelle, G. and E. Woodcock (1995). “Measuring complexity as an aid to developing operational strategy”. In: *International Journal of Operations & Production Management* 15.5, pp. 26–39.

Gietl, E. B., E. W. Gholdston, B. A. Manners, and R. A. Delventhal (2000). “The Electric Power System of the International Space Station - A Platform for Power Technology Development”. In: *2000 IEEE Aerospace Conference*. Vol. 4, pp. 47–54. DOI: 10.1109/AERO.2000.878364.

Gleick, P. H. (1994). “Water and Energy”. In: *Annual Review of Energy and the Environment* 19.1, pp. 267–299.

Holland, J. H. (1992). “Complex Adaptive Systems”. In: *Daedalus* 121.1, pp. 17–30.

Keating, C., R. Rogers, R. Unal, D. Dryer, A. Sousa-Poza, R. Stafford, W. Peterson, and G. Rabadi (2003). “System of Systems Engineering”. In: *Engineering Management Journal* 15.3, pp. 36–45.

Klabbers, J. H. G. (2006). “A Framework for Artifact Assessment and Theory Testing”. In: *Simulation & Gaming* 37.2, pp. 155–173. DOI: 10.1177/1046878106287943.

Levri, J. A., J. W. Fisher, H. W. Jones, A. E. Drysdale, M. K. Ewert, A. J. Hanford, J. A. Hogan, J. A. Joshi, and D. A. Vaccari (2003). *Advanced Life Support Equivalent System Mass Guidelines Document*. Tech. rep. NASA/TM-2003-212278. Moffett Field, California: National Aeronautics and Space Administration.

Maier, M. W. (1998). “Architecting Principles for Systems-of-Systems”. In: *Systems Engineering* 1.4, pp. 267–284. DOI: 10.1002/(SICI)1520-6858(1998)1:4$<$267::AID-SYS3$>$3.0.CO;2-D.

Manthorpe, W. H. J. (1996). “The Emerging Joint System of Systems: A Systems Engineering Challenge and Opportunity for APL”. In: *Johns Hopkins APL Technical Digest* 17.3.

McDaniels, T., S. Change, K. Peterson, J. Mikawoz, and D. Reed (2007). “Empirical Framework for Characterizing Infrastructure Failure Interdependencies”. In: *Journal of Infrastructure Systems* 13.3, pp. 175–184. DOI: 10.1061/(ASCE)1076-0342(2007)13:3(175).

Nader, S. (2009). “Paths to a low-carbon economy–The Masdar example”. In: *Energy Procedia* 1.1, pp. 3951–3958. DOI: 10.1016/j.egypro.2009.02.199.

National Academy of Engineering (2008). *Grand Challenges for Engineering*. URL: http://www.engineeringchallenges.org/ (visited on 10/25/2013).

National Aeronautics and Space Administration (2007). *NASA Systems Engineering Handbook*. Tech. rep. NASA/SP-2007-610. Washington, DC: National Aeronautics and Space Administration.

National Research Council (2009). *Sustainable Critical Infrastructure Systems: A Framework for Meeting 21st Century Imperatives.* Washington, DC: The National Academies Press.

Oeftering, R. C. (2010). "The Impact of Flight Hardware Scavenging on Space Logistics". In: *AIAA Space 2010 Conference & Exposition.* AIAA 2010-8700. Anaheim, California.

O'Rourke, T. D. (2007). "Critical Infrastructure, Interdependencies, and Resilience". In: *The Bridge* 37.1, pp. 22–29.

Owens, W. A. (1996). "The Emergying U.S. System-of-Systems". In: *Strategic Forum* 63.

Pederson, P., D. Dudenhoeffer, S. Hartley, and M. Permann (2006). *Critical Infrastructure Interdependency Modeling: A Survey of U.S. and International Research.* Tech. rep. INL/EXT-06-11464. Idaho Falls, Idaho: Idaho National Laboratory.

Pettegrew, R., J. Easton, P. Struck, and E. Anderson (2007). "In-flight Manual Electronics Repair for Deep-space Missions". In: *2007 IEEE Aerospace Conference*, pp. 1–16. DOI: 10.1109/AERO.2007.352727.

President's Commission on Critical Infrastructure Protection (1997). *Critical Foundations: Protecting America's Infrastructures.* URL: https://www.fas.org/sgp/library/pccip.pdf (visited on 03/11/2013).

Rinaldi, S. M., J. P. Peerenboom, and T. K. Kelly (2001). "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies". In: *IEEE Control Systems Magazine*, pp. 11–25.

Sage, A. P. and C. D. Cuppan (2001). "On the Systems Engineering and Management of Systems of Systems and Federations of Systems". In: *Systems Management* 2.4, pp. 325–345.

Sanders, G. B., W. E. Larson, K. R. Sacksteder, and C. A. Mclemore (2008). "NASA In-situ Resource Utilization (ISRU) Project–Development & Implementation". In: *AIAA Space 2008 Conference & Exposition.* AIAA 2008-7853. San Diego, California.

Schlindwein, S. L. and R. Ison (2004). "Human knowing and perceived complexity: Implications for systems practice". In: *Emergence: Complexity and Organization* 6.3, pp. 27–32.

Schön, D. (1983). *The Reflective Practitioner: How Professionals Think in Action.* New York: Basic Books.

Sheard, S. A. and A. Mostashari (2010). "A Complexity Typology for Systems Engineering". In: *8th Conference on System Engineering Research.* Hoboken, New Jersey.

Siddiqi, A. and L. D. Anadon (2011). "The water–energy nexus in Middle East and North Africa". In: *Energy Policy* 39.8, pp. 4529–4540. DOI: 10.1016/j.enpol.2011.04.023.

Sinha, K. and O. L. de Weck (2012). "Structural Complexity Metric for Engineered Complex Systems and its Application". In: *14th International Design Structure Matrix (DSM) Conference.* 101. Kyoto, Japan.

Sridhar, K. R., J. E. Finn, and M. H. Kliss (2000). "In-situ Resource Utilization Technologies for Mars Life Support Systems". In: *Advances in Space Research* 25.2, pp. 249–255. DOI: 10.1016/S0273-1177(99)00955-2.

Suh, N. P. (1999). "A Theory of Complexity, Periodicity and the Design Axioms". In: *Research in Engineering Design* 11.2, pp. 116–131. DOI: 10.1007/PL00003883.

Sussman, J. M. (2000). *Ideas on Complexity in Systems: Twenty Views.* Tech. rep. ESD-WP-2000-02. Cambridge, Massachusetts: MIT Engineering Systems Division.

United Nations World Commission on Environment and Development (1987). *Our Common Future.* Oxford University Press.

U.S. Department of Defense (2013). *Defense Acquisition Guidebook*. Washington, DC. URL: https://dag.dau.mil (visited on 03/11/2013).

U.S. Department of Energy (2006). *Energy Demands on Water Resources: Report to Congress on the Interdependency of Energy and Water*.

U.S. Department of Homeland Security (2009). *National Infrastructure Protection Plan*. URL: http://www.dhs.gov/xlibrary/assets/NIPP_Plan.pdf (visited on 11/01/2012).

Wieland, P. (1998). *Living Together in Space: The Design and Operation of the Life Support Systems on the International Space Station*. Tech. rep. NASA/TM-1998-206956/VOL1. Marshall Space Flight Center, Alabama: NASA.

Chapter 2

Methods for Infrastructure Systems Design

> "Today the network of relationships linking the human race to itself and to the rest of the biosphere is so complex that all aspects affect all others to an extraordinary degree. Someone should be studying the whole system, however crudely that has to be done, because no gluing together of partial studies of a complex nonlinear system can give a good idea of the behavior of the whole."
>
> Murray Gell-Mann in *The Quark and the Jaguar: Adventures in the Simple and the Complex* (1994)

Chapter 1 introduced infrastructure systems as a complex adaptive system (CAS) and a system-of-systems (SoS). Both descriptive terms highlight complex features of infrastructure such as evolutionary dynamics, large size with a distributed structure, and important socio-political factors without centralized control. System designers rely on methods and tools to cope with these features of complex systems and support design activities. This chapter introduces topics in the science of design, described by Simon (1996, p. 113) as "a body of intellectually tough, analytic, partly formalizable, partly empirical, teachable doctrine about the design process." Whereas "the natural sciences are concerned with how things are ... design, on the other hand, is concerned with how things ought to be, with devising artifacts to attain goals" (Simon 1996, p. 114). From this perspective, infrastructure systems are artificial objects which provide functions not present in nature.

The following sections discuss relevant design methodologies for complex systems to frame the contributions of this dissertation. Section 2.1 introduces the role of modeling in systems analysis and design and identifies a number of methods targeting particular dimensions of complexity. Section 2.2 discusses integrated methods spanning multiple dimensions to support decision-making in complex systems and identifies limitations in applying existing methods to infrastructure systems. Finally, Section 2.3 presents the research scope, questions, and methodology pursued in this dissertation to fill limitations in existing methods.

2.1 Model-based Methods for Analysis and Design

Zeigler et al. (2000) describes the three fundamental systems problems as:

Table 2.1: Summary of Model Characteristics

Dimension	Bounding Values	
Form	Physical	Mathematical
Solution	Exact	Approximate
Behavior	Static	Dynamic
Uncertainty	Deterministic	Stochastic
State Changes	Continuous	Discrete

Analysis: understanding behavioral characteristics of an existing or planned system,

Inference: inferring behavior from observations of an existing system, and

Design: coming up with a good design for a system which does not yet exist.

Whereas system analysis uses higher-level data to generate lower-level insights, system inference and system design synthesize lower-level data—inference to predict higher-level structure and design to determine desired functionality. An infrastructure system designer analyzes existing systems to identify needs and synthesizes the design of new systems to meet those needs. In summary, the designer "devises courses of action aimed at changing existing situations into preferred ones" (Simon 1996, p. 111).

Both analysis and design activities rely on information. Existent infrastructure systems are sources of observable data which can be collected over time; however available data is typically temporally or spatially aggregated and only covers one particular history (i.e. the actual events). The field of Modeling and Simulation (M&S) uses models to specify relevant system structure within an experimental frame and a simulator to obey the model instructions and produce new data. The objective of the modeling process is to establish an isomorphic relationship between the model and the real-world system using targeted simplifications while maintaining validity.

Law and Kelton (2003, pp. 3–6) characterize models using a number of dimensions summarized in Table 2.1. Physical models use tangible objects to represent abstractions of real-world system components while mathematical models use logical and quantitative statements to relate system components. Some models have an exact analytical solution which can be found for all possible conditions while others require an approximation using a finite set of inputs to produce outputs. Static methods represent a system at one point in time and dynamic methods represent a system changing over time. Deterministic models do not contain any probabilistic components while stochastic models do, resulting in estimated outputs. Finally, continuous simulation methods allow state changes to take place continuously in time while discrete methods only allow state changes at fixed points.

Designers use the information generated from a model of existing or proposed systems to improve understanding of complex factors. Summarizing some of the concepts introduced in Chapter 1, Table 2.2 lists the main types of complexity in socio-technical systems. Various modeling approaches naturally emphasize certain types of complexity and multiple methods are necessary to address all sources of complexity. For example, Jackson and Keys (1984) classify systems methodologies suitable for design problems in mechanical or systemic contexts with unitary or pluralistic decision-makers. Focusing on engineering systems within the systemic-pluralistic context, the CLIOS process

Table 2.2: Types of System Complexity

Type	Sub-type	Infrastructure system example
Structural	Component	Infrastructure elements may themselves be complex systems
	Interactive	Infrastructure exchange resources as inputs and outputs
	Architectural	Multi-sector infrastructure have a decentralized topology
Behavioral	Short-term	Infrastructure operations respond to regular demand variation and unexpected disturbances
	Long-term	Infrastructure respond to changing contexts in evolutionary processes
Socio-political	Evaluative	Infrastructure stakeholders hold different views leading to potentially conflicting objectives
	Nested	Institutional organizations create policies governing interrelationships between infrastructure stakeholders and the physical system

in Dodder et al. (2004) identifies several tools from technical, economic, and social, political, or organizational perspectives to represent structure and behavior and design, evaluate, and implement complex socio-technical systems. This section present a few tools targeting specific dimensions of complexity.

2.1.1 Static Models

Static models describe the components and interconnections in a system from a time-invariant or steady-state perspective. Owing to the underlying network structure of infrastructure systems, several related methods apply graph-theoretic concepts to abstract a physical structure to logical nodes and edges.

An interdependency network is a deterministic analytical mathematical model describing interconnections between components, similar to a design structure matrix (DSM) or N-squared diagram. For example, Rinaldi et al. (2001) define physical, cyber, logical, and geographic interdependencies between infrastructure systems. Dudenhoeffer et al. (2006) formalize these interdependencies for an infrastructure model where a node is “an entity that acts as a source, produces, consumes, or transforms a resource” and an edge is “a physical or virtual entity that acts as a conduit for a flow for a physical quantity, information or influence ... [representing] a direct level of dependence.” This infrastructure model is applied in a software tool to understand the system-wide effects of changing individual or sets of nodes and edges.

Static network models can also represent a form of dynamics from a static perspective. In these cases, it is assumed the model state does not change during the solution process. For example, Zhang and Peeta (2011) develop a multi-layered network to represent transportation, telecommunication, energy, and power infrastructure systems. They apply a computable general equilibrium (CGE) method to determine equilibrium market prices and resource flows between systems. More general network methods applied to infrastructure, often for transportation systems, solve problems such as the minimum cost flow problem, vehicle routing problem, and facility location problem (e.g. Magnanti and Wong 1984). Whereas one can find an exact solution using analytical methods for some network problems, others rely on heuristic methods to find approximate solutions in reasonable amounts of computation time.

2.1.2 Dynamic Models

Dynamic models or simulations allow for a changing model state to reflect time-varying behaviors. There are several interrelated simulation formalisms which capture and process modeled behaviors with a key distinction between continuous and discrete methods (Vangheluwe and Lara 2002). This section discusses three particular approaches—one continuous and two discrete—commonly applied to infrastructure systems.

System dynamics (SD) models define stocks (state variables) and flows (state changes) using a system of continuous mathematical relationships. A SD simulator calculates flows and updates stocks at each time step using numerical methods. As a system of equations, deterministic SD models are relatively fast to execute and are often applied to represent interactions and feedback effects between aggregated systems (e.g. Min et al. 2007). Spatial disaggregation may be handled by replicating a model structure for each unit of analysis and linking the models. For example, the Critical Infrastructure Protection Decision Support System (CIP/DSS) uses a linked SD model to express interactions between 14 critical infrastructure sectors at national and metropolitan scales (Bush et al. 2005; Conrad et al. 2006).

Discrete event simulation (DES) models define a model state and possible state changes as discrete events in time. A DES simulator manages a list of events and processes them in order assuming no state changes happen between events. A single DES execution is fast to run because the simulator only processes the set of discrete events rather than small delta-time periods as in continuous methods. DES are most commonly applied in logistics-oriented applications including port operations (e.g Legato and Mazza 2001), manufacturing (e.g Detty and Yingling 2000), and health care (e.g Connelly and Bair 2004).

Agent-based models (ABM) are a type of DES model which define rule-based agents with internal state and state changes rather than central list of discrete events (as in a traditional DES). An ABM simulator manages the set of agents which interact with each other using discrete pair-wise message events. ABM methods are emerging as a preferred method to study complex systems due to their superior ability to encapsulate information from a bottom-up approach (Borschchev and Filippov 2004). Applied to infrastructure systems, ABM models define actors as individual production or distribution elements (Dudenhoeffer et al. 2006) or suppliers and customers in a resource market (Barton et al. 2000).

The treatment of uncertainty is an important part of dynamic models as results from a single execution of a stochastic model yields little information on the system of interest. There are two key types of uncertainty to consider. First, epistemic uncertainty arises from limited measurements or data and is a practical constraint of model-building. In the absence of additional data-gathering, methods such as sensitivity analysis seek to bound the impacts of epistemic uncertainty by purposely sampling from possible input values (Loucks et al. 2005). Second, aleatoric uncertainty arises from the underlying statistical uncertainty of factors which cannot be resolved with additional data-gathering or modeling efforts. Two types of approaches address aleatoric uncertainty for short- or long-term behaviors.

Short-term behaviors can be characterized with random variables having specified probability distributions based on past knowledge or experiences.[1] Monte Carlo methods, for example, provide outcome distributions through repeated sampling of random variables. While large numbers of

[1] The field of Operations Research (OR) studies optimal decision-making by quantifying short-term behaviors through data or observations and mathematical models.

random samples are required to statistically characterize uncertainties with naïve Monte Carlo methods, approaches such as stratified sampling and Latin Hypercube methods may be used to reduce the number of required samples (Loucks et al. 2005).

Assumptions on fixed contextual factors are relaxed in longer-term analysis. Due to the immense number of variable factors and wide range of possibilities over long time horizons, statistical or probabilistic views of long-term behaviors are severely limited. Rather, strategic decisions are often addressed with a reduced set of possible futures called scenarios (Wack 1985; Zegras et al. 2004) or eras (Ross and Rhodes 2008; Roberts et al. 2009) in which to execute models. Scenario analysis seeks to identify robust decisions which perform well across a range of plausible futures.

2.1.3 Socio-political Models

Models discussed in the previous sections emphasize the technical components of the physical system of interest. Socio-political models study the corresponding social system of actors with the objective of addressing evaluative and nested complexities relating to preferences and interrelationships between stakeholders.

Applications of decision analysis address evaluative complexity by quantifying stakeholder preferences for system properties. Approaches such as multiple criteria analysis (MCA) create a preference model using a weighted sum of attributes as a value function (e.g. Hajkowicz and Collins 2007). MCA may be used either to rationalize decision-making processes in the presence of stakeholders or as a mathematical model of expected stakeholder decisions in their absence. Such models of stakeholder preference can be applied to stochastic models using dynamic programming (DP) to determine optimal decisions at specific points in time.

In another branch of decision analysis, game theory addresses nested complexity by considering reactive effects of small numbers of non-cooperative players in an analytical mathematical formulation. One application of game theory in infrastructure systems determines resource flows as a Cournot-Nash equilibrium among self-interested players (Zhang et al. 2005). As a more general method, stakeholder analysis models interactions between stakeholders to address nested complexity. A stakeholder value network (SVN) is a static network-oriented model describing the structure and value flows of various stakeholders (Cameron et al. 2011). Important design attributes can be identified through SVN analysis to meet needs across various interrelated organizations.

2.2 Integrated Socio-technical Modeling

The previous section introduced several modeling approaches used as tools to address specific complexities in infrastructure systems. Even through methods exist to address both technical and social factors, they are seldom combined in integrated modeling methods and the "interface between technical and social considerations is poorly understood" (Hansman et al. 2006). A few exceptions in emerging literature include dynamic multi-attribute tradespace exploration (Ross 2006) and epoch-era analysis (Ross and Rhodes 2008). These methods combine static or dynamic technical models with multi-attribute utility to address evaluative complexity, but (to date) focus on singular systems design without considering the interrelationships between actors present in SoS.

There are two main limitations to combining social and technical methods. First, the state of social system science is much less mature than that of physical systems which can be decomposed and studied in detail and behave in accordance with governing laws or rules. Second, human decision-making is known to follow a bounded rationality (March 1978), relying on heuristics to make decisions with known biases (Tversky and Kahneman 1974). As the decision-makers are seldom the same people performing analysis and design activities, there is a challenge in communicating insights from modeling activities to overcome the normal course of decisions.

Rather than focusing solely on mathematical models, another potential approach combines actors' experience and knowledge of social systems with a technical model. Not only do the participants address evaluative and nested complexities through their innate understanding, but the process of participating in a modeling exercise may be more effective than analysis alone. Hansman et al. (2006) write "decision makers and other stakeholders need means for seeing, experiencing, and experimenting with infrastructure alternatives and their implications" and "a major research effort should be to develop methodologies to increase effective communication, understanding and alignment among stakeholders" and provide a "virtual 'experimentation' capability." This view also agrees with the call in NRC (2009) for "collaborative, systems-based approaches" which "[recognize] the interdependencies among critical infrastructure systems to enable the achievement of multiple objectives."

Simulation games, viewed as a technical model embedded in a social activity, are an approach which potentially meets these objectives. The physical components of a real-world system are represented in simulation models with which the social components of the real-world system (i.e. stakeholders and decision-makers) interact. This section discusses three existing approaches with similar emphasis currently applied in concurrent engineering, wargames, and other serious games.

2.2.1 Concurrent Engineering

Concurrent Engineering (CE) is "a systematic approach to the integrated, concurrent design of products and their related processes ... intended to cause the developers from the very outset to consider all elements of the product life cycle" (Pennell and Winner 1989). CE emphasizes the couplings between design decisions and outcome performance as an integrated process involving wide participation by designers and stakeholders. In other words, CE considers both the technical integration of actors' design decisions and the social collaboration among participating actors.

Most CE facilities specialize in the design of technically complex systems. For example, both the Jet Propulsion Laboratory's Team X and the European Space Agency's Concurrent Design Facility design space systems, a domain characterized by strong couplings between design variables due to physics-based constraints of spaceflight. In these cases, the objective of a design task is to create a system composed of multiple interdependent subsystems with centralized oversight by the systems engineer (Kolfschoten et al. 2012). Provided a fully integrated system model, methods such as multidisciplinary design optimization (MDO) optimize global objectives across subsystem models and mitigate technical complexity. Past design experiments provide evidence of improved outcome performance using this approach in parallel with human decision-making (Schuman et al. 2005).

The primary social emphasis of CE is communication between subsystem designers relating to technical constraints. Some recent extensions also consider evaluative complexity of designers and decision-makers. For example, MATE-CON applies multi-attribute tradespace exploration (MATE) to CE processes to build preference models using multi-attribute utility theory (MAUT) (Ross et al. 2004). These preference models are used to propose a reduced set of candidate designs which the decision-maker is likely to prefer. Similar approaches are used in Collopy and Hollingsworth (2011) where value is used as an objective function to evaluate design trades in an integrated model.

While CE methods most commonly address system design problems having a centralized management or decision-making process and a systems engineering role, some applications study more distributed decision-making. The Palo Alto Collaborative Testbed (PACT) was an early attempt at improving interoperation in distributed CE systems. The key concept that "tool data and models are encapsulated rather than standardized and unified," allows designers to use the tools and models most natural for their subsystem design (Cutkosky et al. 1993). PACT, and a later study of the Federation of Collaborative Design Agents (FCDA), use federated architectures where facilitators route TCP/IP messages between design agents (Khedro and Teicholz 1997). More recently, the Global Integrated Architecture for iNnovative Utilisation of space for Security (GIANUS) project (ESA 2010) involves multiple ESA establishments to integrate space assets for Earth observation, telecommunications, and navigation. The social dimensions of a SoS CE process includes both the evaluative complexity of particular system stakeholders and the nested complexity of interrelationships between them for collaboration.

2.2.2 Wargaming

Wargaming[2] is an interactive model-based method for planning or training for military conflicts. Portions of the technical system (e.g. terrain, troops, and weapons platforms) are incorporated in a model which is used by human players to make decisions and respond to outcomes from a systems perspective. Differentiating from purely quantitative methods, wargames are an "exercise in human interaction, and the interplay of human decisions and the simulated outcomes of those decisions makes it impossible for two games to be the same" (Perla 1990, p. 164). Further:

> They are of little use in providing rigorous, quantitative measures to 'objectively' prove or disprove technical or tactical theories. Instead, they can often provide the kernel of new theories that can be tested with other tools. Wargaming is most productive when used as an organizing and exploratory tool or as an explanatory device. (Perla 1990, p. 180)

Wargames have a long history applied to systems problems and played a contributing role in developing contemporary systems theory and methods. Wilson (1968) describes a comprehensive history of simulations and games applied to military domains up to the Cold War. Some of the earliest examples of wargames date to the Prussian Empire, with the development of Kriegsspiel (literally, wargames) by von Reiswitz in 1812. Embraced by King Wilhelm III for planning for military engagements, Kriegsspiel used table-top maps, tokens for military units, and an extensive rulebook of quantitative mechanics required to run a wargame. Refinements to these mechanics led to developments such as Lanchester's laws for estimating military force strength during the First World War and scientific management of military operations during the Second World War.

During the 1950s the U.S. military was asked to prepare strategies for potential war scenarios that had never before been experienced (Ghamari-Tabrizi 2000). Analysis methods derived from experience in the Second World War required large amounts of operational data and were not able to plan for an unpredictable future. Organizations including the RAND Corporation developed systems analysis techniques, Monte Carlo methods, game theory, and wargaming, among other methods to address this challenge (Kahn and Mann 1957a; Kahn and Mann 1957b; Kahn and Mann 1957c; Kahn and Mann 1957e). Wargaming in particular was applied to "unfactorable" problems:

[2] The term "wargaming" is also written as "war gaming" in some communities.

those "whose analysis appears to require appreciable context" (Mood 1954). They involved an umpire for adjudication and red (opponents) and blue (allies) teams of experts operating within the context of a scenario to interpret information from a data base and make decisions for which effects are evaluated by models (Perla 1990, pp. 164–167). Wargames were created to meet operational, tactical, and strategic planning tasks and also for teaching and training exercises.

The strength of wargames emerged from their experiential nature and from the communication among experts. Specht (1957) writes "as a teaching device a war game has unparalleled effectiveness, for the player teaches himself and persuades himself in a manner more convincing than any lecture can possibly be" and "a war game teaches both intellectually and emotionally—it is an experience that one lives through." Mood (1954) observes that a wargame "pools knowledge of numerous experts" and forces them to "put down in writing a basic structure which must necessarily be a part of any intelligent consideration of any nonfactorable military problem" such that "people can see it and study it and debate it, and over a period of time arrive at some sort of general agreement about it." Perla (1990, p. 257) also reflects on the benefit of "playing Red" or role-playing one's enemy where identifying "how and why the opponent employs his force as he does is often the most critical element of learning, and also one of the most difficult to interpret."

The amount of detail in wargames expanded with the realization of the early belief that "modern high-speed computers will enable the number of factors which can be included in a game to be increased tremendously, if necessary, without adding to the complexity of the game from the player's standpoint" (Mood 1954). The advent of digital computing greatly increased memory available for capturing state (description of the model variables) and reduced the time required for update calculations. However, complex models introduced new challenges in models, simulations, and games (MSG). Brewer (1975) writes "the actual users are not able to determine how the original knowledge contained in the MSG was generated, for what reasons, and with what limitations" and "the analysts responsible for the information contained in the MSG have abnegated responsibility for their products through disinterest, contempt, and ignorance." Furthermore, extensions of wargames to societal problems were limited as "data on social issues, e.g. the city, poverty, health, housing, do not normally exist in sufficient quantity, quality, or under sufficient control to allow much model building to go on" (Brewer 1975).

Entering the 1980s, efforts at the RAND Corporation attempted to address perceived limitations and "make war gaming more efficient, rigorous, and analytical" using "artificial intelligence techniques to produce computer models able to replace some or all of the human teams" (Davis and Winnefeld 1983). These efforts were not entirely successful, as removing people from the games acted contrary to the role of a game as an exercise in human interaction. Today there are calls for "reinserting people in M&S and related analysis" where "gaming is often a preferred method for operational planners and strategic planners" (Davis and Henninger 2007).

While the classic wargame emphasizes tactical-level decisions, other examples demonstrate strategic-level decisions which tend to border on national policy. Political-military or "pol-mil" games are a direct extension of wargames to understand nested social complexity. Some of the first political games grew out of the RAND Social Science Division to study foreign affairs (Goldhamer and Speier 1959). These policy games were "an attempt to simulate the interaction between states by having individuals play the role of governments dealing with international problems" (Bloomfield 1959). Use of policy games has continued to date—Mayer (2009) reviews the application of games to policy-oriented topics building up from their origins in wargames. In particular, he distinguishes between the "neat and rational" theoretical view of policy-making and the "chaotic messy" realities which can be addressed with human interaction in games. Recently games have been used to address large-scale policy issues such understanding the issues of climate risk management (Suarez et al. 2012).

2.2.3 Serious Gaming

Building on the successes of policy games, serious games apply many of the same principles of military wargaming to meet learning or educational objectives in other domains. Broadly viewed, "a game is an *activity* among two or more independent *decision-makers* seeking to achieve their *objectives* in some *limiting context*" (Abt 1970, p. 6). A serious game combines "analytic and questioning concentration of the scientific view-point with the intuitive freedom and rewards of imaginative, artistic arts" (Abt 1970, pp. 11-12). The qualities which make serious games powerful educational tools such as experiential learning, freedom to fail, and challenging existing mental models also create effective analysis tools. In a sense, the difference is only in learning something known versus unknown. Sterman (1994) argues simulation is essential to improve the decision-making process ("organizational learning") by providing an alternative source of information feedback compared to the costly, time-intensive, and permanent real world.

Business and management practitioners were some of the first game adopters to understand dynamics and social complexities in a competitive environment. The first management game dates to the 1950s when the RAND Corporation created *Monopologs*, a game about the U.S. Air Force logistics system (Jackson 1959). Game use in business education has grown in recent decades. Surveys in Faria (1998) indicate 97.5% of AACSB[3] members use simulation games, and 62.2% of businesses use games in management training programs. Some games are even described as a "management flight simulator"—a manager's analog to a pilot's flight simulator for learning and practicing decisions (Sterman 1992).

There has been a renewed interest in gaming approaches for science and engineering education in recent years. First, in recognition of the blurring of boundaries between modeling, simulation, and gaming, technologies involved in game development can address the "lack of highly-skilled computational scientists and engineers able to fully leverage the current state of the art in [high performance computing] for science-based modeling and simulation" (NRC 2010). Second, simulations and games have potential to meet science learning objectives by supporting "inquiry-based approaches to science instruction, providing virtual laboratories or field learning experiences," and allowing "learners to visualize, explore, and formulate scientific explanations for scientific phenomena" (NRC 2011).

Over the past decade, a series of "infra-games" study various dimensions of infrastructure planning problems. While each game focuses on a different problem, common features emphasize collaborative decision-making and strategic behaviors between actors. Table 2.3 summarizes five selected infra-games discussed in detail below.[4]

The Urban Network Game was created to gain insights to opportunities and threats to developing urban networks of cities with good transportation connectivity (Mayer et al. 2004). Gameplay sessions involved about 50 representatives from relevant administrative, private, and social parties involved in urban network development. Two sessions considered differing scenarios of an existing

[3] AACSB is the Association to Advance Collegiate Schools of Business

[4] Notably, all of these examples come from Delft Technical University Centre for Serious Gaming (CPS) and many under the Next Generation Infrastructures (NGI) Foundation.

Table 2.3: Summary of Selected Infra-games

Name	Reference	Purpose	Game Structure and Interface
Urban Network Game	Mayer et al. (2004)	Generate insights for urban network development	Open negotiation between players with a common physical map and tokens
Infrastratego	Kuit et al. (2005)	Study strategic behaviors in electricity markets	Open negotiation between players with computer model support
SimPort-MV2	Bekebrede (2010)	Develop construction plan for port expansion	Multiplayer computer application with graphical user interface
SprintCity	Nefs et al. (2010); Mayer et al. (2010)	Study interaction between rail	Computer-based multi-player interactive simulation
Rail Cargo Market (RCM)	Meijer et al. (2011)	Manage cargo capacity in rail systems	Paper models with computer support for automated scheduling

urban network (Brabant City) in 2030. Players worked in groups of two to four to develop "innovative spatial designs and projects" to be placed on a common map using physical tokens after negotiation and approval from other players.

Infrastratego is a game developed to study strategic behavior in the Dutch electricity market which was gradually liberalized starting in 1998 (Kuit et al. 2005). It uses a large number of participants (40–50) to play a wide range of roles including electricity generators, national and regional grid managers, suppliers, and other interest groups. The game proceeds in 6-month rounds between 2001–2006 in an open format where participants decide how to play and with whom to negotiate. Gameplay is supported by a computer model based on the Netherlands electricity industry.

SimPort-MV2 is a game developed to experience complexities involved in a large land reclamation project at the Port of Rotterdam in the Netherlands (Bekebrede 2010). 4–6 participants play the board of directors and make construction, negotiation, and financing decisions over 30 simulated years. Each round of 10 years progresses with a timed simulation and performance is evaluated using financial, spatial, and process indicators after each round. Gameplay is supported by a multi-player computer application and graphical user interface, however much of the decision-making takes place outside the computer (Warmerdam et al. 2007).

SprintCity is a game to study the interrelations between rail infrastructure and urban development (Nefs et al. 2010; Mayer et al. 2010). 6–12 players control one of six station areas and experience the interactions between rail infrastructure investment and spatial urban development. Players develop their station areas in five rounds of four years between 2010–2030. The game uses a computer-based simulation to model the interactions of new rail infrastructure and urban development in a mobility-land use reinforcing feedback loop.

Meijer (2012) applies gaming methods to railways in the Netherlands, where infrastructure management and train services are tightly interconnected but controlled by multiple organizations, requiring synchronized processes to increase capacity. Six games were developed to work with the infrastructure management company ProRail during the project period, of which five dealt with operational issues. The exception was the Rail Cargo Market (RCM) game which evaluated the potential value of tactical market mechanisms to manage cargo capacity (Meijer et al. 2011). It included 15–25 players representing cargo transport clients, rail cargo transporters, passenger transporters, rail capacity planners, and rail asset management. Three or four scenarios were used in each session to explore market mechanisms. The simulation game used paper-based models in continuous time with computer support to allow automation of train path reservations.

2.2.4 Limitations in Existing Methods

This section introduced several integrated methods to address complex design problems by bridging technical and social dimensions. For example, CE overlays a social structure on technical models to improve communication and reduce feedback delays between subsystem designers. Gaming combines a technical model with an interactive social environment to engage players and promote learning in a virtual environment. Existing applications of these methods, however, are not sufficient to tackle the design challenges of infrastructure systems. As introduced in Chapter 1, infrastructure systems exhibit a decentralized SoS structure. Modeling methods replicating this structure may result in a stronger isomorphism with the real-world task. Methods must address the dual challenges of *integration* of constituent system models and *collaboration* among independent actors having decentralized authority.

CE methods are most often applied to designs with a centralized authority embodied in the systems engineering role. While CE serves as an integrated environment for sharing data among technical models, it may not be interoperable. Interoperability is defined by Wegner (1996) as "the ability of two or more software components to cooperate despite difference in language, interface, and execution platform." Under a centralized authority designers can be coerced to use a common integrated system. For example, the software environment at ESA's concurrent design facility is based on Excel spreadsheets, imposing specific modeling, software, and operating system requirements on designer models (Schumann et al. 2010). Furthermore, a feasible system design in CE applications is dependent on a compatible set of solutions from all designers. This "consensus" paradigm differs greatly from the "collaborative" perspective of infrastructure systems where the higher degree of independence between system designers allows more autonomy in decisions.

Gaming methods, on the other hand, are often applied to systems problems having a more limited centralized authority. Indeed, the major source of value in wargames and policy games is in role-playing an adversary to identify weaknesses in one's own decisions. Similarly, infra-games use participants to role-play various stakeholders to elicit the nested complexity of a particular design problem. To date, however, the design aspect of infra-games is constrained to a single sector (e.g. electricity, rail, sea ports) in contrast to larger infrastructure systems which involve design activities across constituent systems. Further, the technical models associated with infra-games are often centrally-designed in a lab or studio to capture the particular problem of interest without consideration for integration or interoperability. By the time a game development studio is contracted and completes the work, it is likely to be too late to influence an actual decision, as was experienced in SimPort-MV2 (Bekebrede 2010, p. 264).

Finally, both model-based approaches have long-understood pitfalls which are the subject of several cautionary notes and critical reviews (e.g. Kahn and Mann 1957d; Lee 1973; Brewer 1975). Foremost, a game designer holds great power to shape the perceived reality of players. Specht (1957) writes "the game may persuade us equally convincingly of things that are not true in the real world" and designers should ensure "detail and complexity are compatible both with our knowledge of the real world and with the purposes of the game." Perla (1990, p. 182) writes "a poorly designed game could allow players access to an unrealistic quantity and quality of information and so give those players a false picture of the worth of a weapon system that relies on just such unattainable information to be effective" and cautions "there is always a possibility that intentional or unintentional advocacy or particular ideas or programs may falsely color the events and decisions made in a game and lead to self-fulfilling prophecies." To partially address these concerns, Sterman writes:

> In practice, effective learning from models occurs best, and perhaps only, when the decision makers participate actively in the development of the model. Modeling here includes the elicitation of the participants' existing mental models, including articulating the issues (problem structuring), selecting the model boundary and time horizon, and mapping the causal structure of the relevant system. (Sterman 1994)

Thus, effective games for engineering design should be constructed on the behalf of individuals as participants similar to the nature of CE approaches but with greater interoperability.

To summarize, CE and gaming methods have made significant advances to improving understanding of many sources of complexity in socio-technical systems problems. CE methods rely on integrated modeling techniques but are directed towards systems design problems having a centralized authority. Interactive games incorporate greater nested complexity of various stakeholder perspectives; however recent applications to infrastructure projects focus on a single project and use centrally-developed models.

2.3 Research Outline

To address the limitations in existing methods discussed in the previous section this research seeks to develop interoperable simulation games for infrastructure systems design. This concept extends the technical design emphasis of concurrent engineering to interoperable models necessary to represent the decentralized authority in infrastructure systems and emphasize the social interaction present in gaming. This section outlines the approach taken in this dissertation including the scope of research considered, specific questions to answer, and the overall research methodology.

2.3.1 Research Scope

As introduced in Chapter 1, the overall objective of this research is to influence the design of future infrastructure systems to promote the efficient and effective use of scarce resources. This objective, however, is not possible to study in a timescale of a few years, nor would one expect conclusive experimental results in real-world applications. The research scope of this dissertation is thus limited to developing a design method and evaluating related hypotheses about the behavior of human designers. Evaluation of the method on its ability to improve infrastructure systems design in an applied context is left for future work in a different setting.

Klabbers (2003) identifies two complementary levels of design to frame this distinction. Design-in-the-large (DIL) is "focused on intervention, on devising courses of action aimed at changing existing situations into preferred ones," borrowing the science of design terminology from Simon (1996). On the other hand, design-in-the-small (DIS) deals with creating artifacts used for DIL activities. Here, design artifacts include constructs, models, methods, or instantiations (March and Smith 1995). Thus, with the overall objective of contributing to DIL activities of infrastructure systems design, this dissertation builds and evaluates DIS artifacts contributing to a prototype interoperable simulation game.

A DIS-centered research scope requires additional clarification as some may perceive the researcher as "being more interested in the model than in the real world" (Kahn and Mann 1957d). However:

> This is a criticism only if the analyst is trying to influence policy; if he is trying to advance the state of the art or consciously introducing new tools, then his activities should presumably be judged on a technical basis and it is not necessary for him to introduce substantive considerations. (Kahn and Mann 1957d)

This chapter argues for the utility of interoperable simulation games as an extension of existing CE and gaming methods. While real-world problems are used for context and motivation, the primary contribution is largely methodological.

To further scope the work of artifact design, this dissertation emphasizes three particular sources of complexity. First, it recognizes architecture as a significant source of structural complexity in infrastructure systems. Most attention is given to representing a decentralized architecture with less emphasis on particular infrastructure systems and their interactions. Second, motivated by strategic decision-making, this dissertation emphasizes long-term dynamics rather than short-term. Using aggregated time-steps, the main sources of uncertainty arise from context changes which are difficult to quantify in a probabilistic sense. Rather, the potential sources of contextual uncertainty become discussion points in the social gaming environment similar to the approach taken by scenario analysis. Finally, this dissertation emphasizes nested complexity of multiple actors involved in collaborative design. Integration (via model interoperability) and collaboration are highlighted as key approaches to address design problems with distributed authority.

2.3.2 Research Questions

This dissertation poses three core questions to sequentially 1) motivate, 2) develop, and 3) apply the concept of interoperable simulation gaming for infrastructure systems design.

First, collaborative design problems such as those encountered in infrastructure systems involve complexity from technical sources in the design problem itself and social sources in the structure and interrelationships between designers. However, it is not known how design activities are affected by varying degrees of technical versus social complexity. To motivate the development of integrative and collaborative methods, the first research question asks:

1. What are the relative costs of technical and social complexity in design activities with barriers to collaboration?

This question quantifies the cost of collaboration by evaluating the impact of technical and social complexity in design tasks. Purposeful barriers to collaboration represent the decentralized authority present in systems-of-systems. Insights from answering this question motivate the development of integrated and collaborative methods to reduce the costs of collaboration.

The second research question addresses limitations in existing methods to develop artifacts for interoperable simulation gaming of infrastructure systems. It asks:

2. How can interoperable simulation gaming addresses the dual challenges of integration and collaboration in infrastructure systems?

(a) What generalized modeling framework represents the structure and dynamics of infrastructure systems for integrated modeling activities?

(b) What simulation architecture enables collaboration with decentralized authority over component infrastructure system models?

This question addresses the findings of Rinaldi et al. (2001) that "simply 'hooking' several existing infrastructure models together generally does not work: every model has its own assumptions, data, and numerical requirements ... that may not be compatible with other models." Two sub-questions seek to define a method for integrating and collaborating among infrastructures systems models. Question 2(a) seeks a modeling framework to represent the structure and dynamics of infrastructure systems; specifically, a framework which is generalizable across diverse infrastructure systems capable of producing integrated models. Question 2(b) addresses the independence of infrastructure systems to seek an interoperable simulation architecture enabling independent control over constituent models and collaborative modeling activities.

Finally, the third research question identifies factors contributing to effective design outcomes in a prototype interoperable simulation game. It broadly asks:

3. What design-in-the-small elements of an interoperable simulation game can lead to improved design activity outcomes?

This question explores design-in-the-small options for an interoperable simulation game to improve design-in-the-large outcomes. Although only a first step is taken in this dissertation, future research on this topic can establish design principles for effective interoperable simulation games.

2.3.3 Research Methodology

This dissertation follows a hybrid research methodology combining analytical and design science approaches to answer the three posed research questions. First, one must clarify the differences between analytical and design science methods to address potential misunderstandings. Building on past work including March and Smith (1995) and Hevner et al. (2004), Klabbers (2006) describes analytical science as the "driving force of scientific discovery in physics, chemistry, and biology" by which experimental research studies isolated and purified phenomena. Analytical sciences are performed by communities of observers organized in isolated disciplines, often using a variable approach (i.e. correlations among variables) of justifying and testing theories. On the other hand, the science of design "deals with synthesis" and "is both the world of engineering and the world of social construction of reality as, for example, in education, policy making, and management." Design science is performed by communities of practice who create artifacts to serve human purposes. Artifacts are evaluated and assessed within particular context with a process approach (i.e. linkages between actions and events). Furthermore:

> Games take a dual position. They can be used both to develop and test theories, which is an analytical science objective, and to change existing situations into preferred ones, which is a design science purpose. Both roads require different and, to some extent, mutually exclusive methodologies. (Klabbers 2006)

This dissertation embraces the dual position of games to pursue both analytical and design science methodologies.

Questions 1 and 3 apply simulation games to study human behavior using an analytical science methodology. As the focus is on human behavior, this falls within the boundaries of the social sciences. Answering both questions involves the formulation of a causal hypothesis linking conditions to expected outcomes. Hypothesis evaluation uses a variable approach of measuring correlations between various conditions with observational data. Using a simulation game, behaviors are induced through a controlled human experiment rather than occurring naturally (Babbie 2009, pp. 231–234).

A key component of analytical science, and especially its application to the social sciences, is in establishing strong validity. Internal validity deals with the extent to which the causal correlations observed are correct. The research designs to address Questions 1 and 3, for example, rely on randomized experiments to strengthen internal validity (Campbell and Stanley 1963). Measurement or construct validity deals with the translation from concepts in a hypothesis to quantitative indicators measured in data, i.e. whether "the observations meaningfully capture the ideas contained in the concepts" (Adcock and Collier 2001). Finally, external validity deals with the generalization of findings beyond the experimental frame.

Question 2 develops an interoperable simulation game as an artifact to support design activities for infrastructure systems. As a software-oriented application, this research falls within the field of Information Systems (IS) which "deals with systems for delivering information and communications services in an organization and the activities and management of the information systems function in planning, designing, developing, implementing, and operating the systems and providing services" (Davis 2000).[5] Question 2(a) develops a modeling framework to define constructs and a meta-model of the structure and behavior of infrastructure systems. The constructs provide the language in which problems and solutions are defined and communicated (Schön 1983, pp. 78–79). The meta-model defines relationships between constructs to describe a generalized infrastructure system as a representation of a real world situation. Question 2(b) develops a simulation architecture using these constructs and model to define a method for integrating interoperable models. Finally, the realization of the constructs, models, and methods are embodied in a prototype simulation game to answer Question 2.

Following the framework in Hevner et al. (2004), IS research builds artifacts to address unsolved problems and evaluate them "with respect to the utility provided in solving those problems." They describe seven guidelines for effective research summarized in Table 2.4 with the action taken in this dissertation. In particular, Hevner et al. emphasize that artifacts which interact with humans (including simulation games) must consider knowledge and theories of human behavior and "empirical work is necessary to construct and evaluate such artifacts." Thus, in addition to answering the behavioral science objectives in Question 3, the design experiments are also an opportunity to evaluate the usability of the prototype simulation game artifact.

References for Chapter 2

Abt, C. (1970). *Serious Games*. New York, NY: Viking Press.

Adcock, R. and D. Collier (2001). "Measurement Validity: A Shared Standard for Qualitative and Quantitative Research". In: *Americal Political Science Review* 95.3, pp. 529–546. DOI: 10.1017/S0003055401003100.

Babbie, E. R. (2009). *The Practice of Social Research*. 12th Edition. Cengage Learning.

[5] Alternative names for the IS field include: management information systems (MIS), information management, and management of information systems (MoIS).

Table 2.4: Guidelines and actions for IS design research

Guideline	Description	Action in this Dissertation
1. Design as an Artifact	The result of design science research is, by definition, an purposeful artifact created to address a problem.	Artifacts include constructs and models (Chapters 4), methods (Chapter 5), and a prototype instantiation (Chapter 6).
2. Problem Relevance	Research must address problems faced by practitioners working in the constituent community.	This chapter introduces motivation for new design artifacts based on gaps in existing literature.
3. Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.	Application use cases (Chapter 4), prototype system instantiations (Chapter 6), and human design experiments (Chapter 7) evaluate design artifacts.
4. Research Contributions	Design science research must provide clear contributions in the areas of the design artifact.	Chapters 4, 5, 6, and 8 discuss research contributions.
5. Research Rigor	Design science research requires the application of rigorous methods in the construction and evaluation of the design artifact.	Chapters 4, 5, and 6 document the design and evaluation of contributed artifacts.
6. Design as a Search Process	Design science is recognized as being iterative and problem simplification represents a starting point.	This dissertation builds on past constructs and contributes a prototype system as a first iteration.
7. Communication of Research	Design science research must be presented to technology-oriented and management-oriented audiences.	This dissertation publicly disseminates key research results.

Barton, D. C., E. D. Eidson, D. A. Schoenwald, K. L. Stamber, and R. K. Reinert (2000). *Aspen-EE: An Agent-based Model of Infrastructure Interdependency.* Tech. rep. SAND2000-2925. Albuquerque, New Mexico: Sandia National Laboratories.

Bekebrede, G. (2010). "Experiencing Complexity: A gaming approach for understanding infrastructure systems". PhD thesis. Delft, Netherlands: Technische Universiteit Delft.

Bloomfield, L. (1959). *Political Gaming.* Tech. rep. C/59-27. Cambridge, Massachusetts: Massachusetts Institute of Technology Center for International Studies. URL: `http://dspace.mit.edu/handle/1721.1/83139`.

Borschchev, A. and A. Filippov (2004). "From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools". In: *The 22nd International Conference of the System Dynamics Society.* Oxford, England.

Brewer, G. D. (1975). *An Analyst's View of the Uses and Abuses of Modeling for Decisionmaking.* Tech. rep. P-5395. Santa Monica, California: The RAND Corporation.

Bush, B., L. Dauelsberg, R. LeClaire, D. Powell, S. DeLand, and M. Samsa (2005). "Critical Infrastructure Protection Decision Support System (CIP/DSS) Project Overview". In: *2005 International System Dynamics Conference.* LA-UR-05-1870. Boston, Massachusetts.

Cameron, B. G., T. Seher, and E. F. Crawley (2011). "Goals for Space Exploration Based on Stakeholder Value Network Considerations". In: *Acta Astronautica* 68.11-12, pp. 2088–2097. DOI: `10.1016/j.actaastro.2010.11.003`.

Campbell, D. T. and J. Stanley (1963). *Experimental and quasi-experimental designs for research.* Chicago: Rand McNally.

Collopy, P. D. and P. M. Hollingsworth (2011). "Value-driven Design". In: *Journal of Aircraft* 48.3, pp. 749–759. DOI: `10.2514/1.C000311`.

Connelly, L. G. and A. E. Bair (2004). "Discrete Event Simulation of Emergency Department Activity: A Platform for System-level Operations Research". In: *Academic Emergency Medicine* 11.11, pp. 1177–1185. DOI: `10.1197/j.aem.2004.08.021`.

Conrad, S. H., R. J. LeClaire, G. P. O'Reilly, and H. Uzunalioglu (2006). "Critical national infrastructure reliability modeling and analysis". In: *Bell Labs Technical Journal* 11.3, pp. 57–71. DOI: `10.1002/bltj.20178`.

Cutkosky, M. R., R. S. Engelmore, R. E. Fikes, M. R. Genesereth, T. R. Gruber, W. S. Mark, J. M. Tenebaum, and J. C. Weber (1993). "PACT: An Experiment in Integrating Concurrent Engineering Systems". In: 26.1, pp. 28–37.

Davis, G. B. (2000). "Information systems conceptual foundations: looking backward and forward". In: *eds) Organizational and Social Perspectives on Information Technology, Proceedings of the IFIP TC8 WG8.2 international conference.* Kluwer Academic, pp. 61–82.

Davis, P. K. and A. Henninger (2007). *Analysis, Analysis Practices, and Implications for Modeling and Simulation.* Tech. rep. OP-176-OSD. Santa Monica, California: The RAND Corporation.

Davis, P. K. and J. A. Winnefeld (1983). *The RAND Strategy Assessment Center: An Overview and Interim Conclusions about Utility and Development Options.* Tech. rep. R-2945-DNA. Santa Monica, California: The RAND Corporation.

Detty, R. B. and J. C. Yingling (2000). "Quantifying benefits of conversion to lean manufacturing with discrete event simulation: a case study." In: *International Journal of Production Research* 38.2, pp. 429–445. DOI: `10.1080/002075400189509`.

Dodder, R., J. Sussman, and J. McConnell (2004). "The Concept of the 'CLIOS Process:' Integrating the Study of Physical and Policy Systems using Mexico City as an Example". In: *Engineering Systems Symposium*. Cambridge, Massachusetts.

Dudenhoeffer, D. D., M. R. Permann, and M. Manic (2006). "CIMS: A Framework for Infrastructure Interdependency Modeling and Analysis". In: *Proceedings of the 2006 Winter Simulation Conference*.

European Space Agency (2010). *GIANUS System-of-Systems Architecture Study in the CDF*. URL: http://www.esa.int/esaMI/CDF/SEMCHU7OTCG_0.html (visited on 08/04/2012).

Faria, A. (1998). "Business Simulation Games: Current Usage Levels—An Update". In: *Simulation & Gaming* 29.3, pp. 295–308. DOI: 10.1177/1046878198293002.

Gell-Mann, M. (1994). *The Quark and the Jaguar: Adventures in the Simple and the Complex*. New York: Henry Holt and Company, LLC.

Ghamari-Tabrizi, S. (2000). "Simulating the Unthinkable: Gaming Future War in the 1950s and 1960s". In: *Social Studies of Science* 30.2, pp. 163–223.

Goldhamer, H. and H. Speier (1959). "Some Observations on Political Gaming". In: *World Politics* 12.1, pp. 71–83.

Hajkowicz, S. and K. Collins (2007). "A Review of Multiple Criteria Analysis for Water Resource Planning and Management". In: *Water Resources Management* 21.9, pp. 1553–1566. DOI: 10.1007/s11269-006-9112-5.

Hansman, R. J., C. Magee, R. de Neufville, R. Robins, and D. Roos (2006). "Research agenda for an integrated approach to infrastructure planning, design and management". In: *International Journal of Critical Infrastructures* 2.2, pp. 146–159.

Hevner, A. R., S. T. March, J. Park, and S. Ram (2004). "Design Science in Information Systems Research". In: *MIS Quarterly* 28.1, pp. 75–105.

Jackson, J. R. (1959). "Learning from Experience in Business Decision Games". In: *California Management Review* 1.1, pp. 23–29.

Jackson, M. C. and P. Keys (1984). "Towards a System of Systems Methodologies". In: *Journal of the Operational Research Society* 35.6, pp. 473–486.

Kahn, H. and I. Mann (1957a). *Game Theory*. Tech. rep. P-1166. Santa Monica, California: The RAND Corporation.

— (1957b). *Monte Carlo*. Tech. rep. P-1165. Santa Monica, California: The RAND Corporation.

— (1957c). *Techniques of Systems Analysis*. Tech. rep. RM-1829-1-PR. Santa Monica, California: The RAND Corporation.

— (1957d). *Ten Common Pitfalls*. Tech. rep. RM-1937. Santa Monica, California: The RAND Corporation.

— (1957e). *War Gaming*. Tech. rep. P-1167. Santa Monica, California: The RAND Corporation.

Khedro, T. and P. Teicholz (1997). *The Federation of Collaborative Design Agents*. Tech. rep. 113. Stanford, California: Stanford University Center for Integrated Facility Engineering.

Klabbers, J. H. G. (2003). "Gaming and Simulation: Principles of a Science of Design". In: *Simulation & Gaming* 34.4, pp. 569–591. DOI: 10.1177/1046878103258205.

— (2006). "A Framework for Artifact Assessment and Theory Testing". In: *Simulation & Gaming* 37.2, pp. 155–173. DOI: 10.1177/1046878106287943.

Kolfschoten, G., S. Lukosch, and A. Mathijssen (2012). "Supporting Collaborative Design: Lessons from a case study at the ESA concurrent design facility". In: *Proceedings of the International Conference on Group Decision and Negotiation (GDN)*. Recife, Brazil.

Kuit, M., I. S. Mayer, and M. de Jong (2005). "The INFRASTRATEGO game: An evaluation of strategic behavior and regulatory regimes in a liberalizing electricity market". In: *Simulation & Gaming* 36.1, pp. 58–74. DOI: 10.1177/1046878104272666.

Law, A. M. and W. D. Kelton (2003). *Simulation Modeling and Analysis*. Third Edition. McGraw-Hill, Inc.

Lee Jr, D. B. (1973). "Requiem for Large-Scale Models". In: *Journal of the American Institute of Planners* 39.3, pp. 163–178. DOI: 10.1080/01944367308977851.

Legato, P. and R. M. Mazza (2001). "Berth planning and resources optimisation at a container terminal via discrete event simulation". In: *European Journal of Operational Research* 133.3, pp. 537–547. DOI: 10.1016/S0377-2217(00)00200-9.

Loucks, D. P., E. van Beek, J. R. Stedinger, J. P. Dijkman, and M. T. Villars (2005). "Chapter 9: Model Sensitivity and Uncertainty Analysis". In: *Water Resources Systems Planning and Management: An Introduction to Methods, Models and Applications*. Paris, France: UNESCO, pp. 254–290.

Magnanti, T. and R. Wong (1984). "Network Design and Transportation Planning: Models and Algorithms". In: *Transportation Science* 18.1, pp. 1–55.

March, J. G. (1978). "Bounded Rationality, Ambiguity, and the Engineering of Choice". In: *The Bell Journal of Economics* 9.2, pp. 587–608.

March, S. T. and G. F. Smith (1995). "Design and Natural Science Research on Information Technology". In: *Decision Support Systems* 15.4, pp. 251–266. DOI: 10.1016/0167-9236(94)00041-2.

Mayer, I., S. Meijer, M. Nefs, P. Gerretsen, and D. Dooghe (2010). "Gaming the Interrelation between Rail Infrastructure and Station Area Development: Part 2 – Insights from the serious game 'SprintCity'". In: *Third International Conference onInfrastructure Systems and Services: Next Generation Infrastructure Systems for Eco-Cities*. Shenzhen, China. DOI: 10.1109/INFRA.2010.5679218.

Mayer, I. S. (2009). "The Gaming of Policy and the Politics of Gaming: A Review". In: *Simulation & Gaming* 40.6, pp. 825–862. DOI: 10.1177/1046878109346456.

Mayer, I. S., L. Carton, M. de Jong, M. Leijten, and E. Dammers (2004). "Gaming the future of an urban network". In: *Futures* 36.3, pp. 311–333. DOI: 10.1016/S0016-3287(03)00159-9.

Meijer, S. (2012). "Chapter 11: Gaming Simulations for Railways: Lessons Learned from Modeling Six Games for the Dutch Infrastructure Management". In: *Infrastructure Design, Signalling and Security in Railway*. Ed. by X. Perpinya. InTech, pp. 275–294. DOI: 10.5772/2405.

Meijer, S. A., I. S. Mayer, J. van Luipen, and N. Weitenberg (2011). "Gaming Rail Cargo Management: Exploring and Validating Alternative Modes of Organization". In: *Simulation & Gaming* 43.1, pp. 85–101. DOI: 10.1177/1046878110382161.

Min, H.-S. J., W. Beyeler, T. Brown, Y. J. Son, and A. T. Jones (2007). "Toward modeling and simulation of critical national infrastructure interdependencies". In: *IIE Transactions* 39.1, pp. 57–71. DOI: 10.1080/07408170600940005.

Mood, A. M. (1954). *War Gaming as a Technique of Analysis*. Tech. rep. P-899. Santa Monica, California: The RAND Corporation.

National Research Council (2009). *Sustainable Critical Infrastructure Systems: A Framework for Meeting 21st Century Imperatives*. Washington, DC: The National Academies Press.

— (2010). *The Rise of Games and High-performance Computing for Modeling and Simulation*. Washington, DC: The National Academies Press.

National Research Council (2011). *Learning Science Through Computer Games and Simulations.* Ed. by M. A. Honey and M. L. Hilton. Washington, DC: The National Academies Press.

Nefs, M., P. Gerretsen, D. Dooghe, I. S. Mayer, and S. Meijer (2010). "Gaming the Interrelation between Rail Infrastructure and Station Area Development: Part 1 – Modeling the Serious Game 'SprintCity'". In: *Third International Conference onInfrastructure Systems and Services: Next Generation Infrastructure Systems for Eco-Cities.* Shenzhen, China. DOI: 10.1109/INFRA.2010.5679221.

Pennell, J. P. and R. I. Winner (1989). "Concurrent Engineering: Practices and Prospects". In: *Global Teleconnomications Conference and Exhibition.* Dallas, Texas, pp. 647–655. DOI: 10.1109/GLOCOM.1989.64049.

Perla, P. P. (1990). *The Art of Wargaming: A Guide for Professionals and Hobbyists.* Annapolis, Maryland: Naval Institute.

Rinaldi, S. M., J. P. Peerenboom, and T. K. Kelly (2001). "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies". In: *IEEE Control Systems Magazine*, pp. 11–25.

Roberts, C. J., M. G. Richards, A. M. Ross, D. H. Rhodes, and D. E. Hastings (2009). "Scenario Planning in Dynamic Multi-attribute Tradespace Exploration". In: *2009 International Systems Conference.* Vancouver, Canada.

Ross, A. M. (2006). "Managing Unarticulated Value: Changeability in Multi-attribute Tradespace Exploration". PhD thesis. Cambridge, Massachusetts: Massachusetts Institute of Technology.

Ross, A. M. and D. H. Rhodes (2008). "Using Natural Value-centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis". In: *INCOSE International Symposium 2008.* Utrecht, the Netherlands.

Ross, A. M., D. E. Hastings, J. M. Warmkessel, and N. P. Diller (2004). "Multi-attribute Tradespace Exploration as Front End for Effective Space System Design". In: *Journal of Spacecraft and Rockets* 41.1, pp. 20–28. DOI: 10.2514/1.9204.

Schön, D. (1983). *The Reflective Practitioner: How Professionals Think in Action.* New York: Basic Books.

Schuman, T. E., O. L. de Weck, and J. Sobieszczanski-Sobieski (2005). "Integrated System-level Optimization for Concurrent Engineering with Parametric Subsystem Modeling". In: *1st AIAA Multidisciplinary Design Optimization Specialist Conference.* AIAA-2005-2199. Austin, Texas.

Schumann, H., H. Wendel, A. Braukhane, A. Berres, A. Gerndt, and A. Schreiber (2010). "Concurrent Systems Engineering in Aerospace: From Excel-based to Model Driven Design". In: *8th Conference on Systems Engineering Research.* Paper #1569270867. Hoboken, New Jersey.

Simon, H. A. (1996). *The Sciences of the Artificial.* Third edition. Cambridge, Massachusetts: MIT Press.

Specht, R. D. (1957). *War Games.* Tech. rep. P-1041. Santa Monica, California: The RAND Corporation.

Sterman, J. D. (1992). "Flight simulators for management education". In: *OR/MS Today* 15.5, pp. 40–44.

— (1994). "Learning in and about complex systems". In: *System Dynamics Review* 10.2–3, pp. 291–330. DOI: 10.1002/sdr.4260100214.

Suarez, J. M. de, P. Suarez, and C. Bachofen, eds. (2012). *Games for a New Climate: Experiencing the Complexity of Future Risks.* Boston: The Frederick S. Pardee Center for the Study of the Longer-Range Future, Boston University.

Tversky, A. and D. Kahneman (1974). "Judgment under Uncertainty: Heuristics and Biases". In: *Science* 185.4157, pp. 1124–1131. DOI: 10.1126/science.185.4157.1124.

Vangheluwe, H. and J. de Lara (2002). "Meta-models are Models Too". In: *Proceedings of the 2002 Winter Simulation Conference*. Ed. by E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes.

Wack, P. (1985). "Scenarios: uncharted waters ahead: How Royal Dutch/Shell developed a planning technique that teaches manages to think about an uncertain future". In: *Harvard Business Review*.

Warmerdam, J., M. Knepfle, G. Bekebrede, I. S. Mayer, and R. Bidarra (2007). "The serious game Simport: Overcoming technical hurdles in educational gaming". In: *Learning With Games*, pp. 307–314.

Wegner, P. (1996). "Interoperability". In: *ACM Computing Surveys* 28.1, pp. 285–287.

Wilson, A. (1968). *The Bomb and the Computer: Wargaming from Ancient Chinese Mapboard to Atomic Computer*. New York: Delacorte Press.

Zegras, C., J. Sussman, and C. Conklin (2004). "Scenario Planning for Strategic Regional Transportation Planning". In: *Journal of Urban Planning & Development* 130, pp. 2–13. DOI: 10.1061/(ASCE)0733-9488(2004)130:1(2).

Zeigler, B. P., H. Praehofer, and T. G. Kim (2000). *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Second Edition. Academic Press.

Zhang, P. and S. Peeta (2011). "A generalized modeling framework to analyze interdependencies among infrastructure systems". In: *Transportation Research Part B* 45.3, pp. 553–579. DOI: 10.1016/j.trb.2010.10.001.

Zhang, P., S. Peeta, and T. Friesz (2005). "Dynamic Game Theoretic Model of Multi-Layer Infrastructure Networks". In: *Networks and Spatial Economic* 5.2, pp. 147–178. DOI: 10.1007/s11067-005-2627-0.

This page is intentionally left blank.

Chapter 3

Collaborative Design in Coupled Problems

> "Experience has shown that collaborative projects almost invariably lead to increased costs. When additional participants join a project, the basic costs remain, but the costs of duplicating management systems and of managing interactions must be added. It is also important to recognize that even though the overall cost of the program may increase, the cost to each partner is often decreased, thus making a program more affordable to each partner."
>
> In summary of NRC report *Assessment of Impediments to Interagency Collaboration on Space and Earth Science Missions* (2011)

Design of modern complex systems is a sufficiently challenging task such that multiple actors working as designers are necessary to overcome limitations of an individual (Arias et al. 2000). This is most visible in a system-of-systems (SoS) where multiple organizations manage and operate independent systems rather than relying on a centralized design authority (Maier 1998). While a decentralized structure allows organizations to each work on isolated design problems, it also introduces barriers to collaboration when there are dependencies between each system such as the resource interdependencies observed in infrastructure systems.

The decentralized structure of SoS contributes additional social complexity compared to a centralized system design process. The impact of groups on task performance is multi-faceted (e.g. Cohen and Bailey 1997; Kerr and Tindale 2004). On one hand, there may be improvement through parallel work flows, multiple perspectives on the problem, or specialization. On the other hand, there may be decreased performance through feedback delays between designers with coupled decisions, misalignment of objectives, or poor group dynamics. At the level of agencies or other large-scale organizations, collaborative projects are expected to have an increased cost as discussed in NRC (2011) above in the context of space systems; however it is not known by what extent it exceeds the cost of a centralized effort.

This chapter quantifies and compares the relative contributions of technical and social factors in design tasks. Improved understanding of the social costs to collaboration may help motivate methods to reduce the cost of collaborative design in cases where a centralized authority is not possible. Section 3.1 reviews related literature in complexity and collaboration topics and proposes a research objective. Section 3.2 proposes a multi-actor model of a design activity implemented in a software tool. Section 3.3 outlines the experimental methodology calling for a human subjects experiment. Section 3.4 presents results which are analyzed and discussed in Section 3.5 with topics for future work.

3.1 Related Literature and Objectives

This section reviews related literature in complexity theory and its relation to design activities, cognitive and social psychology research into the capabilities of individuals and teams, and recent research in collaborative engineering. Gaps in existing literature motivate the research question to be studied and an initial hypothesis linking technical and social complexity and design cost.

3.1.1 Related Literature

A large body of literature exists towards describing, and in some cases quantifying, design complexity (Braha and Maimon 1998; Bashir and Thomson 1999; Suh 1999; El-Haik and Yang 1999; Ameri et al. 2008) and especially for software (Card and Agresti 1988; McCabe and Butler 1989; Fenton and Neil 1999). As discussed in the previous chapters, complexity can be aggregated under three types with multiple subtypes: structural (component, interaction, and architecture), dynamics (short- and long-term), and socio-political (Sheard and Mostashari 2010). Even aggregated as such, there are likely relationships between types; for example, between structural architecture and social organization under Conway's Law which recognizes that "products tend to 'mirror' the architectures of the organizations in which they are developed" (MacCormack et al. 2011).

Few studies directly investigate the effect of technical complexity on design performance, although it is generally perceived as contributing to more errors and lower productivity (Card and Agresti 1988) and to higher schedule and cost overruns (Bashir and Thomson 1999; Arena et al. 2008). Hirschi and Frey (2002) quantify the effect of technical complexity on time to complete a task using a linear system of equations as a surrogate for a parameter design task. They find the normalized time to solve the task grows linearly with the number of uncoupled variables but geometrically with the number of coupled variables (much faster then the polynomial growth in numerical solvers). The differences are explained from the cognitive psychology perspective of short-term memory, limited to seven plus or minus two chunks of information (Miller 1956). Sinha (2014) finds similar results for a super-linear effect of complexity on development effort across a range of cyber-physical systems. In a related time-constrained human subjects study on building systems, Flager et al. (2014) shows the resulting design quality exponentially decreases with the number of variables.

In a multi-actor setting, however, there is no single memory and knowledge is distributed, requiring a shift from theories of cognitive psychology to those of social psychology. Several studies identify negative effects of group size described as social loafing or the Ringelmann Effect, attributed to coordination and motivation losses (Kravitz and Martin 1986; Ingham et al. 1974). Other studies investigate various factors impacting group performance such as cohesion composed predominately of group pride (Mullen and Copper 1994), friendship mediated through cooperation and commitment (Jehn and Pradhan 1997), task and team familiarity (Goodman and Leyden 1991), and trust mediated through motivation (Dirks 1999). While these studies investigate factors at the individual- and team-level, it is not known if they may be extensible to inter-organizational relationships as well.

A few studies specifically address the effect of complexity—emphasizing task complexity—on group performance. Weingart (1992) finds component complexity, defined as the number of unique actions required to complete a task, increases both the amount and quality of planning for some aspects of the task and decreases the group effort, both effects mediating lower group performance. Argote et al. (1995) finds component complexity has a negative main effect on group performance which grows as groups gain experience. Turnover of group members also has a larger effect in simple tasks compared to complex ones, possibly again due to social loafing behaviors.

The application of social science research to support group design tasks is described as collaborative or collaboration engineering, a field which "facilitates the communal establishment of technical agreements among a team of interdisciplinary stakeholders, who work jointly towards a common goal with limited resources or conflicting interests" (Lu et al. 2007). As a nascent area of research, there is little empirical literature relating technical and social complexity to collaborative performance, however several contributing factors are identified. A common goal of collaboration support systems is to create shared knowledge among designers (Arias et al. 2000). This concept is expanded by considering collaborative design as a negotiation with four steps: 1) interaction among designers to 2) construct a common understanding leading to 3) a group preference, and finally 4) attain agreement on a design (Lu et al. 2007).

To summarize, literature on engineering design emphasizes technical complexity of the design task with limited consideration of social factors relevant for multi-actor design. On the other hand, literature from social psychology focuses on social factors contributing to group performance with only modest consideration of technical complexity. These studies also focus on routine tasks rather than the more creative process of design. Literature in collaborative design combines both the technical and social dimensions, but has not yet quantified a relationship between the two factors and design performance.

3.1.2 Research Objective

This research seeks to address Question 1 previously posed in Chapter 2:

1. What are the relative costs of technical and social complexity in design activities with barriers to collaboration?

Past research identifies a link between complexity and cost; however the relative contributions from technical and social factors are not known. The past work of Hirschi and Frey (2002) shows that task completion time (as a measure of cost) grows linearly for technically-simple tasks and geometrically for technically-difficult tasks; however only single-designers are considered.

Based on related literature, it is hypothesized that increasing social complexity is positively correlated with design cost. Furthermore, it is hypothesized that there is a positive interaction between technical and social complexity on design cost (i.e. higher costs of increasing social complexity for technically-complex problems). To evaluate these hypotheses, this chapter proposes a human subjects experiment using a simplified multi-actor design model described in the following section providing experimental control over variables of interest.

3.2 Multi-actor System Design Model

This section describes a surrogate for a parameter design process based on Hirschi and Frey (2002). This method focuses on technical and social complexity by providing experimental control over:

1. Number of input and output variables in the design task,
2. Degree of coupling between input and output variables, and
3. Assignment of input and output variables among multiple designers including the degree of coupling between designers.

The surrogate also removes all real-world context from the design problem to eliminate effects of domain knowledge or experience. The resulting design tasks can be solved in a short time period suitable for study. The surrogate design task is described in the following sections including the underlying multi-actor system model, its limitations, and a distributed software application implementing model.

3.2.1 Parameter Design Formulation

Suh (1999) describes the design process as a mapping between functional requirements (FRs) in the functional domain and design parameters (DPs) in the physical domain. The objective of the design process is to find the set of DPs (input variables) to achieve a specified set of FRs (target output variables).

In its most general form, the surrogate design task uses the parameter design formulation to map an input vector $\mathbf{x}$ to an output vector $\mathbf{y}$ through a system model $\mathcal{M}$ in Eq. 3.1.

$$\mathcal{M}(\mathbf{x}) = \mathbf{y} \tag{3.1}$$

An error model $\mathcal{E}$ in Eq. 3.2 maps the difference between the output vector $\mathbf{y}$ and target output vector $\mathbf{y}^\star$ to an error vector $\mathbf{z}$.

$$\mathcal{E}(\mathbf{y} - \mathbf{y}^\star) = \mathbf{z} \tag{3.2}$$

The objective of the design task can be framed in Eq. 3.3 as finding an input vector $\mathbf{x}$ such that the output error is less than a specified bounds $\mathbf{z}^\star$.

$$\text{find } \mathbf{x} \text{ s.t. } \mathcal{E}(\mathcal{M}(\mathbf{x}) - \mathbf{y}^\star) < \mathbf{z}^\star \tag{3.3}$$

Hirschi and Frey (2002) constrain the general surrogate design task to a single-actor design problem using a linear system model shown in Eq. 3.4.

$$\mathcal{M}(\mathbf{x}) = M\mathbf{x} \tag{3.4}$$

Here, M is a transformation matrix mapping input to output variables, effectively a design structure matrix. The problem space is limited to square, orthonormal M matrices to ensure a consistent linear system. For a system model with N inputs and outputs, Eq. 3.5 shows the M matrix structure where element m_{ij} quantifies the coupling between input i and output j, i.e. $m_{ij} = \partial y_j / \partial x_i = \mathrm{d}y_j / \mathrm{d}x_i$.

$$M = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1N} \\ m_{21} & m_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ m_{N1} & m_{N2} & \dots & m_{NN} \end{bmatrix} \tag{3.5}$$

The corresponding error model takes the absolute value of the difference between each output and the target value,

$$\mathcal{E}(\mathbf{y} - \mathbf{y}^\star) = \{|y_i - y_i^\star|\} \;\forall\, i, \tag{3.6}$$

and the error bounds are specified by a constant ϵ such that

$$\mathbf{z}^\star = \{\epsilon\} \;\forall\, i. \tag{3.7}$$

This solution criteria requires all outputs be within ϵ of the target value. While there is a unique solution with zero error for a consistent linear system (namely $\mathbf{x} = M^{-1}\mathbf{y}^\star$), there are infinitely-many solutions within the specified error bounds.

The multi-actor surrogate extends the single-actor formulation by assigning each input and output to a designer. From this perspective, input assignments represent control over design parameters and output assignments represent objectives tied to functional requirements. For a design task with n designers, the assignments are formalized by two binary (0,1) matrices. An $n \times N$ matrix I assigns inputs, where element I_{ij} is defined in Eq. 3.8.

$$I_{ij} = \begin{cases} 1 & \text{if input } j \text{ is assigned to designer } i \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

Similarly, an $n \times N$ matrix O assigns outputs, where element O_{ij} is defined in Eq. 3.9.

$$O_{ij} = \begin{cases} 1 & \text{if output } j \text{ is assigned to designer } i \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

The assignment matrices can be composed with the M matrix to compute the social coupling matrix D shown in Eq. 3.10.

$$I \times M \times O^\intercal = D = \begin{bmatrix} D_{11} & \dots & D_{1n} \\ \vdots & \ddots & \vdots \\ D_{n1} & \dots & D_{nn} \end{bmatrix} \tag{3.10}$$

Element $D_{ij} \neq 0$ identifies social coupling between designers i and j, specifically that designer j outputs are dependent on designer i inputs.

An example multi-actor design task with three designers ($n = 3$) and four variables ($N = 4$) is defined in Eq. 3.11. Inputs are assigned to designers $\{1, 1, 3, 4\}$ and outputs are assigned to designers $\{1, 1, 3, 2\}$.

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 \\ m_{21} & m_{22} & m_{23} & 0 \\ m_{31} & m_{32} & m_{33} & 0 \\ 0 & 0 & 0 & m_{44} \end{bmatrix}, I = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, O = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.11}$$

The resulting D matrix computed from M, I, and O using Eq. 3.10 is shown in Eq. 3.12 which

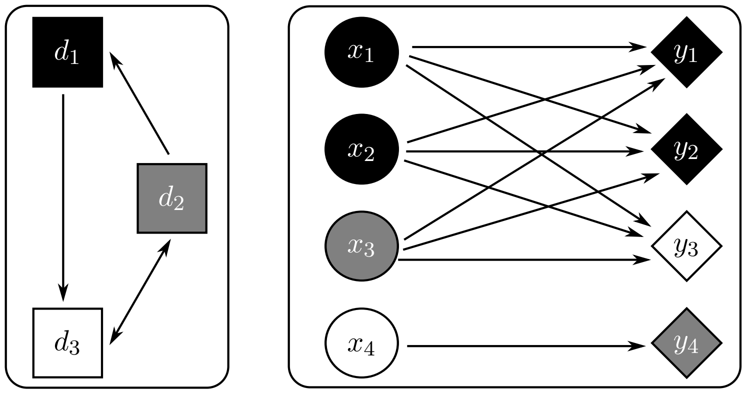

Figure 3-1: The multi-actor formulation can be represented as a hyper graph connecting the inputs (circles, x_i), outputs (diamonds, y_i), and designers (squares, d_i) in technical (right) and social (left) layers. The M matrix describes technical coupling while the D matrix describes social coupling. Implicit edges connect designers, inputs, and outputs having the same color (assignment).

identifies dependencies between all three actors.

$$D = \begin{bmatrix} m_{11} + m_{12} + m_{21} + m_{22} & 0 & m_{13} + m_{23} \\ m_{21} + m_{32} & 0 & m_{33} \\ 0 & m_{44} & 0 \end{bmatrix} \tag{3.12}$$

Figure 3-1 illustrates a two-layer hyper graph of the design task. The technical layer (right) illustrates the coupling between inputs and outputs due to the M matrix while the social layer (left) illustrates the coupling between designers derived from the D matrix (with self-loops omitted). Input and output assignments derived from the I and O matrices respectively connect the two layers, illustrated with color-coded assignments for the three designers (black, gray, and white).

3.2.2 Model Assumptions and Limitations

The multi-actor system design model makes several simplifying assumptions which introduce limitations to the generalization of results outside the experimental frame. The two key assumptions include the existence of a single zero-error solution and a linear form of the system model.

This formulation assumes there is exactly one zero-error solution to the design task. This is one of three possible cases for a general design task which may be:

1. Over-determined if there are no zero-error solutions,
2. Under-determined if there are more than one zero-error solution, and
3. Uniquely determined if there is exactly one zero-error solution.

In the first case, there are no zero-error solutions to the design task. Arguably, this case covers some real-world tasks where no design meets all requirements. There are two potential outcomes: either the task is an exercise in futility or requirements must be relaxed to find a feasible solution. The process of relaxing requirements, operationalized in changing target outputs, is potentially interesting to study in future work focusing on negotiation in collaborative design. Such an extension, however, may cause the task to become under-determined from the new framing.

In the second case, there are more than one solution exactly meeting all functional requirements. Many real-world design tasks also follow this case, with decisions oriented around meeting desirability objectives such as minimizing cost or maximizing value. Without including such secondary objectives in the formulation, however, all solutions are equivalent, and one would expect the same outcome as a reduced-order system (i.e. a similar problem with additional constraints) having a unique solution. An extension providing an objective function rather than a solution criteria may potentially confound experimental variables by measuring the designers' ability to maximize desired objectives in addition to finding a feasible solution.

The third case of a unique solution is least similar to real-world design tasks, but most applicable to the experimental framing. A single solution criteria provides a concise measurement of design task complexity in the number of coupled variables. It aligns all designers' goals such that secondary objectives potentially requiring negotiation are not needed. Thus, while not directly representative of real-world design, it includes many of the same characteristics of interest. Finally, it should be noted that while there is a single non-zero error solution, the error bounds ϵ provide a range of acceptable solutions. This is a practical consideration rather than theoretical, allowing designers to find a "close-enough" solution using discretized input methods. For example, finding an exact solution may be impossible if inputs are discretized and the solution is an irrational number. Thus, ϵ should be small enough to approximate a single solution while large enough to allow its discovery.

A second limitation arises from the use of a linear system of equations described by the coupling matrix M. This is partially justified by the intentional lack of context in the surrogate design tasks. While most real-world systems are not linear, they also have context to understand non-linearities. Physical laws and mathematical models encapsulated in domain knowledge allow experts to manage non-linearities. However, in this context-free case, even linear systems are not necessarily perceived as simple due to limited cognitive abilities without quantitative aids as found in Hirschi and Frey (2002). Finally, a linear system model provides three practical advantages. First, it is the simplest model of technical coupling, requiring minimal assumptions. Second, consistent linear systems with a single zero-error solution can be generated for arbitrarily-complex design tasks using randomized orthonormal matrices. Finally, outputs can be rapidly computed with matrix multiplication.

3.2.3 Software Implementation

The multi-actor system model is implemented in a distributed software application illustrated in Figure 3-2 with separate components for each designer and an administrator. The designer components include a graphical user interface (GUI) to specify inputs and view outputs. The administrator component assigns inputs and outputs to each designer using the M, I, and O matrices defined for a given task. Changes to input values are sent from a designer to the administrator component which computes and sends the new output values.

The application is designed to simulate barriers to collaboration in three ways. First, the system model is hidden such that each designer can only observe effects of input value changes on their assigned outputs. Second, no quantitative information is displayed to prevent designers from mathematically solving the linear system outright. Finally, designers are limited to verbal communication to simulate barriers across organizational boundaries.

The designer GUI includes slider components for the assigned inputs and outputs, as shown in Figure 3-3 for a designer with two inputs and two outputs. When in use, the GUI runs in full-screen mode to focus attention on the task. A randomly-generated task name is displayed at the top, also

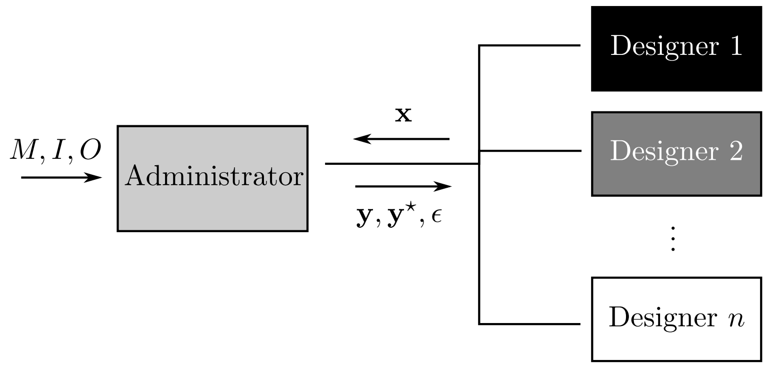

Figure 3-2: The multi-actor system model implemented as a distributed software application. The controller receives inputs (x_i) from designers and sends outputs (y_i) based on the design task specified by the M, I, and O matrices.

indicating whether it is a team (socially coupled) or individual (socially uncoupled) task. Inputs have randomly-assigned Greek letters and outputs have randomly-assigned labels. These are not intended for, nor perceived as physically intuitive, but rather meant to identify specific variables. A designer's input can be modified by dragging the slider thumb up or down with a mouse or by pressing the up or down keys (moves 0.5% or 0.01 units on the expected -1.0, 1.0 range) or the page up and page down keys (moves 5% or 0.1 units on the expected -1.0, 1.0 range). While dragging the slider thumb, inputs only update once the thumb is released. In both cases, only one update request per designer can be active at any time to limit the update rate.

The output sliders display a green region approximating the acceptable output values within the error bound ϵ of the target. When the slider thumb enters this region, the signal icon changes from a red cross to a green check mark and the background changes from light red to green. In situations where the slider thumb is "out of range," (i.e. its true value cannot be displayed), the background turns gray. All three states were designed and verified for use by individuals with color blindness. A design task is completed when all outputs are within the target range, at which time all designers are prevented from modifying their input values and the manager application plays a short success sound effect. Although a minor addition, the audio feature had a strong effect on the observed satisfaction of designers.

The GUI design differs slightly from that of Hirschi and Frey (2002). First, it does not require the designer to explicitly press a "Refresh Plot" button to limit feedback rates, however updates are still rate-limited by network latency. By allowing easier and more rapid updates, but not necessarily changing the underlying task, one would expect this design to produce a similar scaling law with smaller time-scales and potentially more noise. For example, a few seconds of delay or distraction has more impact in a short-duration task as compared to a long-duration task. This limitation is offset, however, by the shorter total time required to complete tasks, allowing more data points to be collected. Finally, the output display is modified such that the sliders are horizontal (in contrast to vertical input sliders) to prevent unintentional aids from visually aligning the input slider thumbs.

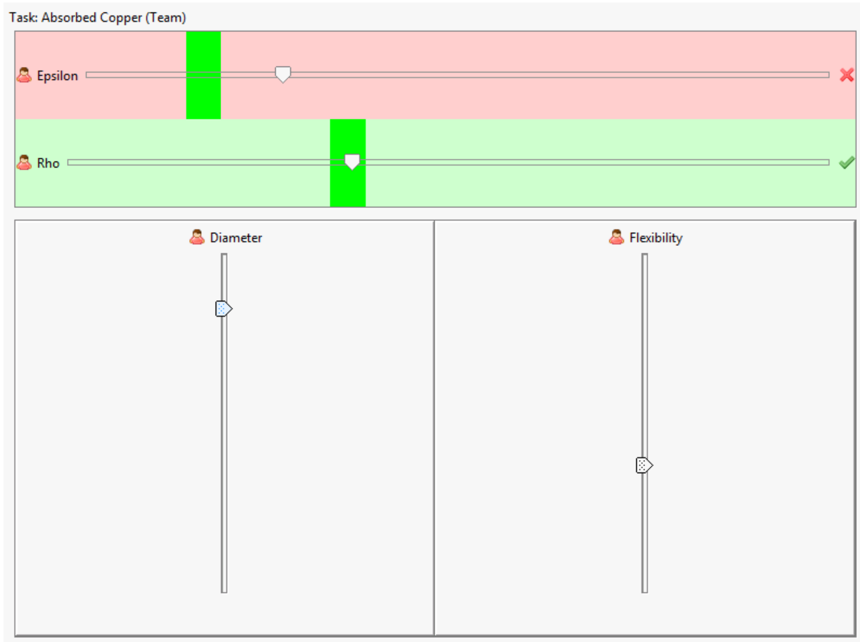

Figure 3-3: The designer GUI includes slider(s) for assigned inputs and outputs. The input DP slider(s) respond to user actions and the output FR slider(s) show the target value and an icon signaling when within an acceptable range.

3.3 Experimental Methodology

The experiment is structured as a within-subjects study with a design task as the unit of analysis. The number of variables and degree of coupling in a task operationalize technical complexity. The number of designers operationalizes social complexity. Finally, the time required to complete a task (i.e. from the time a subject makes the first input change to the time that all output targets are met) operationalizes design cost.

3.3.1 Experimental Design

The multi-actor system formulation leads to four types of design tasks:

I. Uncoupled decisions within designers (M and D diagonal)

II. Coupled decisions within designers (M unconstrained, D diagonal)

III. Uncoupled decisions across designers (M diagonal, D unconstrained)

IV. Coupled decisions across designers (M and D unconstrained)

where the first two items are the cases studied by Hirschi and Frey (2002). This experimental design adds the additional dimension of social coupling to the existing results to comparatively evaluate type III and IV design tasks.

Table 3.1: Design task type with n coupled designers and N variables

n	N (uncoupled)					N (coupled)			
	2	3	4	5	6	2	3	4	5
1 (Individual)	$\text{I}^{\dagger}$	$\text{I}_1^{\dagger}$	$\text{I}_2^{\dagger}$	$\text{I}^{\dagger}$	I_3	$\text{II}_4^{\dagger}$	$\text{II}_5^{\dagger}$	$\text{II}_6^{\dagger}$	$\text{IV}^{\dagger}$
2 (Pair)	III	III_7	III	III	III	IV_8	IV_9	IV_{10}	IV
3 (Triad)	–	III_{11}	III	III	III_{12}	–	IV_{13}	IV_{14}	IV

†: task in Hirschi and Frey (2002)
$_{\#}$: task number in this experimental design

Table 3.1 illustrates the design task type for a range of variables and designers. Only cases with three designers were considered for experimentation for practical scheduling reasons. The 14 numbered design tasks address the following experimental objectives:

1. Validate results from Hirschi and Frey (2002): five cases (tasks 1, 2, 4, 5, 6)

2. Vary social coupling while holding technical coupling constant: three cases with three levels each (tasks 1, 7, 11; 5, 9, 13; 6, 10, 14)

3. Vary technical coupling while holding non-zero social coupling constant: one case with three levels (tasks 8, 9, 10).

For pair tasks, the third designer either has a blank screen and does not participate (tasks 8, 9, and 10) or completes a decoupled problem (task 7). No design tasks involve a partially-coupled technical system model in this design (i.e. M is either diagonal or complete). If implemented, such tasks could benefit from an organized task sequence to reduce complexity (Eppinger et al. 1991). This condition could potentially be a source for future design experiments.

Figure 3-4 illustrates designer assignments in each of the 14 tasks and the number of experimental replications in each session. Individual tasks are conducted in parallel for the three designers, with one replication for tasks 1–3 and two replications for tasks 4–6. Three replications with rotating assignments are also used for tasks 7, 8, and 9, and two replications are used for task 11. Finally, due to time constraints, only one replication is used for tasks 10, 12, 13, and 14. In all, this experimental design calls for 9 individual and 15 team tasks to be solved for a total of 24 tasks—42 replications considering parallel individual tasks.

Unique design problems for the 42 replications are generated for each session to mitigate potential effects of a particular design problem instance. Each transformation matrix M for coupled tasks is generated from orthonormal bases of random vectors with elements drawn from a uniform (0,1) distribution. The resulting orthonormal matrix guarantees well-conditioned and balanced relationships between inputs and outputs and consistency of a single solution for all tasks. For uncoupled tasks the diagonal of M contains flips between 1 and -1 with probability 0.5. The error bounds is set to $\epsilon = 0.05$ for all tasks such that the target range with width $2\epsilon = 0.1$ covers 5% of the range between the expected output range of -1.0 to 1.0.

The initial value of all inputs is zero ($\mathbf{x}_0 = 0$), and therefore the initial output is also zero ($\mathbf{y}_0 = 0$). A target output vector ($\mathbf{y}^{\star}$) is generated from an orthonormal basis of a random vector with elements drawn from a uniform (0,1) distribution subject to the constraint that the solution $\mathbf{x}^{\star} = M^{-1}\mathbf{y}^{\star}$ must be a specified distance $\delta = 0.05$ from the initial conditions, i.e. $|x_i^{\star} - x_{0,i}| > \delta \; \forall \; i$.

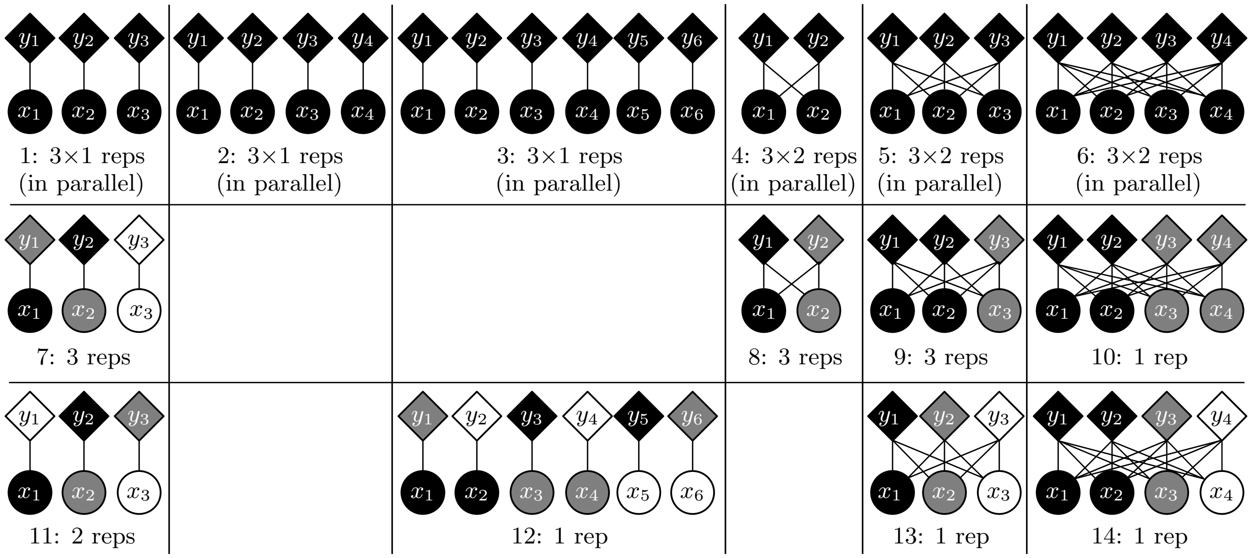

Figure 3-4: Graph representation of the 14 design tasks and 42 total replications with input/output assignments identified by color.

The resulting target has a Euclidean norm of 1 (i.e. $||\mathbf{y}^\star|| = 1$) to provide a standard distance between initial and final values for all tasks and requires a minimum change of δ for each input to achieve the zero-error solution.

3.3.2 Experimental Procedure

Six groups of three subjects participated in this study under an IRB-approved protocol. Volunteers were recruited from email solicitation and a convenience sample of graduate programs at MIT and were not paid for their efforts. Subjects were predominately male (72.2%) and 25–29 years of age (50.0%) with more college education than work experience in technical fields. Most subjects had never interacted with each other in the past (47.2% of pairs). Table 3.2 summarizes the complete subject demographics.

Design sessions are scheduled when three volunteers are available to form ad-hoc teams. Thus, while there is not random assignment of subjects to sessions, there is also no purposeful selection for sessions. All experiments are conducted in university classrooms using wireless network connections. At the start of the session subjects are assigned a color (red, green, or blue) and are seated on one side of a four-seat rectangular designer table with the fourth seat reserved for the administrator. The table is arranged such that each computer display is only visible to the individual seated at the corresponding seat.

Each experimental session is conducted using a standard procedure. Participants may exit the study at any point, however no such events occurred. A scripted presentation introduces the experimental objectives of studying collaborative design in coupled problems using software-assisted tools and issues consent forms and a questionnaire to all subjects. Next, a series of five training tasks introduce subjects to the software and design process. The training tasks are identical in structure to experimental tasks and include three individual tasks (in parallel) with one uncoupled, two

Table 3.2: Collaborative design subject demographics in six sessions

Category	Value	Count	(%)
Gender	Male	13	72.2
	Female	5	27.8
Age	18–24	6	33.3
	25–29	9	50.0
	30–34	2	11.1
	35–39	1	5.6
	40–49	0	0.0
	50+	0	0.0
Years of college education in a technical field	0	0	0.0
	1–2	0	0.0
	3–4	1	5.6
	5–6	6	33.3
	7–8	7	38.9
	9+	4	22.2
Years of professional work experience in a technical field	0	5	27.8
	1–2	10	55.6
	3–4	1	5.6
	5–6	0	0.0
	7–8	1	5.6
	9+	1	5.6
Frequency of past interactions with other subjects	Never	17	47.2
	Once	2	5.6
	Rarely	9	25.0
	Monthly	4	11.1
	Weekly	4	11.1
	Daily	0	0.0

uncoupled, and two coupled variables and two team tasks with three uncoupled and three coupled variables. While completing the training tasks the administrator explains the software interface, design objectives, and communications limitations of verbal conversation and/or gestures. The five training tasks take approximately 15 minutes to complete.

After completing the training design tasks, the 24 experimental design tasks are conducted in randomized order with a constraint that no coupled four-variable design problems (tasks 6, 10, and 14) can occur within the first ten tasks. This constraint acknowledges learning effects to avoid conditions observed in pilot sessions where some subjects could not solve large design problems early in the session. There is no time limit on solving each task, though participants are instructed the expected time to complete all 24 tasks is 60 minutes. During the experimental tasks, all designer input modifications are automatically logged to file for post-processing and a screen recording program captures the administrator's display and records audio. After each task the subjects are allowed time before the next task begins, which is manually controlled by the administrator. Finally, following completion of the final task, the participants are issued a second questionnaire.

3.3.3 Limitations and Threats to Validity

The study design incorporates several limitations which introduce threats to the validity of results. First, the decision to include unique design tasks for each session potentially introduces additional variance in the results. By virtue of the random generation of M matrices and $\mathbf{y}^\star$ targets, some tasks may be easier to solve in some sessions than others. Generally, if coupling factors are close to zero, the problems are less coupled and easier to solve. While using the same set of tasks across all sessions would reduce this error, it also would introduce a wider bias to the specific set of tasks selected for all sessions.

The restriction preventing the four-variable design tasks within the first ten tasks likely biases the results due to learning effects. While a practical consideration due to limited time for training, the impact of only completing the most difficult tasks in the second half of the study almost certainly causes the time to complete these tasks to be biased low. More broadly, the learning effect is also expected for other tasks. While the results from a particular session are subject to learning effects, the results across sessions are mitigated by the randomized task ordering.

The pair tasks (tasks 7–11) are imperfect in a three-subject session. Task 7 studies an uncoupled pair design task, however a third designer simultaneously completes an independent task. Tasks 8–10 provide a blank interface for the third designer. In both cases, there are likely bias effects of having the third subject in the room and/or completing an independent task, if only the communication required to establish the third designer is not a participant in the task at hand. One would expect the time to complete these tasks to be biased high.

Finally, there are limitations to the generalizability of results. As previously discussed, the surrogate design task is simplified to provide experimental control. Real design tasks are likely non-linear, much larger (more variables), context-rich, and may have no or many solutions. Ad-hoc team formation is also limited as real design tasks include specialized designers who may have significant working experience together. The selection frame of graduate students is somewhat limited as few have significant professional work experience. This effect is somewhat limited, however, by significant experience with technical problems. There may also be selection bias of the volunteers responding to email and convenience solicitation.

3.4 Results and Analysis

Table 3.3 summarizes raw results from the experimental sessions. Appendix A provides detailed raw results from each task. A few data points were removed from the study. In two sessions, one replication of task 6 was eliminated to meet scheduled time constraints. This may bias results due to selection effects, but is moderated by the large number of samples retained. In another session, task 3 was removed due to network latency issues experienced during the task. This is not expected to have a large impact on the results.

Figure 3-5 illustrates data from a typical coupled individual design problem (task 4). Each designer controls two inputs and has two output objectives. Times are relative to the first input change for each designer such that all start at $t = 0$. Note the red designer uses large input changes (page-up, page-down key presses) compared to the green and blue designers (arrow key presses). The "initial input overshoot" behavior observed in the red and green designers is common for coupled problems. The error plots below are post-processed using the Euclidean norm of the difference between output and target vectors to illustrate progress towards finding the solution.

Figure 3-6 illustrates data from a typical coupled triad design problem (task 14). Here, the red designer controls two inputs and outputs while the other designers only control one. The "one factor at a time" process of changing inputs is common in team tasks, especially while rotating between each designer's inputs. Gaps in changes, e.g. around 400 seconds, correspond to team discussion. The error plot at the bottom is post-processed using the Euclidean norm of the difference between output and target vector. It illustrates initial progress, followed by regression (often as designers reset their inputs to initial conditions), followed by a slow iteration to a valid solution.

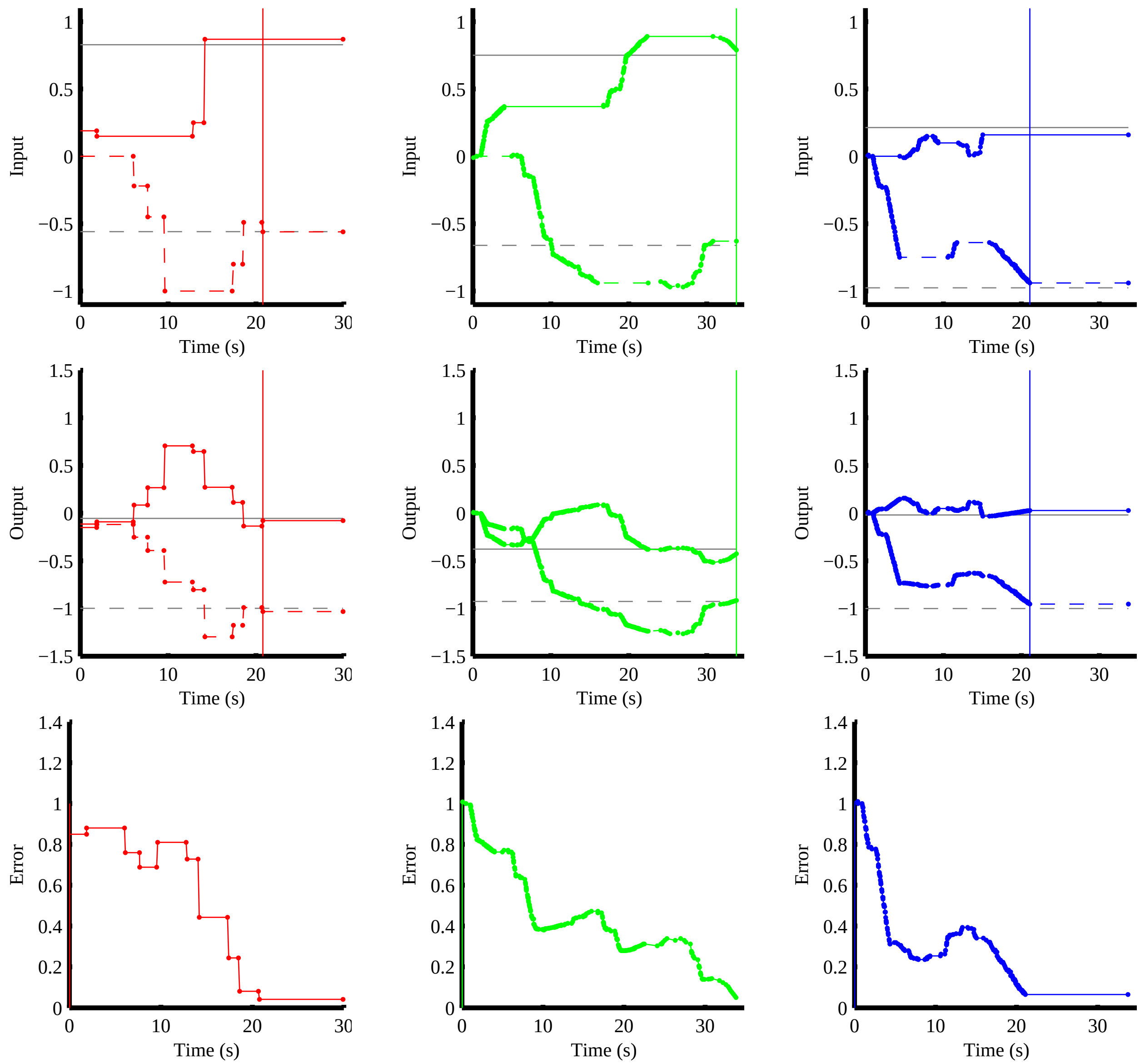

Figure 3-5: Typical input and output data for solving a coupled 2 × 2 individual design task (Task 4). Each designer (red, green, blue) has two inputs to achieve two target outputs. Times are relative to the time of the first input change for each designer. Computed zero-error solution values are shown in gray lines. Vertical lines show when a valid solution was achieved.

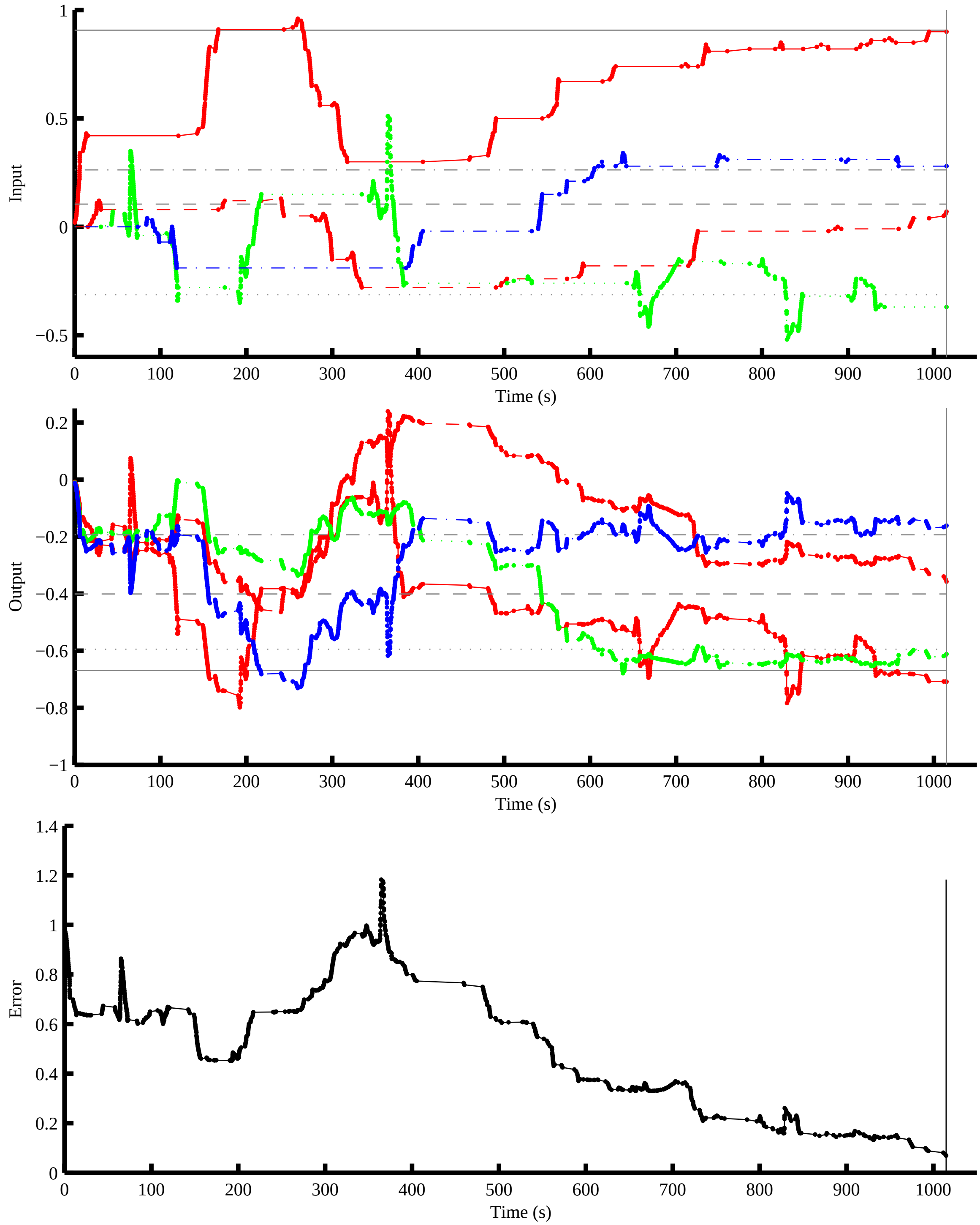

Figure 3-6: Typical input and output data for solving a coupled 4 × 4 triad design task (Task 14). Times are relative to the time of the first input change. Computed zero-error solution values are shown in gray lines. Error data is post-processed using a standard distance norm.

Table 3.3: Summary of raw experimental results

Task j	n	N	Coupled (C) or Uncoupled (U)	Samples Retained	Samples Removed	Raw Percentile Times (s) 25th	Median	75th	Variance (s^2)
1	1	3	U	18	0	12.3	15.1	20.1	57.6
2	1	4	U	18	0	11.5	16.2	26.0	103.4
3	1	6	U	15	3	25.3	33.1	46.3	151.3
4	1	2	C	36	0	9.6	14.4	21.0	76.7
5	1	3	C	36	0	32.0	42.9	58.9	1755.8
6	1	4	C	30	6	72.1	125.5	165.7	5713.3
Subtotal Individual				153	9	162.7	247.2	337.9	7858.2
7	2	3	U	18	0	29.4	34.2	45.0	123.0
8	2	2	C	18	0	21.1	30.6	54.5	345.2
9	2	3	C	18	0	91.4	127.8	228.7	12368.1
10	2	4	C	6	0	114.2	270.5	374.7	21084.3
11	3	3	U	12	0	30.2	39.1	64.0	465.5
12	3	6	U	6	0	73.9	99.9	130.8	1277.8
13	3	3	C	6	0	123.6	297.0	530.8	63580.9
14	3	4	C	6	0	167.3	641.8	792.3	114946.3
Subtotal Team				90	0	651.1	1540.9	2220.8	214191.1

3.4.1 Normalization Procedure

Raw results exhibit large variation in individual and team performance to solve similar tasks. A portion of the variance may arise from differing abilities to solve the surrogate design task which is not the subject of this study. A normalization procedure is applied in the analysis to moderate this effect of baseline ability. It assumes each unit (individual or group) has some inherent capability C which linearly scales the completion time to a normalized time. A value $C < 1$ indicates inferior performance compared to others and a value $C > 1$ indicates superior performance. Two procedures are discussed with discussion to their relative strengths and weaknesses.

Procedure 1: Single Task Type

The first normalization procedure defines a capability factor C_i based on a unit of analysis i's completion time across tasks J_i and replications K_{ij} compared to the median task completion time $\bar{t}_j$ across the same set of tasks and replications for all comparable units. Units of analysis include individuals (for $n = 1$ tasks) and teams (for $n > 1$ tasks). The normalized task completion time t'_{ijk} for unit i, task j, and replication k is defined by

$$t'_{ijk} = t_{ijk} \cdot C_i \tag{3.13}$$

where

$$C_i = \frac{\sum_{j \in J_i} \sum_{k \in K_{ij}} \bar{t}_j}{\sum_{j \in J_i} \sum_{k \in K_{ij}} t_{ijk}} \tag{3.14}$$

and

$$\bar{t}_j = \text{median}_{ik}(t_{ijk}). \tag{3.15}$$

This formulation can accommodate missing or removed data points as the j index iterates across the completed set of design tasks J_i and the k index iterates across the completed set of replications K_{ij} for unit i and task j.

Although it is possible to define units for individuals, pairs, and triads, this approach is not taken due to potential bias and over-fitting. As the pair and triad tasks are conducted in a similar setting with potential confounding factors, the two units are coalesced into the group unit of analysis. Future studies could study these units with more control to potentially reduce variance in these tasks. In particular, this approach require additional replications with a pair of individuals.

This normalization procedure is biased towards longer-duration tasks. An alternative metric definition in Eq. 3.16 has no bias to task duration, however it also conflicts with the with the assumption that capability linearly scales the magnitude of task completion time. In other words, the capability effect is most visible during longer tasks and the metric should capture this effect.

$$C'_i = \frac{\sum_{j \in J_i} \sum_{k \in K_{ij}} 1}{\sum_{j \in J_i} \sum_{k \in K_{ij}} t_{ijk}/\bar{t}_j} \tag{3.16}$$

Finally, this procedure differs from Hirschi and Frey (2002) which normalizes task times by the time to complete one particular $N = 2$ coupled task solved at a random point during the study. Their approach has two main limitations. First, it amplifies noise in a single data point from an individual's task completion time which may have considerable variance. Second, it produces a small signal by relying on a relatively simple task with a short task duration. The proposed procedures use the relatively large contribution of long-duration tasks for a stronger signal and leverage all available data points to reduce noise.

Figure 3-7 summarizes the normalization factors computed using procedure 1 which are detailed in Appendix A. Individual capability factors range from 0.61 to 1.57 with a median of 0.81. Team capability factors range from 0.78 to 1.45 with a median of 0.90. A summary of the normalized completion times is shown in Table 3.4.

Procedure 2: Multiple Task Types

Recognizing that inherent capability may vary across different task types, procedure 2 adds an additional constraint on the set of tasks J_{im} to restrict to a task type m for coupled and uncoupled tasks. For example, the uncoupled factor may measure the unit's capability of using the computer interface while the coupled factor may measure the unit's capability of solving the more difficult tasks. Thus, for each unit i there are two normalization factors: one for coupled tasks ($C_{i,C}$) and one for uncoupled tasks ($C_{i,U}$). All equations remain the same except substituting J_{im} for J_i in Eq. 3.14.

Figure 3-8 summarizes the normalization factors computed using procedure 2 which are detailed in Appendix A. Individual coupled factors range from 0.62 to 1.64 with a median of 0.86. Individual uncoupled factors range from 0.48 to 1.98 with a median of 0.87. Team coupled factors range from

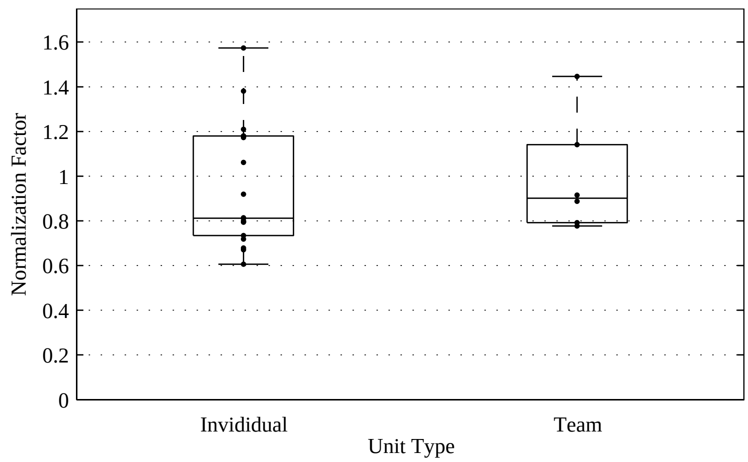

Figure 3-7: Box plot of individual and team capability factors C_i using normalization procedure 1. Boxes bound the first and third quartiles and whiskers bound extremes within 1.5 times the interquartile range.

Table 3.4: Summary of normalized experimental results using procedure 1

Task j	n	N	Coupled / Uncoupled	Normalized Percentile Times (s) 25th	Median	75th	Variance (s^2)	Relative Variance γ_j
1	1	3	U	12.1	14.8	20.0	32.5	0.56
2	1	4	U	12.2	15.7	21.0	69.8	0.68
3	1	6	U	19.7	32.6	39.1	179.5	1.19
4	1	2	C	8.8	14.4	21.9	73.6	0.96
5	1	3	C	27.9	42.1	57.3	1235.1	0.70
6	1	4	C	75.1	100.3	140.3	2312.1	0.40
Subtotal Individual				155.8	219.9	299.5	3902.7	0.50†
7	2	3	U	26.2	38.0	42.2	111.4	0.91
8	2	2	C	19.1	29.0	44.3	486.2	1.41
9	2	3	C	82.2	139.0	185.0	12494.6	1.01
10	2	4	C	146.9	229.0	360.2	14656.3	0.70
11	3	3	U	28.6	36.0	60.8	921.9	1.98
12	3	6	U	59.9	97.5	143.1	2836.7	2.22
13	3	3	C	131.5	292.5	565.0	60462.5	0.95
14	3	4	C	232.1	564.4	645.2	56275.5	0.49
Subtotal Team				726.5	1425.5	2045.9	148245.2	0.69†

† Relative to raw subtotal variance

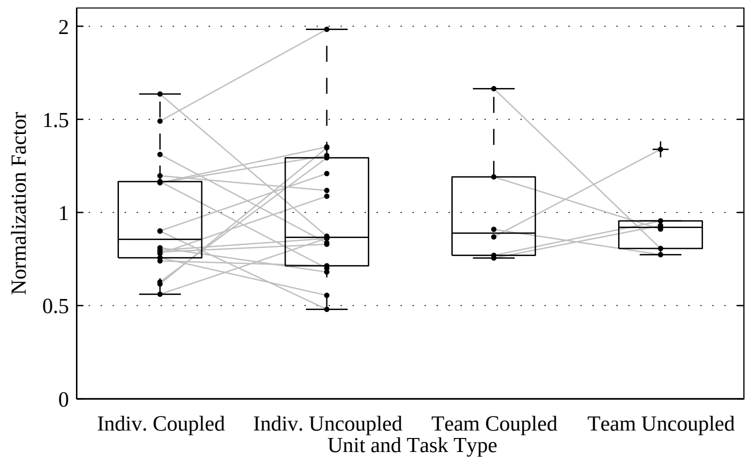

Figure 3-8: Box plot of individual and team capability factors C_{im} using normalization procedure 2. Boxes bound the first and third quartiles and whiskers bound extremes within 1.5 times the interquartile range. Gray lines connect values of specific individuals and teams.

0.76 to 1.67 with a median of 0.89. Team uncoupled factors range from 0.77 to 1.34 with a median of 0.92. As illustrated by the gray lines, there is little correlation between a unit's coupled and uncoupled factor, suggesting it is a valid approach. A summary of the normalized times is shown in Table 3.5.

Comparison of Normalized Results

Normalization attempts to reduce unwanted variance, specifically that arising from differences in inherent capabilities. The relative variance metric γ_j is defined in Eq. 3.17 to compare the variance of task j times under a normalization procedure to that of the raw results. A desirable value of $\gamma_j < 1$ indicates the normalization reduced variance. Relative variance metrics are illustrated in Tables 3.4 and 3.5.

$$\gamma_j = \frac{\text{var}_{ik}(t'_{ijk})}{\text{var}_{ik}(t_{ijk})} \tag{3.17}$$

Normalization procedure 1 reduces the sum of individual design task variances to 50% of the raw value. All except task 3 experience a reduction in variance, with the largest decrease in task 6 to 40% of the raw value. The procedure also reduces the sum of team design task variances to 69%. Four tasks experience a reduction in variance with the largest decrease in task 14 to 49%. Four other tasks experience an increase in variance. Two tasks (11 and 12) in particular increase by 98% and 122% respectively. The results suggest normalization procedure 1 does not adequately capture effects governing these tasks.

Normalization procedure 2 reduces the sum of individual design task variances to 44% of the raw value. All tasks experience a reduction in variance, with large decreases in tasks 1, 2, and 3 to 12%, 16%, and 15% respectively. The procedure also reduces the sum of team design task variances to 66%. Five tasks experience a reduction in variance with the largest decrease in task 12 to 30%. Three other tasks experience an increase in variance. Task 8 in particular experiences

Table 3.5: Summary of normalized experimental results using procedure 2

Task j	n	N	Coupled / Uncoupled	Normalized Percentile Times (s)			Variance (s^2)	Relative Variance γ_j
				25th	Median	75th		
1	1	3	U	13.5	15.2	17.3	7.1	0.12
2	1	4	U	14.1	16.4	20.2	16.4	0.16
3	1	6	U	28.0	31.1	36.8	23.0	0.15
4	1	2	C	8.9	14.7	21.7	76.1	0.99
5	1	3	C	27.2	44.2	60.8	1217.6	0.69
6	1	4	C	83.0	99.0	134.3	2130.9	0.37
Subtotal Individual				174.8	220.7	291.1	3471.1	0.44†
7	2	3	U	26.9	31.0	45.9	156.7	1.27
8	2	2	C	19.2	29.7	43.1	603.9	1.75
9	2	3	C	87.9	147.7	184.6	13440.1	1.09
10	2	4	C	149.1	238.8	357.6	14924.5	0.71
11	3	3	U	28.4	39.2	56.3	308.5	0.66
12	3	6	U	74.9	90.9	105.8	383.2	0.30
13	3	3	C	135.9	294.9	614.5	62148.2	0.98
14	3	4	C	263.3	560.4	622.3	48514.8	0.42
Subtotal Team				785.6	1432.6	2030.1	140479.9	0.66†

† Relative to raw subtotal variance

Table 3.6: Summary of uncoupled individual task regression model results

Variable	Model U-I-1		
	Coef.	SE	t stat.
Constant	-3.43	2.16	-1.59
N	5.71	0.49	11.66^{**}
d.f.		49	
R^2		0.74	
F stat.		136.03^{**}	

* $p < 0.05$
** $p < 0.01$

a 75% increase.

In comparing the two procedures, procedure 2 produces lower variances on a whole for both individual and group tasks. All individual tasks under procedure 2 have lower variances than those in procedure 1, but only three team tasks have lower variance compared to five with higher variance. The magnitude of changes, however, are more significant for tasks with lower variance than those with higher variance to establish preference for procedure 2.

Finally, one must justify the value of using twice as many parameters in procedure 2 to avoid over-fitting. Coupled and uncoupled tasks require fundamentally different solution modes. Uncoupled tasks are largely an exercise in computer-human interaction where coupled tasks are a mentally-challenging activity. The results show benefit of this distinction as all three individual uncoupled tasks and two of the three group uncoupled tasks experienced large reductions in variance under procedure 2 as compared to procedure 1. Based on these advantages, the remainder of the analysis applies procedure 2 to normalize task times.

3.4.2 Uncoupled Task Analysis

This section analyzes uncoupled tasks across individual, pairs, and triads of designers. Hirschi and Frey (2002) report the time to complete individual uncoupled tasks varies linearly in the problem size N. This hypothesis is expressed in Model U-I-1 with the equation

$$t'(N) = b_0 + b_1 \cdot N. \tag{3.18}$$

Table 3.6 summarizes a linear regression analysis of Model U-I-1, finding the b_1 parameter to be highly significant, $t(49) = 11.66, p < 0.001$. Its value of 5.71 can be interpreted as the additional normalized time to complete a task with an extra uncoupled variable. The constant factor b_0 is small and not significant, $t(49) = -1.59, p = 0.118$. Graphical inspection of model residuals indicate they are representative of a normal distribution. The overall model is highly significant, $F(1, 49) = 136.03, p < 0.001$, explaining about 74% of the variance in the data. Figure 3-9 overlays the regression model on a box plot of individual uncoupled task times to illustrate goodness of fit.

Figure 3-10 illustrates box plots of pair and triad tasks. Regression analysis is not performed for pairs and triads alone due to the limited data points (two and one value of N, respectively).

Uncoupled task completion time is expected to be a function of problem size N and team size

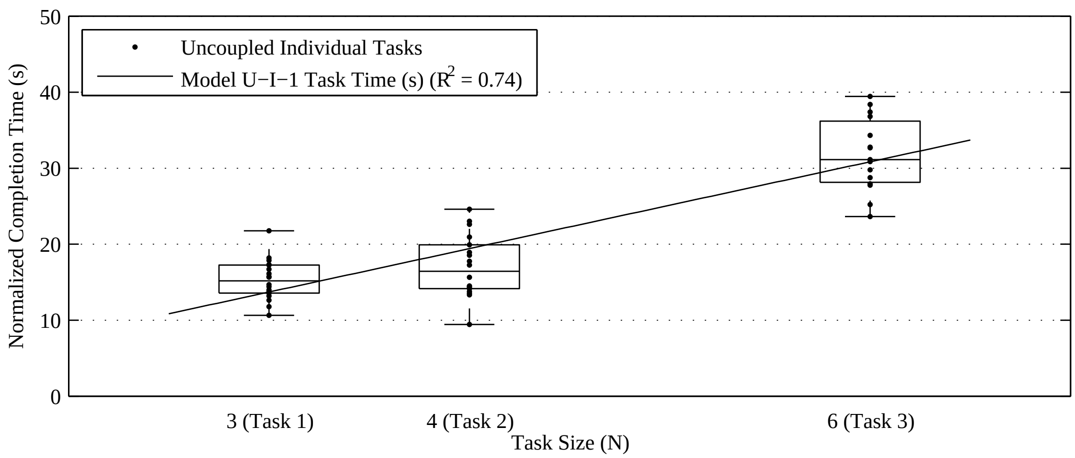

Figure 3-9: Box plot of normalized task times for uncoupled individual tasks with overlay of results from Model U-I-1. Boxes bound the first and third quartiles and whiskers bound extremes within 1.5 times the interquartile range.

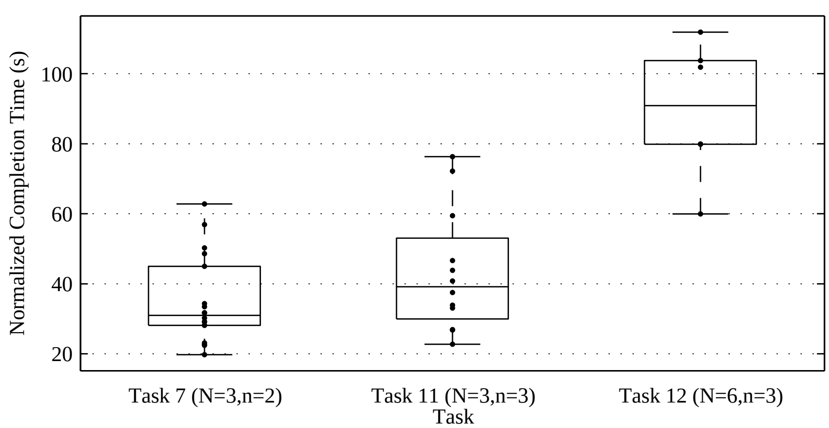

Figure 3-10: Box plots of coupled pair and triad tasks. Boxes bound the first and third quartiles and whiskers bound extremes within 1.5 times the interquartile range.

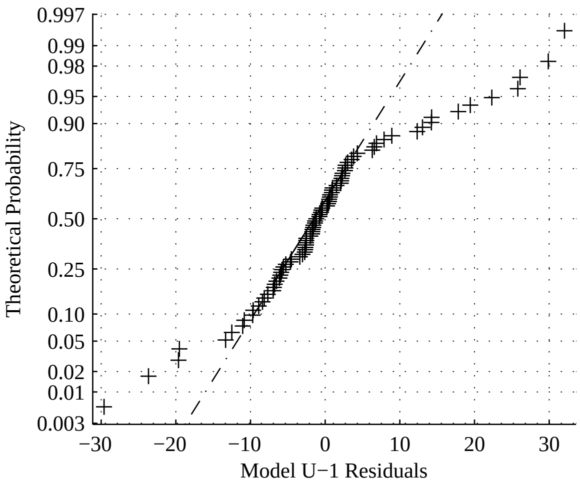

Figure 3-11: Normal probability plot of Model U-1 residuals. The S-curved tails indicate a non-normal distribution.

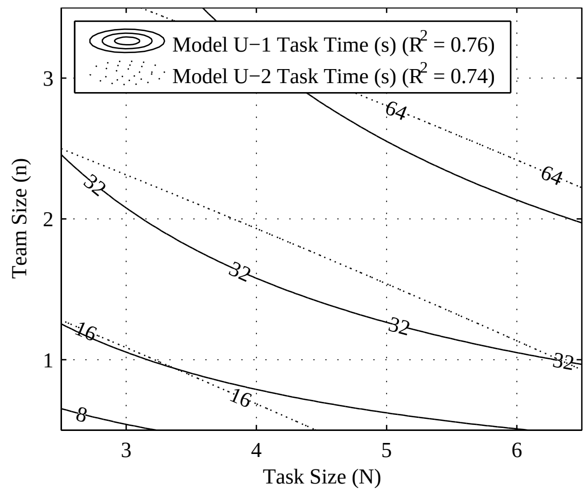

Figure 3-12: Contour plot of normalized uncoupled task times in seconds under Models U-1 and U-2. Note log-scale contours.

n. Model U-1 hypothesizes a functional form of

$$t'(N,n) = b_0 + b_1 N + b_2 n + b_3 N \cdot n \tag{3.19}$$

where the interaction term coefficient b_3 is interpreted as the effect of team size amplifying the challenge of problem size. Table 3.7 summarizes a linear regression analysis of the hypothesized model, finding the b_3 interaction parameter significant, $t(83) = 4.33, p < 0.001$. Graphical inspection of the residuals in Fig. 3-11 suggest they do not follow a normal distribution, thus limiting the use of Model U-1 despite its fairly good fit.

As an attempt to improve the residual distribution, Model U-2 log-transforms the task completion times to hypothesize a functional form of

$$\begin{aligned} \ln\left(t'(N,n)\right) &= b_0 + b_1 N + b_2 n + b_3 N \cdot n \\ \iff t'(N,n) &= \exp\left(b_0 + b_1 N + b_2 n + b_3 N \cdot n\right). \end{aligned} \tag{3.20}$$

While several coefficients are again significant in Table 3.7, graphical inspection of the residuals still suggest a non-normal distribution.

Model U-3 hypothesizes an alternative functional form using inverse team size $1/n$:

$$\begin{aligned} \ln\left(t'(N,n)\right) &= b_0 + b_1 N + b_2 \frac{1}{n} + b_3 \frac{N}{n} \\ \iff t'(N,n) &= \exp\left(b_0 + b_1 N + b_2 \frac{1}{n} + b_3 \frac{N}{n}\right). \end{aligned} \tag{3.21}$$

By virtue of the log transform, the inverse team size factor with a negative coefficient produces an upper bound on task time with increasing team size. Regression analysis finds coefficients b_0, b_1, and b_2 to be highly significant, $t(83) = 13.96, p < 0.001, t(83) = 3.78, p < 0.001, t(83) = -5.03, p < 0.001$ respectively. The interaction coefficient b_3 is small and not significant $t(83) = 0.10, p = 0.92$.

Table 3.7: Summary of uncoupled task regression model results

Variable	Model U-1			Model U-2		
	Coef.	SE	t stat.	Coef.	SE	t stat.
Constant	-1.89	8.86	-0.21	1.47	0.25	5.97**
N	0.48	2.03	0.24	0.23	0.06	4.12**
n	1.73	4.56	0.38	0.59	0.13	4.67**
$N \cdot n$	4.63	1.07	4.33**	-0.01	0.03	-0.28
d.f.		83			83	
R^2		0.76			0.74	
F stat.		88.94**			78.35**	

* $p < 0.05$
** $p < 0.01$

Table 3.8: Summary of uncoupled task regression model results (continued)

Variable	Model U-3			Model U-4		
	Coef.	SE	t stat.	Coef.	SE	t stat.
Constant	3.54	0.25	13.96**	3.52	0.11	32.04**
N	0.24	0.06	3.78**	0.25	0.02	10.40**
$1/n$	-1.64	0.33	-5.03**	-1.61	0.10	-15.97**
N/n	0.01	0.08	0.10			
d.f.		83			84	
R^2		0.78			0.78	
F stat.		99.03**			150.32**	

* $p < 0.05$
** $p < 0.01$

Graphical inspection of the residuals suggest a normal distribution. The overall model is highly significant, $F(2, 83) = 99.03, p < 0.001$, explaining about 78% of the variance in task times.

Model U-4 removes the insignificant interaction parameter to hypothesize a functional form of

$$\ln\left(t'(N,n)\right) = b_0 + b_1 N + b_2 \frac{1}{n} \iff t'(N,n) = \exp\left(b_0 + b_1 N + b_2 \frac{1}{n}\right). \qquad (3.22)$$

Regression analysis finds all three coefficients b_0, b_1, b_2 to be highly significant, $t(84) = 32.04, p < 0.001$, $t(84) = 10.40, p < 0.001$, $t(84) = -15.97, p < 0.001$ respectively. Graphical inspection of the residuals suggest a normal distribution. The overall model is also highly significant, $F(2, 84) = 150.32, p < 0.001$, explaining about 78% of task time variance. Figure 3-13 illustrates Models U-3 and U-4 as a contour plot of expected task completion times.

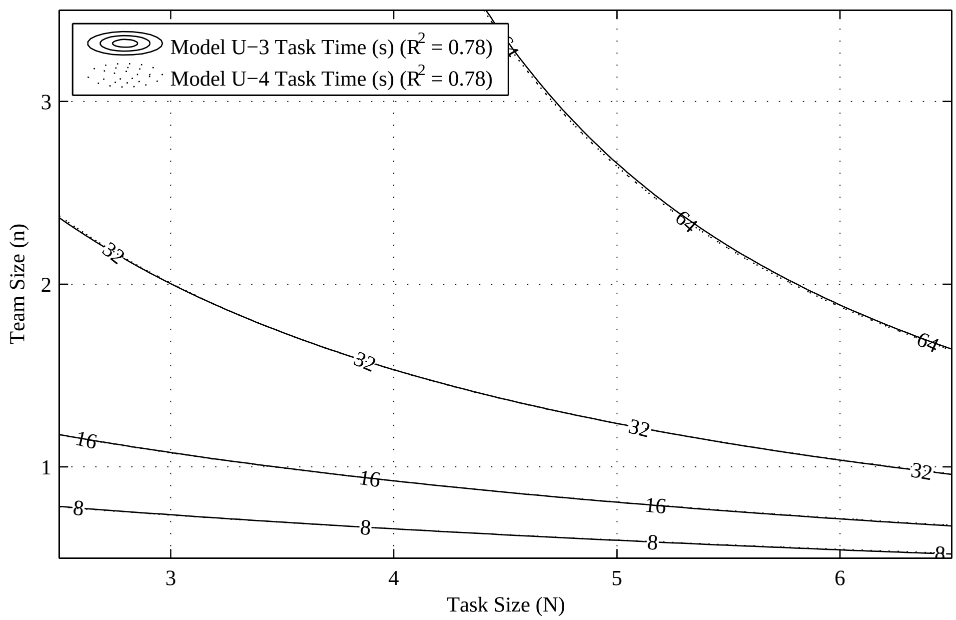

Figure 3-13: Contour plot of normalized uncoupled task times in seconds under Models U-3 and U-4. Note log-scale contours.

3.4.3 Coupled Task Analysis

This section analyzes coupled tasks across individual, pairs, and triads of designers. Hirschi and Frey (2002) report the time to complete individual coupled tasks varies geometrically in the problem size N. This hypothesis is expressed in Model C-I-1 with the equation

$$\ln\left(t'(N)\right) = b_0 + b_1 \cdot N \iff t'(N) = \exp\left(b_0 + b_1 \cdot N\right). \tag{3.23}$$

Table 3.9 summarizes a linear regression analysis of Model C-I-1, finding both the b_0 and b_1 parameters to be highly significant, $t(100) = 2.90, p = 0.005$ and $t(100) = 15.71, p < 0.001$. The value $e^{b_1} = 2.81$ can be interpreted as the multiplicative factor on normalized time to complete a task with an additional coupled variable. Graphical inspection of model residuals indicate they are representative of a normal distribution. The overall model is highly significant, $F(1, 100) = 246.89, p < 0.001$, explaining about 71% of the variance in the data. Figure 3-9 overlays the regression model on a box plot of individual uncoupled task times, illustrating the goodness of fit.

Pair tasks are analyzed using a similar regression analysis as in the validation of coupled individual tasks. Model C-P-1 uses the functional form in Eq. 3.23. Table 3.9 summarizes the regression analysis. Both model parameters b_0 and b_1 are highly significant, $t(40) = 3.85, p < 0.001$ and $t(40) = 8.15, p < 0.001$ respectively. Graphical inspection of model residuals are representative of a normal distribution. The overall model is also highly significant, $F(1, 37) = 66.35, p < 0.001$, explaining about 62% of the variance in the data. Figure 3-15 overlays the regression model on a box plot of the coupled pair task times, illustrating the goodness of fit.

Regression analysis was not performed for the coupled triad tasks due to the limited data points

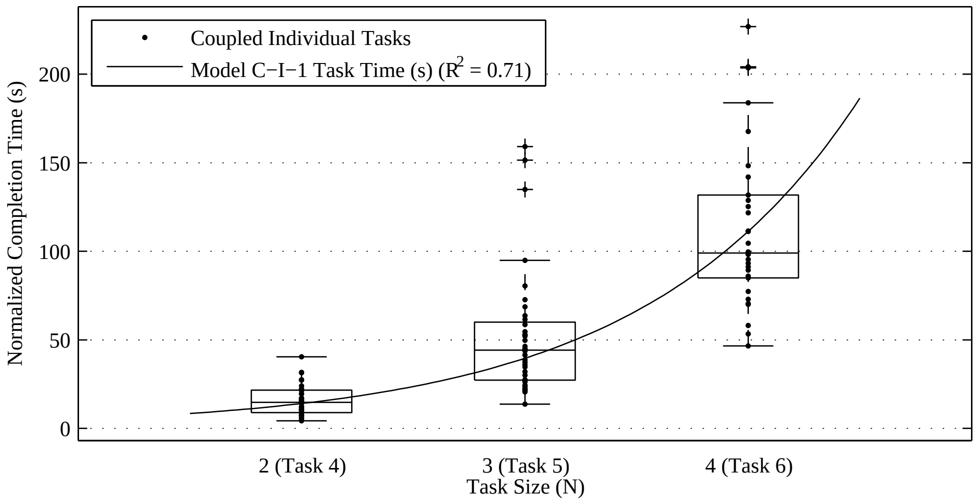

Figure 3-14: Box plot of normalized task times for coupled individual tasks with overlay of results from Model C-I-1. Boxes bound the first and third quartiles and whiskers bound extremes within 1.5 times the interquartile range.

Table 3.9: Summary of coupled individual and pair task regression model results

Variable	Model C-I-1			Model C-P-1		
	Coef.	SE	t stat.	Coef.	SE	t stat.
Constant	0.58	0.20	2.90^{**}	1.43	0.37	3.85^{**}
N	1.03	0.07	15.71^{**}	1.08	0.13	8.15^{**}
d.f.		100			40	
R^2		0.71			0.62	
F stat.		246.89^{**}			66.35^{**}	

$^{*}\ p < 0.05$
$^{**}\ p < 0.01$

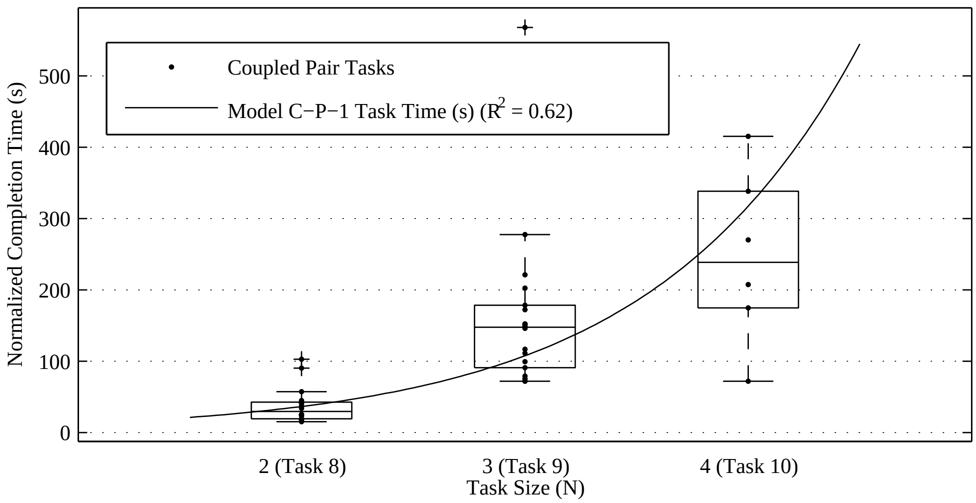

Figure 3-15: Box plot of normalized task times for coupled pair tasks with overlay of results from Model C-P-1. Boxes bound the first and third quartiles and whiskers bound extremes within 1.5 times the interquartile range.

(two values of N), however Fig. 3-16 illustrates the data in a box plot.

Coupled task completion time is generally expected to be a function of problem size N and team size n. Model C-1 hypothesizes a functional form of

$$\begin{aligned} \ln t'(N,n) &= b_0 + b_1 N + b_2 n + b_3 (N \cdot n) \\ \iff t'(N,n) &= \exp\left(b_0 + b_1 N + b_2 n + b_3 (N \cdot n)\right) \end{aligned} \tag{3.24}$$

where the interaction coefficient b_3 is interpreted as team size amplifying the effect of problem size. Table 3.10 summarizes the regression analysis. Coefficients b_1 and b_2 are highly significant, $t(152) = 7.99, p < 0.001$ and $t(152) = 4.01, p < 0.001$ respectively. The constant and interaction coefficients are not significant, $t(152) = -1.63, p = 0.106, t(152) = -1.30, p = 0.194$ respectively. Graphical inspection of model residuals indicate they are representative of a normal distribution. The overall model is highly significant, $F(3, 152) = 159.18, p < 0.001$, explaining about 76% of the variance in the task times.

Model C-2 removes the insignificant interaction parameter to hypothesize a functional form of

$$\ln t'(N,n) = b_0 + b_1 N + b_2 n \iff t'(N,n) = \exp\left(b_0 + b_1 N + b_2 n\right). \tag{3.25}$$

Coefficients b_1 and b_2 are highly significant, $t(153) = 18.40, p < 0.001$ and $t(153) = 13.46, p < 0.001$ respectively. Graphical inspection of model residuals indicate they are representative of a normal distribution. The overall model is highly significant, $F(2, 153) = 236.84, p < 0.001$, explaining about 76% of the variance in the task times. Figure 3-17 illustrates a contour plot of expected task completion times comparing Models C-1 and C-2.

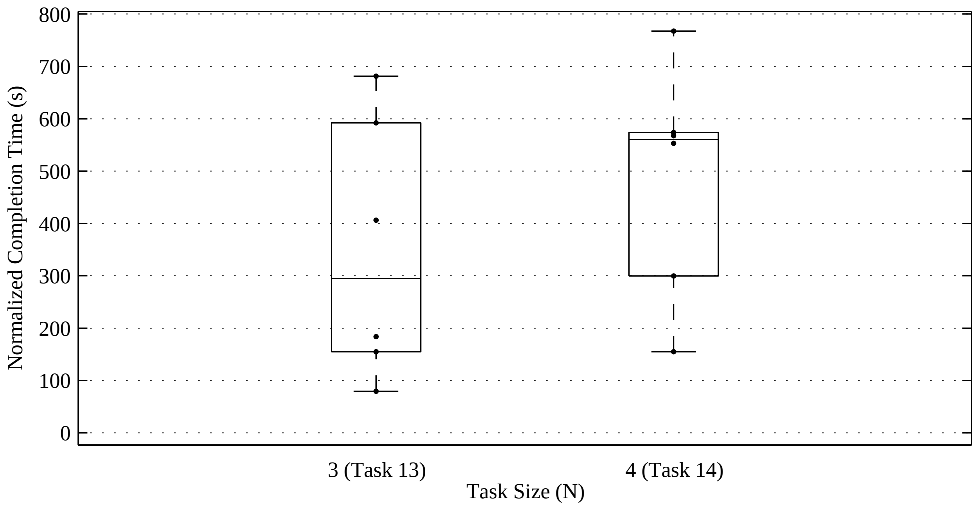

Figure 3-16: Box plot of normalized task times with the regression model for coupled triad tasks. Boxes bound the first and third quartiles and whiskers bound extremes within 1.5 times the interquartile range.

Table 3.10: Summary of coupled task regression model results

Variable	Model C-1			Model C-2		
	Coef.	SE	t stat.	Coef.	SE	t stat.
Constant	-0.77	0.47	-1.63	-0.21	0.20	-1.05
N	1.18	0.15	7.99^{**}	1.00	0.05	18.40^{**}
n	1.29	0.32	4.01^{**}	0.89	0.07	13.46^{**}
$N \cdot n$	-0.13	0.10-1.30				
d.f.		152			153	
R^2		0.76			0.76	
F stat.		159.18^{**}			236.84^{**}	

$^{*}\ p < 0.05$
$^{**}\ p < 0.01$

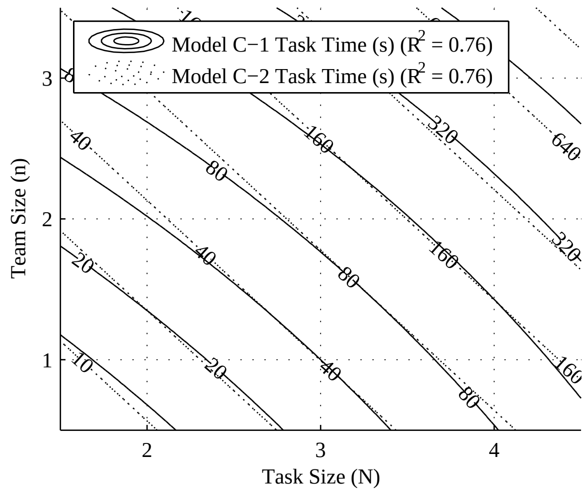

Figure 3-17: Contour plot of normalized coupled task times under Models C-1 and C-2. Note log-scale contours.

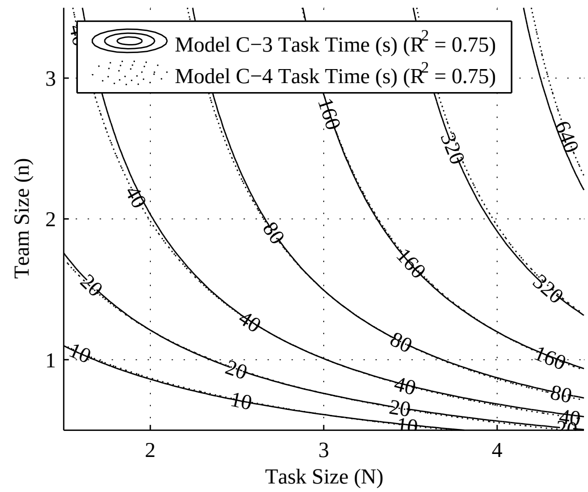

Figure 3-18: Contour plot of normalized coupled task times under Models C-3 and C-4. Note log-scale contours.

As an attempt to adopt the form of Model U-3, Model C-3 hypothesizes a functional form of

$$\ln t'(N,n) = b_0 + b_1 N + b_2 \frac{1}{n} + b_3 \frac{N}{n}$$
$$\iff t'(N,n) = \exp\left(b_0 + b_1 N + b_2 \frac{1}{n} + b_3 \frac{N}{n}\right). \tag{3.26}$$

Table 3.11 summarizes the regression analysis. Coefficients b_0, b_1, and b_2 are highly significant, $t(152) = 3.95, p < 0.001$, $t(152) = 5.46, p < 0.001$, and $t(152) = -2.65, p = 0.009$ respectively. Graphical inspection of model residuals indicate they are representative of a normal distribution. The overall model is highly significant, $F(3,152) = 156.04, p < 0.001$, explaining about 75% of the variance in the task times.

Model C-4 removes the insignificant interaction parameter to hypothesize a functional form of

$$\ln t'(N,n) = b_0 + b_1 N + b_2 \frac{1}{n} \iff t'(N,n) = \exp\left(b_0 + b_1 N + b_2 \frac{1}{n}\right). \tag{3.27}$$

All coefficients b_0, b_1, and b_2 are highly significant, $t(153) = 11.67, p < 0.001$, $t(153) = 17.89, p < 0.001$, and $t(153) = -12.14, p < 0.001$ respectively. Graphical inspection of model residuals indicate they are representative of a normal distribution. The overall model is highly significant, $F(2,153) = 235.38, p < 0.001$, explaining about 75% of the variance in the task times. Figure 3-18 shows a contour plot of task completion times comparing Models C-3 and C-4.

3.5 Discussion

This section discusses the results of the regression analysis in the context of past work and considers implications of the results in collaborative design. For convenience, Table 3.12 summarizes the results for each of the regression models analyzed in the previous section.

Table 3.11: Summary of coupled task regression model results (continued)

Variable	Model C-3			Model C-4		
	Coef.	SE	t stat.	Coef.	SE	t stat.
Constant	2.48	0.63	3.95**	2.67	0.23	11.67**
N	1.11	0.20	5.46**	1.05	0.06	17.89**
$1/n$	-1.92	0.72	-2.65**	-2.14	0.18	-12.14**
N/n	-0.08	0.23	-0.32			
d.f.		152			153	
R^2		0.75			0.75	
F stat.		156.04**			235.38**	

* $p < 0.05$
** $p < 0.01$

Table 3.12: Summary of regression models results

Model	Functional Form	
U-I-1	$t'(N)$	$= -3.42 + 5.71N$
U-1†	$t'(N, n)$	$= -1.89 + 0.48N + 1.73n + 4.63N \cdot n$
U-2†	$t'(N, n)$	$= \exp(1.47 + 0.23N + 0.59n - 0.01N \cdot n)$
U-3	$t'(N, n)$	$= \exp(3.54 + 0.24N - 1.64/n + 0.01N/n)$
U-4	$t'(N, n)$	$= \exp(3.52 + 0.25N - 1.61/n)$
C-I-1	$t'(N)$	$= \exp(0.58 + 1.03N)$
C-P-1	$t'(N)$	$= \exp(1.43 + 1.08N)$
C-1	$t'(N, n)$	$= \exp(-0.77 + 1.18N + 1.29n - 0.13N \cdot n)$
C-2	$t'(N, n)$	$= \exp(-0.21 + 1.00N + 0.89n)$
C-3	$t'(N, n)$	$= \exp(2.48 + 1.11N - 1.92/n - 0.08N/n)$
C-4	$t'(N, n)$	$= \exp(2.67 + 1.05N - 2.14/n)$

† Non-normal distribution of model residuals

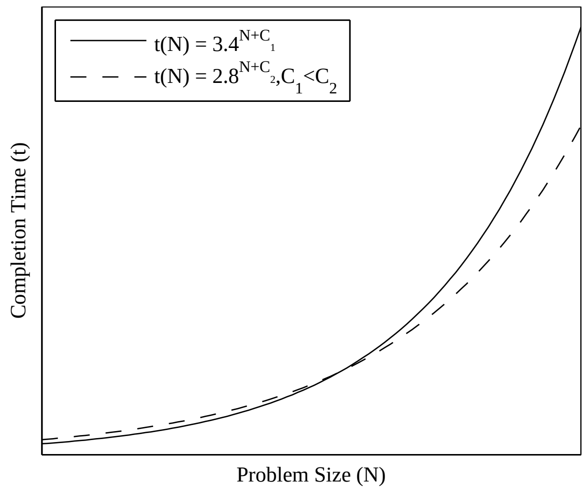

Figure 3-19: Comparison of coupled individual task model scaling factors: 3.4 from Hirschi and Frey (2002) and 2.8 in this study. Notional constants $C_1 < C_2$ illustrate potential differences in short-duration tasks.

3.5.1 Validation

Both the uncoupled and coupled individual task data fit the functional forms of models hypothesized by Hirschi and Frey (2002) in Models U-I-1 and C-I-1, however there are potentially differences in coefficient values. The linear factor for uncoupled tasks was not previously published, though the difference in time normalization would otherwise invalidate any comparison. The geometric factor for coupled tasks was previously reported as 3.4, but found to be $e^{1.03} = 2.80$ in this study. As a multiplicative factor, this should be robust to normalization procedures, but even a 95% confidence interval of [2.47, 3.20] excludes the previous value. This suggests two possibilities as illustrated in Fig. 3-19:

1. the longest-duration tasks take a comparatively shorter time to complete, and/or
2. the shortest-duration tasks take a comparatively longer time to complete.

Given the changes to the user interface to eliminate a "Refresh Plot" button-click to update values, it is likely the longest-duration tasks are comparatively easier in this study. Learning effects resulting from the randomized task order restriction previously discussed may also contribute to lower durations for the most complex tasks. It is also possible, but not well-supported from observation, that having multiple designers in the same environment may lengthen the time to solve shorter-duration tasks through distraction. The difference in coefficient value suggests it may be more difficult than expected to isolate a natural scaling law for solving design problems from the interface and context used in experimental studies and the previous results are also likely a function of the selected tool and user interface.

3.5.2 Uncoupled Tasks

Model U-1 hypothesizes the time to complete uncoupled tasks varies linearly with the problem size N and team size n. A regression model find significant a positive interaction effect between problem size N and team size n, however the results should be approached cautiously as the model residuals do not follow a normal distribution. Combining results from Models U-1 and U-I-1, an additional uncoupled variable requires about 5 seconds of task duration. Model U-2 log-transforms task duration to hypothesize a geometric scaling with problem size N and team size n, but again results in non-normal residuals. At a minimum, Models U-1 and U-2 suggest team size increases task time which is supported by observation of "one factor at a time" methods to find solutions in a sequential process with communication overhead.

Models U-3 and U-4 substitute team size n for inverse team size $1/n$ to produce results with normal residuals and significant contributions for problem size N and inverse team size $1/n$ on (log) task duration. The problem size multiplicative scaling factor in Model U-4 is $e^{0.25} = 1.28$, i.e. each additional variable increases the completion time by 28%. As the team size grows, the task time changes by a multiplier

$$\frac{t'(N, n+1)}{t'(N, n)} = \exp\left(\frac{1.61}{n} - \frac{1.61}{n+1}\right) \tag{3.28}$$

which decreases with increasing team size. For example, from $n = 1$ to $n = 2$ is an increase of 124% and from $n = 2$ to $n = 3$ is an increase of 31%. The lack of significant interaction term in Model U-2 suggests independence in contributions from technical complexity (changes in N) and social complexity (changes in n).

Results of Models U-3 and U-4 are limited. In particular, their form conflicts with Model U-I-1 for individual tasks. One would not expect task time to increase geometrically in uncoupled tasks, especially if considering only a single designer. Over the range of tasks considered, however, the problem size growth factor is mostly linear for small values of N as shown in the Taylor series expansion of Model U-4 at $N = 2$ for $n = 1$:

$$t'(N, n = 1) = e^{1.91+0.25N} \approx 11.13 + 2.78(N-2) + 0.34(N-2)^2 + 0.03(N-2)^3 + \mathcal{O}(N^4) \tag{3.29}$$

It is possible with additional data points the linear form of Model U-1 may be a better fit. In reality, there are likely two components to uncoupled tasks. The first component corresponds to the operational steps of finding a solution and is likely linear in the problem size as found in the individual task analysis. The second component corresponds to communication overhead in team tasks and is likely super-linear in the problem size, possibly geometric as found in the team task analysis. More data is required to evaluate this hypothesis in detail to incorporate a wider range of problem and team sizes.

3.5.3 Coupled Tasks

Model C-P-1 applies the form of Model C-I-1 to a coupled pair tasks. It produces a similar multiplicative scaling factor $e^{1.08} = 2.94$ (compared to $e^{1.03} = 2.80$ for C-I-1). The models differ substantially in the constant term which is larger for pair tasks, indicating a communication "overhead" factor of $e^{1.43} = 4.2$ compared to $e^{0.58} = 1.8$ for individual tasks.

Models C-1 and C-2 extend the model to hypothesize the time to complete coupled tasks varies geometrically with the problem size N and team size n. They find positive effects for problem size N and team size n and a small negative interaction from $N \cdot n$ removed in Model C-2. The problem size multiplicative scaling factor in Model C-2, $e^{1.00} = 2.72$, is similar to that found for individual and pair tasks. The team size multiplicative scaling factor in Model C-2 is of similar magnitude,

$e^{0.89} = 2.44$. The predicted form for individual and pair tasks under Model C-2

$$t'(N, n = 1) = \exp(0.68 + 1.00N) \tag{3.30}$$

$$t'(N, n = 2) = \exp(1.57 + 1.00N) \tag{3.31}$$

are similar to the results from Models C-I-1 and C-P-1.

Models C-3 and C-4 apply the uncoupled task model form to hypothesize the time to complete coupled tasks varies geometrically with the problem size N and inverse team size $1/n$. They find a positive effect for problem size N, a negative effect for inverse team size $1/n$ and a small negative interaction from N/n removed in Model C-4. The problem size multiplicative scaling factor in Model C-4, $e^{1.05} = 2.86$, is similar to that in Model C-2 and much larger than that in Model U-3 ($e^{0.25} = 1.28$). The team size multiplicative scaling factor in Model C-4,

$$\frac{t'(N, n+1)}{t'(N, n)} = \exp\left(\frac{2.14}{n} - \frac{2.14}{n+1}\right), \tag{3.32}$$

is also larger than that in Model U-3. For example, from $n = 1$ to $n = 2$ is in increase of 192% and from $n = 2$ to $n = 3$ is an increase of 43% (compared to 125% and 31% respectively).

While there are significant theoretical differences between Models C-2 and C-4, they cannot be distinguished in this study due to limited data points. Future work should study a wider range of design tasks involving larger problems and team sizes to adequately evaluate the model. In particular, it is a practical concern whether time scales geometrically with the team size n or with the inverse team size $1/n$. The former increases without bound while the latter converges to an upper bound.

3.5.4 Implications for Collaborative Design

The effects of problem size and team size can be isolated as independent factors for both types of tasks due to the lack of significant interaction terms in regression model results. Figure 3-20 shows the effect of problem size on task duration normalized to $N = 2$. One would expect real-world design tasks to lie between the upper- and lower-bounds of fully-coupled and fully-uncoupled problems studied in this experiment. Efforts to reduce technical complexity may also mitigate the geometric contribution of problem size similar to the observed difference in scaling factor compared to Hirschi and Frey (2002) possibly attributed to changes in user interface. Extensions may investigate centralized control over coupled problems using numerical optimization methods which could potentially reduce the time to solve design tasks to a polynomial scale.

Figure 3-21 shows the effect of team size on task duration normalized to $n = 1$. Team size factors for uncoupled and coupled tasks are similar under the barriers to collaboration included in this experiment. The results show greater than $2\times$ completion times for pairs as compared to individuals. An ideal collaborative process would reduce this factor to maintain or improve upon the efficiency of an individual designer. For example, if parallel work flows were enabled, the time required for teams may even be lower than that of an individual. Future efforts to develop collaborative processes may evaluate effectiveness by measuring changes to this factor.

While interaction terms between problem and team size were not significant in regression models of completion time for both uncoupled and coupled task, the two isolated factors produce combined effects on *design cost* which is a function of task duration and team size. Figure 3-22 illustrates

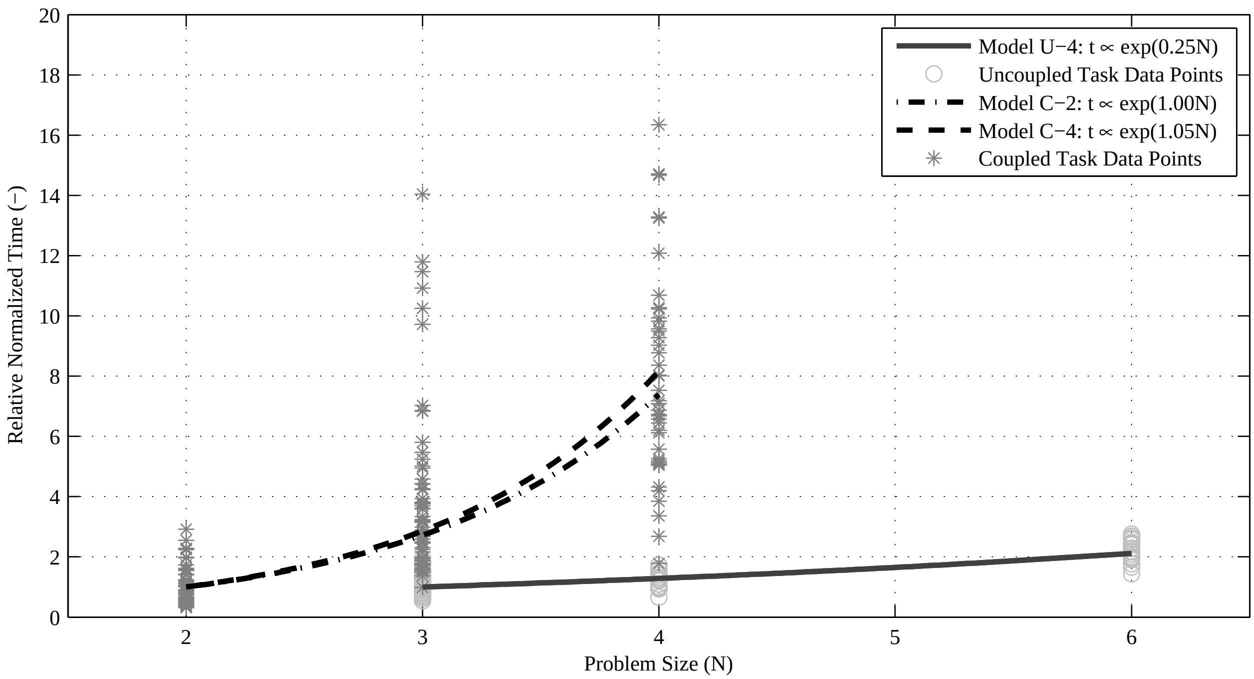

Figure 3-20: Effect of problem size on normalized task duration relative to $N = 2$ (for coupled) or $N = 3$ (for uncoupled) task.

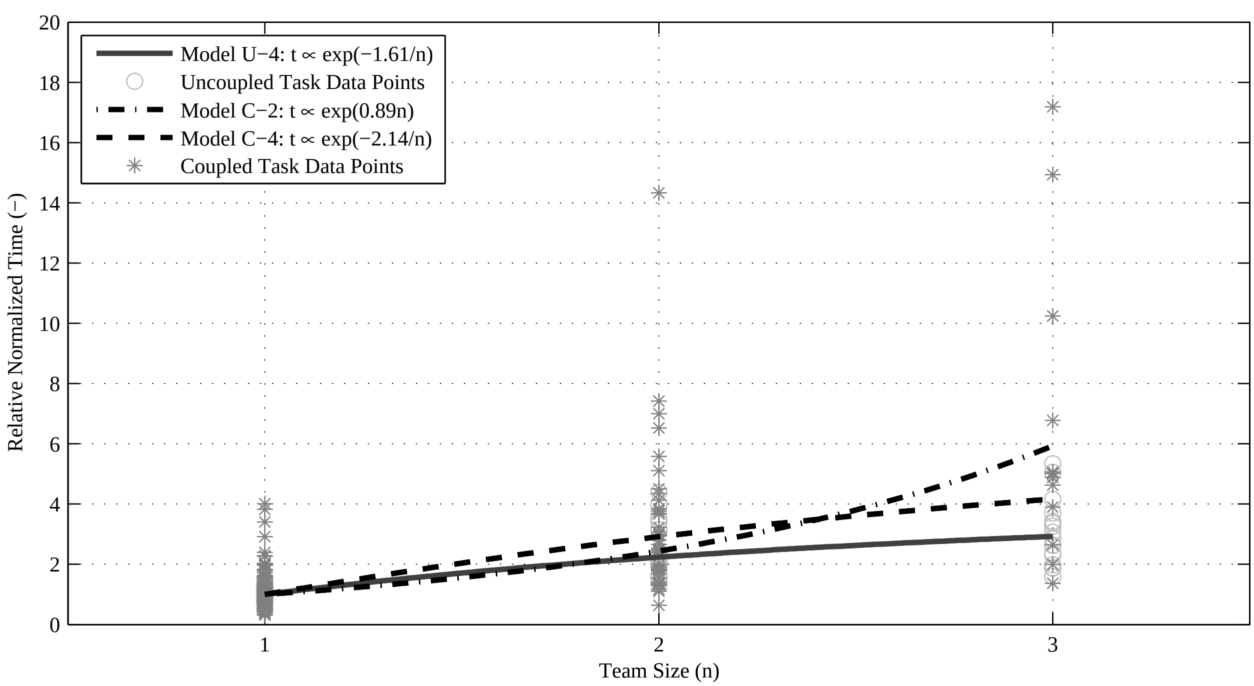

Figure 3-21: Effect of team size on normalized task duration relative to $n = 1$ (individual) task.

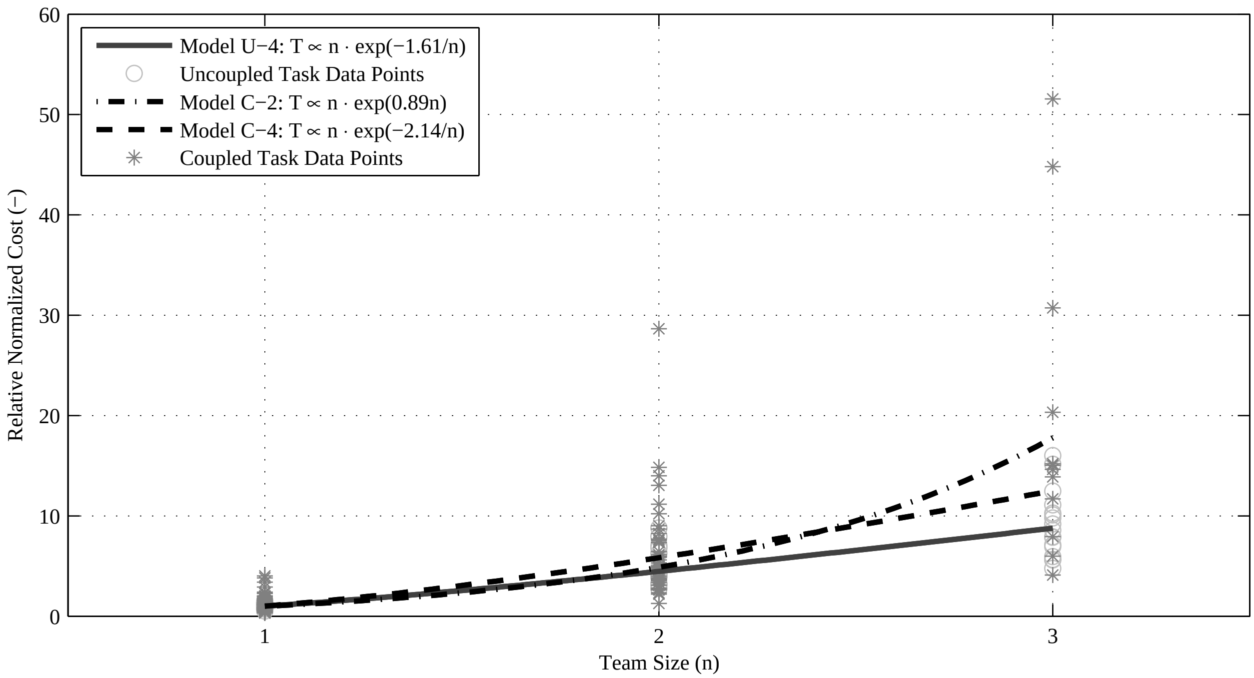

Figure 3-22: Effect of team size on normalized task cost (person-hours) relative to $n = 1$ (individual) task.

the effect of team size on design cost $T'(n, N) = n \cdot t'(n, N)$ for uncoupled and coupled tasks. Whereas Model C-4 predicts pairs take 3× the time as individuals and triads 4×, the total cost is closer to 6× for pairs and 12× for triads. These large factors may help explain cost and schedule overruns commonly seen in complex technically-coupled projects where both the technical and social difficulty may be underestimated.

3.5.5 Future Work

There are several potential extensions of this study left for future work. First, one may improve the external validity of the experimental results with revised design tasks. While the surrogate design task in this study used a linear system of equation, any general system model could be substituted. Non-linear, context-rich, larger, or partially-coupled system models would help improve the generalizability of results to real-world design tasks. In particular, De Jong functions (De Jong 1975) used in optimization evaluation may maintain a context-free task while introducing non-linear models. More complex system models will warrant an objective function to avoid a potentially difficult-to-find single zero-error solution. This will require additional practical considerations such as implementing a time limit and/or local compared to global objectives. In addition to a revised design task, a future study may also benefit from revised designer teams to include larger, experienced, or more familiar designers. In solving partially-coupled system models, the internal team structure becomes important to parallelize work. Also, there are some interesting research opportunities supported by the distributed software tool to evaluate virtual teams compared to in-person teams.

Next, there may be opportunities to combine the uncoupled and coupled task evaluation under a common framework using a complexity metric. While treated as two separate classes of problems in this study, there are likely common underlying features which may be quantified using technical and social complexity metrics rather than the problem size N and team size n variables used in this analysis. One candidate is the comprehensive metric for structural complexity including graph energy as a measure of entropy (Sinha and de Weck 2012). Its complexity metric C takes the form

$$C = \sum_i \alpha_i + \left(\sum_i \sum_j \beta_{ij} A_{ij} \right) \gamma E(A) \tag{3.33}$$

where α_i, β_{ij}, and γ are weighting factors for the component, interactive, and architectural components of structural complexity, A is the adjacency matrix (i.e. binary design structure matrix) of interactions and $E(A)$ is the graph energy of the adjacency matrix defined by $E(A) = \sum_i \sigma_i$ where σ_i is the singular value decomposition of A. Such a metric may be applied to both technical interactions (A based on the M matrix) and social interactions (A based on the D matrix) provided appropriate weighting factors.

Finally, a promising area of future work would study the effect of collaborative methods as compared to the baseline case in this study having significant barriers to collaboration. For example, sharing output values on a common display would allow uncoupled tasks to be completed in parallel. Other numerical outputs may help designers quantitatively evaluate the effect of input changes in complex design problems. For example, displaying the quantitative error between output and target values, normalized to a reasonable scale, could support collaborative decision-making by building a common mental model of "goodness." This aligns with the "discourse group preference" phase of the engineering collaboration via negotiation (ECN) model hypothesized by Lu et al. (2007), which in the present study is limited to qualitative judgments.

References for Chapter 3

Ameri, F., J. D. Summers, G. M. Mocko, and M. Porter (2008). "Engineering Design Complexity: An Investigation of Methods and Measures". In: *Research in Engineering Design* 19.2-3, pp. 161–179. DOI: 10.1007/s00163-008-0053-2.

Arena, M. V., O. Younossi, K. Brancato, I. Blockstein, and C. A. Grammich (2008). *Why Has the Cost of Fixed-Wing Aircraft Risen?* Tech. rep. Santa Monica, California: The RAND Corporation.

Argote, L., C. A. Insko, N. Yovetich, and A. A. Romero (1995). "Group Learning Curves: The Effects of Turnover and Task Complexity on Group Performance". In: *Journal of Applied Social Psychology* 25.6, pp. 512–529. DOI: 10.1111/j.1559-1816.1995.tb01765.x.

Arias, E., H. Eden, G. Fischer, A. Gorman, and E. Scharff (2000). "Transcending the Individual Human Mind – Creating Shared Understanding through Collaborative Design". In: *ACM Transactions on Computer-Human Interaction* 7.1, pp. 84–113. DOI: 10.1145/344949.345015.

Bashir, H. A. and V. Thomson (Sept. 1999). "Estimating Design Complexity". In: *Journal of Engineering Design* 10.3, pp. 247–257. DOI: 10.1080/095448299261317.

Braha, D. and O. Maimon (July 1998). "The Measurement of a Design Structural and Functional Complexity". In: *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans* 28.4, pp. 527–535. DOI: 10.1109/3468.686715.

Card, D. and W. Agresti (June 1988). “Measuring Software Design Complexity”. In: *Journal of Systems and Software* 8.3, pp. 185–197. DOI: 10.1016/0164-1212(88)90021-0.

Cohen, S. G. and D. E. Bailey (1997). “What Makes Teams Work: Group Effectiveness Research from the Shop Floor to the Executive Suite”. In: *Journal of Management* 23.3, pp. 239–290.

De Jong, K. (1975). “An analysis of the behaviour of a class of genetic adaptive systems”. PhD thesis. Ann Arbor, Michicagn: University of Michigan.

Dirks, K. T. (1999). “The Effects of Interpersonal Trust on Work Group Performance”. In: *Journal of Applied Psychology* 84.3, pp. 445–455. DOI: 10.1037/0021-9010.84.3.445.

El-Haik, B. and K. Yang (July 1999). “The Components of Complexity in Engineering Design”. In: *IIE Transactions* 31.10, pp. 925–934. DOI: 10.1080/07408179908969893.

Eppinger, S. D., D. E. Whitney, R. P. Smith, and D. A. Gebala (1991). “Organizing the tasks in complex design projects”. In: *Computer-Aided Cooperative Product Development*, pp. 229–252.

Fenton, N. E. and M. Neil (July 1999). “Software metrics: successes, failures and new directions”. In: *Journal of Systems and Software* 47.2-3, pp. 149–157. DOI: 10.1016/S0164-1212(99)00035-7.

Flager, F., D. J. Gerver, and B. Kallman (2014). “Measuring the impact of scale and coupling on solution quality for building design problems”. In: *Design Studies* 35.2, pp. 180–199. DOI: 10.1016/j.destud.2013.11.001.

Goodman, P. S. and D. P. Leyden (1991). “Familiarity and group productivity”. In: *Journal of Applied Psychology* 76.4, pp. 578–586. DOI: 10.1037/0021-9010.76.4.578.

Hirschi, N. W. and D. D. Frey (2002). “Cognition and complexity: An experiment on the effect of coupling in parameter design”. In: *Research in Engineering Design* 13, pp. 123–131. DOI: 10.1007/s00163-002-0011-3.

Ingham, A. G., G. Levinger, J. Graves, and V. Peckham (1974). “The Ringelmann effect: Studies of group size and group performance”. In: *Journal of Experimental Social Psychology* 10.4, pp. 371–384. DOI: 10.1016/0022-1031(74)90033-X.

Jehn, K. A. and P. Pradhan (1997). “Interpersonal relationships and task performance: An examination of mediation processes in friendship and acquaintance groups”. In: *Journal of Personality and Social Psychology* 72.4, pp. 775–790. DOI: 10.1037/0022-3514.72.4.775.

Kerr, N. L. and R. S. Tindale (2004). “Group Performance and Decision Making”. In: *Annual Review of Psychology* 55, pp. 623–655. DOI: 10.1146/annurev.psych.55.090902.142009.

Kravitz, D. A. and B. Martin (1986). “Ringelmann Rediscovered: The Original Article”. In: *Journal of Personality and Social Psychology* 50.5, pp. 936–941. DOI: 10.1037/0022-3514.50.5.936.

Lu, S. C.-Y., W. Elmaraghy, G. Schuh, and R. Wilhelm (2007). “A Scientific Foundation of Collaborative Engineering”. In: *CIRP Annals – Manufacturing Technology* 56.2, pp. 605–634. DOI: 10.1016/j.cirp.2007.10.010.

MacCormack, A., J. Rusnak, and C. Baldwin (2011). *Exploring the Duality between Product and Organizational Architectures: A Test of the “Mirroring” Hypothesis*. White Paper 08-039. Harvard Business School.

Maier, M. W. (1998). “Architecting Principles for Systems-of-Systems”. In: *Systems Engineering* 1.4, pp. 267–284. DOI: 10.1002/(SICI)1520-6858(1998)1:4$<$267::AID-SYS3$>$3.0.CO;2-D.

McCabe, T. J. and C. W. Butler (Dec. 1989). “Design Complexity Measurement and Testing”. In: *Communications of the ACM* 32.12, pp. 1415–1425. DOI: 10.1145/76380.76382.

Miller, G. A. (1956). “The magic number seven, plus or minus two: some limits on our capacity for processing information”. In: *Psychology Review* 63, pp. 81–97.

Mullen, B. and C. Copper (1994). “The Relation Between Group Cohesiveness and Performance: An Integration”. In: *Psychological Bulletin* 115.2, pp. 210–227. DOI: 10.1037/0033-2909.115.2.210.

National Research Council (2011). *Assessment of Impediments to Interagency Collaboration on Space and Earth Science Missions*. Washington, DC: The National Academies Press.

Sheard, S. A. and A. Mostashari (2010). “A Complexity Typology for Systems Engineering”. In: *8th Conference on System Engineering Research*. Hoboken, New Jersey.

Sinha, K. (2014). “Structural Complexity and its Implications for Design of Cyber-Physical Systems”. PhD thesis. Cambridge, Massachusetts: Massachusetts Institute of Technology.

Sinha, K. and O. L. de Weck (2012). “Structural Complexity Metric for Engineered Complex Systems and its Application”. In: *14th International Design Structure Matrix (DSM) Conference*. 101. Kyoto, Japan.

Suh, N. P. (1999). “A Theory of Complexity, Periodicity and the Design Axioms”. In: *Research in Engineering Design* 11.2, pp. 116–131. DOI: 10.1007/PL00003883.

Weingart, L. R. (1992). “Impact of Group Goals, Task Component Complexity, Effort, and Planning on Group Performance”. In: *Journal of Applied Psychology* 77.5, pp. 682–693. DOI: 10.1037/0021-9010.77.5.682.

Chapter 4

The Infrastructure System-of-systems (ISoS) Modeling Framework

> "Here [the development of general knowledge—or theory—about how cities work] the problems are far more difficult that in particularistic problem-solving, but here also the ultimate payoff is much greater. Only the development of a body of dependable empirical theory about urban processes and structures can clear the way for a massive assault on the many vexing problems which can, without that theory, only be approached tentatively and in piecemeal fashion. Simulation is an analytic technique well adapted to the process of theory-building: stating, testing, revising, retesting. To fail to develop the technique to its fullest potential is socially as well as intellectually irresponsible."
>
> Edward Berger, Harvey Boulay, and Betty Zisk in "Simulation and the City: A Critical Overview" (1970)

This chapter takes the first step towards designing an interoperable simulation game for infrastructure systems design by defining the infrastructure system-of-systems (ISoS) modeling framework. The ISoS framework is designed to reduce the cost of developing decentralized simulation models by identifying a common generic platform on which to base component models. It builds on existing literature in infrastructure modeling and simulation described in Section 4.1 to balance the detailed network flow models used in operational optimization with aggregated system dynamics models used for high-level systems analysis.

The ISoS framework includes two major components suitable for strategic infrastructure planning. The spatial-structural template defined in Section 4.2 specifies the simulation state, i.e. the information required to describe infrastructure at a snapshot in time. It disaggregates infrastructure systems to individual elements using graph-theoretic concepts for spatial resolution. The temporal-behavioral template defined in Section 4.3 specifies the allowable state changes for time dynamics. Element behaviors are functionally classified and resource flows are aggregated at locations to enable strategic analysis without exhaustive modeling.

The applicability and generality of the ISoS framework is demonstrated with four descriptive case studies in Section 4.4. Each case study outlines a potential model instantiation to address infrastructure planning in diverse domains. The first two cases study space systems: a retrospective view of International Space Station assembly and resupply and a prospective view of future human exploration on Mars. The other two cases study terrestrial infrastructure: an introspective view of resource management in Burkina Faso and a prospective view of national infrastructure planning in Saudi Arabia.

4.1 Modeling Infrastructure Systems

While many single-sector models of infrastructure have been developed, the study and analysis of infrastructure with an emphasis on interdependencies is relatively new. As discussed by Pederson et al. (2006), events such as the Oklahoma City bombing and the Information Warfare Report by the Defense Science Board (1996) first brought attention to the increasing reliance on information and computer systems for managing and controlling infrastructure systems in the mid-1990s.

The President's Commission on Critical Infrastructure Protection (PCCIP) released in 1997 specifically called for attention to infrastructure interdependencies (PCCIP 1997). Later, executive orders established the National Infrastructure Advisory Council (NIAC), Department of Homeland Security (DHS), and National Infrastructure Simulation and Analysis Center (NISAC) as a partnership between Sandia National Laboratories and Los Alamos National Laboratory to advance modeling and simulation techniques as applied to infrastructure systems. Widespread infrastructure failures following the September 11, 2001 terrorist attack on the World Trade Center further solidified the role of interdependencies in infrastructure analysis and planning and expanded the initial scope of the DHS (O'Rourke 2007).

A wide range of integrated modeling approaches emerged over the past decade as a result of these national initiatives. While most applications emphasize operational resilience and security, some strategic-level concepts of evolution, design, and planning are also considered. The overview provided here identifies three categories of such approaches: conceptual models, aggregated system models, and detailed network models. The following sections describe each approach from examples in literature and highlight concepts applicable to strategic infrastructure systems design.

4.1.1 Conceptual Models

Conceptual models help create a common understanding of factors relating to infrastructure systems. While not executable or computable as a simulation, they provide guidance through established constructs. Although not by name, Rinaldi et al. (2001) introduces concepts of static and dynamic dependency networks illustrated in Figure 4-1. A static network connects infrastructure components with arcs representing one or more types of interdependencies experienced over some time horizon. For example, a water component may require electricity for pumping from an energy component which, in turn, may require water for cooling. The dynamic or time-expanded network illustrates the effect of time on dependencies. Following the same example, if the water component increases flow to meet additional demands, the first-order effect requires more electricity. A second-order effect may further increase water demand for cooling. Such higher-order effects may quickly encompass many coupled components with complex feedback relationships. In general, any cycles in the static network (e.g. water to energy to water) produce a sequence of nth order effects in the dynamic network.

Rinaldi et al. (2001) further classifies infrastructure interdependencies as physical (resource flows), cyber (information flows), geographic (spatially proximate), and logical (other, such as effects of control). In the previous example in Figure 4-1 there is a physical interdependency between the water and energy components. From this perspective, interdependencies are static attributes of pairs and sets of elements. While physical and geographic interdependencies can

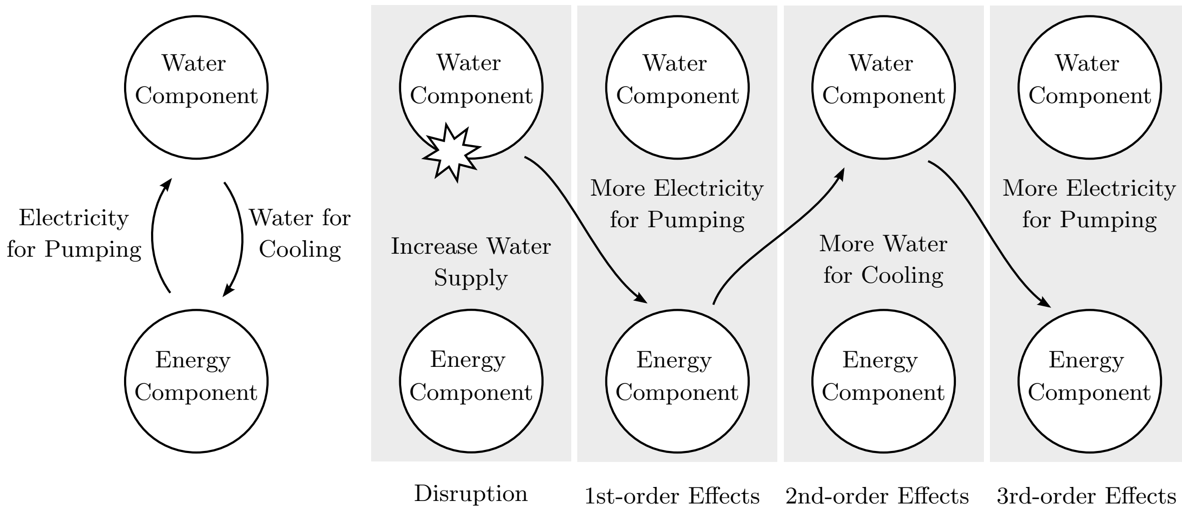

Figure 4-1: A static network (left) illustrates dependencies between infrastructure components over some time horizon. A dynamic network (right) illustrates time-varying dependencies including nth order effects.

Table 4.1: System function classified by process and operand

	Organisms	Matter	Energy	Information	Currency
Transform	F_O	F_M	F_E	F_I	F_C
Transport	P_O	P_M	P_E	P_I	P_C
Store	S_O	S_M	S_E	S_I	S_C
Exchange	X_O	X_M	X_E	X_I	X_C
Control	C_O	C_M	C_E	C_I	C_C

be linked to elements' structural connectivity, cyber and logical interdependencies are based on dynamic behaviors not represented in a spatial form.

As another perspective on interdependencies, de Weck et al. (2012, pp. 38–43) describe engineering system behaviors as a functional classification consisting of a 5×5 matrix of operands and operations shown in Table 4.1. This classification extends a 3×3 matrix from van Wyk (1988) to include exchanging and control operations and finance and organism operands representing the broader scope of engineering systems. Systems are classified by one or more functions—for example, a power plant is F_M, F_E as it transforms matter to create energy. This classification agrees with that of Rinaldi et al. in the existence of physical (matter, energy) and cyber (information) resource flows and some logical effect of control, however here these functions are presented as individual element's behaviors without distinct interdependencies.

4.1.2 Aggregated System Models

Aggregated system models represent large-scale infrastructure with simplified representations as a top-down approach. They are often used to guide high-level decisions and policy because of their level of abstraction and relatively few input parameters. System models are unified in a

Table 4.2: Infrastructures and assets represented in the CIP/DSS

Critical Infrastructure	Critical Infrastructure (cont.)	Key Asset
Agriculture & Food	Information and Telecommunications	National Monuments and Icons
Water	Energy	Nuclear Power Plants
Public Health	Transportation	Dams
Emergency Services	Banking and Finance	Government Facilities
Government	Chemical Industry & Hazardous Materials	Commercial Key Assets
Defense Industrial Base	Postal and Shipping	

single system-of-systems model with shared state variables to capture system couplings. Two examples of this approach use the system dynamics formalism with stock and flow variables to model infrastructure systems (Bush et al. 2005; Min et al. 2007). System dynamics models are represented by a system of coupled equations for rapid simulation of behaviors usable within an optimization routine.

Bush et al. (2005) discuss the Critical Infrastructure Protection Decision Support System (CIP/DSS) as a joint project of Argonne National Laboratory, Los Alamos National Laboratory, and Sandia National Laboratories. The model incorporates 17 critical infrastructures and key asset categories in Table 4.2 to provide insights for decision-making at the metropolitan level. Models are implemented in Vensim, a system dynamics software tool, and integrated using a custom model "linker" to assemble the unified system-of-systems model from single-sector models. The system-of-systems model has 4482 variables, with each sector contributing between 10 (agriculture) and 802 (energy) variables.

Min et al. (2007) discuss the application of system dynamics to model national-level critical infrastructure interdependencies. They use the ICAM Definition for Functional Modeling (IDEFØ) technique to "help define data requirements and describe the exchange of information between the individual simulation models." Nonlinear optimization methods select control parameters to maximize total economic revenue in the case of disruptions. The model includes energy production, transmission, and distribution in regional electric power, natural gas, and petroleum infrastructure sectors in 10 interconnection regions in the US. The economic sector includes regional residential, commercial, and industrial consumers. The model has over 5000 variables and parameters and application to broader system-of-systems is left for future work.

There are two key limitations for aggregated system models. First, their structure defines a static unit of analysis, implying homogeneity of components within its boundary. For example, capacity expansion in a sector may assume simple scaling of performance from existing infrastructure rather than discrete capital investments in potentially different technologies. Furthermore, spatial decomposition—such as the ten interconnection regions in Min et al. (2007) or six regional natural gas stocks in Ellison (2006)—is fixed and cannot be easily modified or extended. The second limitation corresponds to the aggregation of system-level behaviors which cannot be disaggregated to element-level decisions. In other words, aggregated models are more oriented towards systems analysis rather than the synthesis approach of systems design. As a result of both limitations, an aggregated system model may address high-level questions for known systems, but cannot easily evaluate individual element decisions in the context of the larger system.

4.1.3 Detailed Network Models

Detailed network models represent individual infrastructure elements from a bottom-up approach. They are often used to analyze short-term behaviors resulting from disturbances. To increase modeling detail, this class of models use network-based or graph-theoretic representations of infrastructure elements. In most applications, nodes are resource production or consumption elements and edges are infrastructure elements enabling resource flow between nodes. Network models are particularly useful forms for linear programming (Dantzig 1963) and for algorithms to solve the shortest path problem (Dijkstra 1959), maximum flow/minimum cost problem (Bellman 1958; Ford 1956; Edwards and Karp 1972), and multi-commodity flow problem (Hu 1963).

In the critical infrastructure modeling system (CIMS) framework, Dudenhoeffer et al. (2006) define a node as "an entity that acts as a source, produces, consumes, or transforms a resource..." and an edge as "a physical or virtual entity that acts as a conduit for flow for a physical quantity, information, or influence ... [representing] a direct level of dependence." Physical, informational, geo-spatial, policy/procedural, and societal dependencies each have a combined structure-behavior representation as an edge. Infrastructure system sectors are each modeled as a network layer with cross-system interdependencies as inter-layer edges. Existing applications of CIMS are primarily visual presentation of interdependencies in "what-if" analyses. Future applications seek to optimize asset priorities for protection or restoration.

Zhang et al. (2005) describe a generalized transportation network (GTN) as a multi-layered infrastructure network (MIN) consisting of auto, urban freight, and data subnetworks with interdependencies related to limited transport capacity and the option to telecommute. Flow dynamics are driven by a game-theoretic approach based on a Cournot-Nash equilibrium operationalized by agent-based simulation methods. The approach is extended by Zhang and Peeta (2011) to consider transportation, telecommunication, energy, and power systems. Behaviors are determined through a computable general equilibrium (CGE) problem solved with nonlinear programming methods and operationalized by agent-based simulation methods.

There are two key limitations for detailed network models. First, their structure defines infrastructure elements as the unit of analysis, requiring large amounts of data to build a model. For example, if two infrastructure elements are distinct objects but have interdependent input/output resources, an edge must be defined to facilitate the resource transaction. Second, as network models are generally dedicated to operational analysis, there is little consideration for dynamic topology corresponding to network evolution. Infrastructure networks are typically static, with variable node and edge parameters to represent operational details.

4.1.4 ISoS Modeling Framework Objectives

As discussed in the above sections, existing modeling approaches pose several limitations to strategic infrastructure systems design. On one hand, aggregated system models do not consider the spatial structure of constituent infrastructure elements and cannot disaggregate system decisions to the element level for synthesis. On the other hand, detailed network models exhaustively define element-level structure and behavior requiring significant information and often assume a static network topology.

In addition to these limitations, existing modeling methods do not address model interoperability. Aggregated models rely on shared state variables to represent coupled systems, requiring all system models to be integrated in a centralized modeling environment. Similarly, network models compose constituent systems with inter-system edges. Complete integration of a system-of-systems model may not be possible under some circumstances where information-sharing barriers or differences in model implementation prevent distribution of models between organizations with authority over various infrastructure systems.

There is some discussion of interoperability in related literature. Pederson et al. (2006) identifies the High Level Architecture (HLA, IEEE 2010) and Distributed Interactive Simulation (DIS, IEEE 1998) frameworks as options for distributed simulation but recognized that "no standards exists [sic] that directly address infrastructure and specifically cross sector modeling." Tolone et al. (2008) presents a simpler service-oriented architecture specific to infrastructure system-of-systems analysis. Both cases, however, focus on the method rather than the implications of interoperability within the modeling framework.

To address these limitations, the proposed infrastructure system-of-systems (ISoS) modeling framework seeks to meet four objectives.

1. Disaggregate infrastructure systems into elements to capture spatial structure.

2. Aggregate element behaviors at a level suitable for strategic analysis.

3. Dynamically change network topology to correspond to infrastructure evolution.

4. Enable interoperability between system models having decentralized authority.

These objectives are addressed in the following sections defining the ISoS modeling framework.

4.2 Spatial-Structural Template

The spatial-structural template defines the instantaneous state of an infrastructure system-of-systems model. It includes a context template applicable across models in a common domain and an instantiation template for the unique components in a particular model.

4.2.1 Context Template

The context template defines the allowable infrastructure locations and resource types which can be reused for all applications having equivalent spatial characteristics.

The context network is based on graph-theoretic concepts. Unlike other network-based frameworks, nodes do not correspond to infrastructure: they are defined as spatial units of aggregation where resources are freely transferable between co-located infrastructure. Usually nodes correspond to physical areas (zones, cities, regions, etc.) however virtual nodes can represent other concepts such as financial or information repositories.

Network locations are valid infrastructure positions at or between nodes. For a set of nodes $\mathbf{N}$, the set of allowable locations is a set of node pairs

$$\mathbf{L} = \{(n_o, n_d)_i\} : n_o, n_d \in \mathbf{N} \; \forall \; i \tag{4.1}$$

where l_i is a nodal location if $n_o = n_d$ and an edge location if $n_o \neq n_d$. Edge locations are directed such that n_o is the origin node and n_d is the destination node. Figure 4-2 illustrates a set of six elements at allowable locations on a context network of three nodes. Note $\mathbf{L}$ is not a complete graph

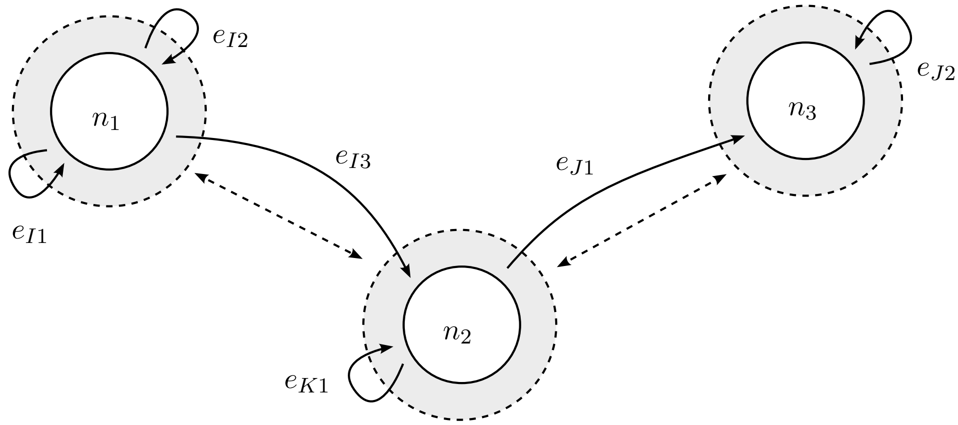

Figure 4-2: A notional spatial-structural network includes three nodes (n_1,n_2,n_3) and seven edges ($(n_1, n_1), (n_1, n_2), (n_2, n_1), (n_2, n_2), (n_2, n_3), (n_3, n_2), (n_3, n_3)$). The instantiation model includes six elements ($e_{I1}, e_{I2}, e_{I3}, e_{J1}, e_{J2}, e_{K1}$) in three systems ($I$, J, K). Elements with nodal locations (e.g. that of e_{I1}) form loops at one node while those with edge locations (e.g. that of e_{I3}) link two nodes.

as the edge locations between nodes n_1 and n_3 are not allowed, indicating constraints preventing infrastructure between these nodes.

The allowable resource types is a set

$$\mathbf{T} = \{\tau_i\} \; \forall \; i \tag{4.2}$$

where each resource type τ_i corresponds to operands of the 5×5 framework: mass-based (e.g. water), energy-based (e.g. electricity), information-based (e.g. bits), currency-based (e.g. US dollars), organism-based (e.g. people) or any other resource measured using a ratio scale with a non-arbitrary zero point. A set of resources r is realized as a set of pairs of a resource type and a real quantity as shown in Eq. 4.3.

$$r = \{(\tau, q)_i\} : \tau \in \mathbf{T}, q \in \mathbb{R} \; \forall \; i \tag{4.3}$$

Resources aggregated by resource type are fungible (i.e. representing commodities) with union and difference operators defined in Eq. 4.4–4.5. While negative quantities are allowed within the general framework to represent reverse flows, they may be disallowed for specific applications such as resource storage.

$$r_i \cup r_j = \{(\tau, q_i + q_j)\} \; \forall \; \tau : (\tau, q_i) \in r_i, (\tau, q_j) \in r_j \tag{4.4}$$

$$r_i - r_j = \{(\tau, q_i - q_j)\} \; \forall \; \tau : (\tau, q_i) \in r_i, (\tau, q_j) \in r_j \tag{4.5}$$

4.2.2 Instantiation Template

The instantiation template defines the infrastructure elements participating in a particular system-of-systems. The instantiated elements are a set

$$\mathbf{E} = \{e_{ij}\} \tag{4.6}$$

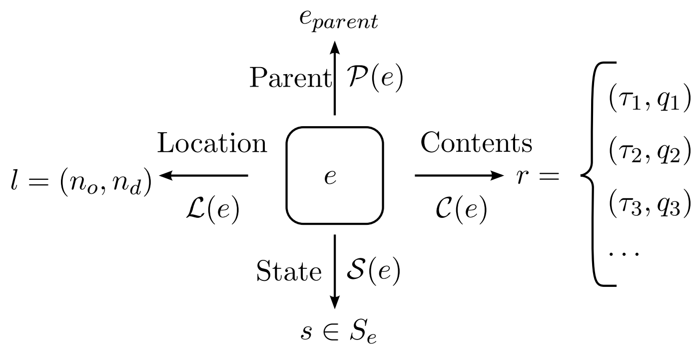

Figure 4-3: Infrastructure element state properties include resource contents ($\mathcal{C}$), location ($\mathcal{L}$), parent element ($\mathcal{P}$), and operational state ($\mathcal{S}$).

where each element e_{ij} is component j of system i. Elements are uniquely assigned to one system; however rather than sector-specific resource functionality, the system assignment designates management or control. Consider element e_{K1} is a combined-cycle power/desalination plant in Fig. 4-2. It may operate with both water and electricity resources, but is managed by one organization (system K) which may differ from other water (system J) or electricity (system I) infrastructure. Elements have four stateful properties illustrated in Figure 4-3: resource contents, location, parent element, and operational state.

Elements are the only containers of resources in the modeling framework. Resources contained within an element are identified by a contents function

$$\mathcal{C} = \mathcal{C}(e) : e \mapsto r \tag{4.7}$$

which maps an element $e \in \mathbf{E}$ to a set of resources r. Consider element e_{I1} is a water tank in Fig. 4-2 containing q units of water of resource type τ. Its contents are $\mathcal{C}(e_{I1}) = \{(\tau, q)\}$.

Elements exist at only one location at a time. An element's spatial position is identified by the location function

$$\mathcal{L} = \mathcal{L}(e) : e \mapsto l \in \mathbf{L} \tag{4.8}$$

which maps an element $e \in \mathbf{E}$ to an allowable location l. Elements at edge locations may be mobile elements in transit between nodes (e.g. trucks, ships) or fixed distribution elements transferring resources between nodes (e.g. pipelines). Shorthand notations $\mathcal{L}_o(e) = (n_o, n_o)$ and $\mathcal{L}_d(e) = (n_d, n_d)$ are used to identify the origin and destination locations for elements where $\mathcal{L}(e) = (n_o, n_d)$. For example, the location of element e_{I3} in Fig. 4-2 is $\mathcal{L}(e_{J1}) = (n_2, n_3)$ and $\mathcal{L}_o(e_{J1}) = (n_2, n_2)$ and $\mathcal{L}_d(e_{J1}) = (n_3, n_3)$.

In addition to spatial location, elements can also be arranged in a hierarchical structure of nested relationships. Nested structure is identified by the parent function

$$\mathcal{P} = \mathcal{P}(e) : e \mapsto e_{parent} \in \mathbf{E} \tag{4.9}$$

which maps an element $e \in \mathbf{E}$ to the element containing it. An element not nested inside another element is defined to be its own parent, i.e. $\mathcal{P}(e) = e$. The parent function may also be raised to multiple powers to map an element to its nth parent, e.g. a "grandparent" relationship is $\mathcal{P}^2(e) = \mathcal{P}(\mathcal{P}(e))$.

Finally, any other attributes necessary to describe an element's state are defined in the state function

$$\mathcal{S} = \mathcal{S}(e) : e \mapsto s \in S_e \tag{4.10}$$

which maps an element $e \in \mathbf{E}$ to a state s among its set of allowable states S_e. Each operational state is linked with a model to express one or more time-dependent behaviors described in the following section.

4.3 Temporal-Behavioral Template

The temporal-behavioral template defines state changes which take place during a simulation execution. Behaviors operate either on resources or elements. The formulations described here assume any required pre-conditions such as spatial compatibility or capacity constraints are satisfied. Any violations should generate errors in a model implementation.

4.3.1 Resource Behaviors

Resource behaviors are exhibited by elements to produce, move, or consume resources during a simulation. Figure 4-4 illustrates the storing, transporting, transforming, and exchanging behaviors acting on element properties and location resource flows. Storing and transforming behaviors modify flows at an element's location while transporting and exchanging modify flows between an element's origin and destination.

The resource storing behavior stores or retrieves resources from an element's internal contents. It is a function of an element $e \in \mathbf{E}$, resources to store r_{in} and resources to retrieve r_{out} with state changes specified in Eq. 4.11.[1]

$$\begin{aligned} \mathcal{R}_{store} &= \mathcal{R}_{store}(e, r_{in}, r_{out}) \\ &\mathcal{C}(e) \leftarrow (\mathcal{C}(e) \cup r_{in}) - r_{out} \\ &\mathcal{F}_{out}(\mathcal{L}(e)) \leftarrow \mathcal{F}_{out}(\mathcal{L}(e)) \cup r_{in} \\ &\mathcal{F}_{in}(\mathcal{L}(e)) \leftarrow \mathcal{F}_{in}(\mathcal{L}(e)) \cup r_{out} \end{aligned} \tag{4.11}$$

The resource transporting behavior moves resources between locations. It is a function of an element $e \in \mathbf{E}$, input resources r_{in}, and output resources r_{out} with state changes shown in Eq. 4.12.

$$\begin{aligned} \mathcal{R}_{transport} &= \mathcal{R}_{transport}(e, r_{in}, r_{out}) \\ &\mathcal{F}_{out}(\mathcal{L}_o(e)) \leftarrow \mathcal{F}_{out}(\mathcal{L}_o(e)) \cup r_{in} \\ &\mathcal{F}_{in}(\mathcal{L}_d(e)) \leftarrow \mathcal{F}_{in}(\mathcal{L}_d(e)) \cup r_{out} \end{aligned} \tag{4.12}$$

Input and output resources may be functionally defined. For example, a model of perfect transportation has $r_{in} = r_{out} = r$. Other models may assign a transportation efficiency $\eta \subseteq [0, 1]$ which,

[1]A word on notation: the expression $\mathcal{X} \leftarrow \mathcal{X}'$ used in this section indicates the value of a state expression $\mathcal{X}$ has been changed to $\mathcal{X}'$ to exhibit a behavior. Additionally, each behavior is indexed by the time t at which it occurs, although this has been omitted for brevity in this chapter.

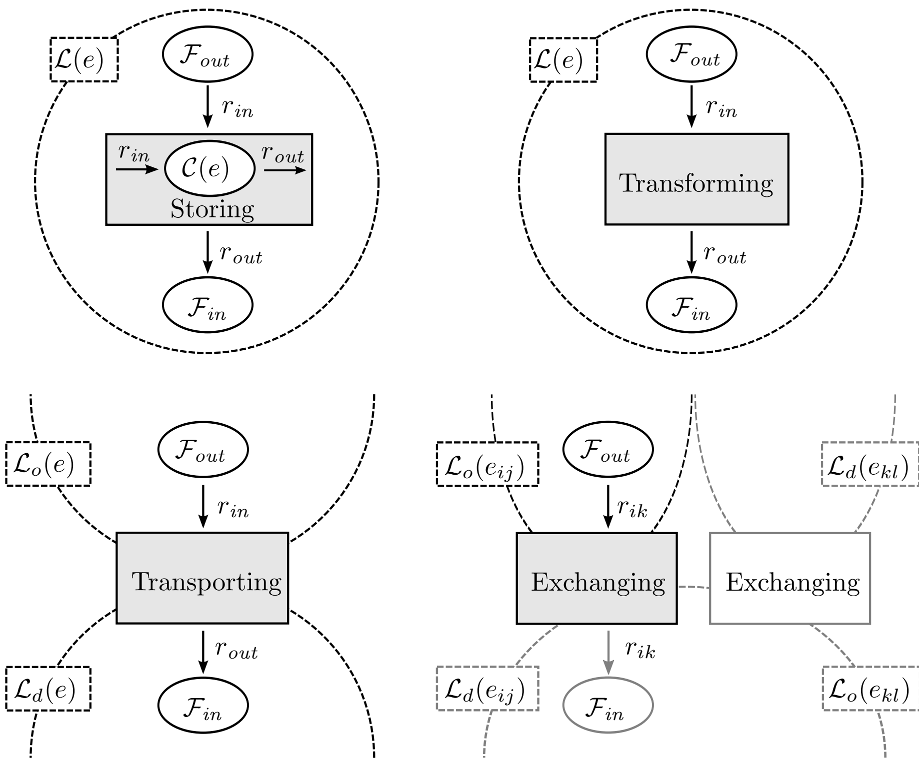

Figure 4-4: Resource storing, transporting, transforming, and exchanging behaviors modify resource flows by consuming out-flow resources ($\mathcal{F}_{out}$) and producing in-flow resources ($\mathcal{F}_{in}$). The total out-flow of resources from all behaviors must equal the in-flow, i.e. $\mathcal{F}_{out} = \mathcal{F}_{in}$, at each location during a time period.

for transport of a resource with a single type $r = \{(\tau_1, q_1)\}$, would be represented in Eq. 4.13–4.14.

$$r_{in}(r) = \{(\tau_1, q_1)\} \tag{4.13}$$

$$r_{out}(r) = \{(\tau_1, q_1 \cdot \eta)\} \tag{4.14}$$

Applications of transport efficiency η include modeling leakage in water pipelines and resistive losses in electrical power lines.

More detailed functions may also represent other resources required for transportation. Continuing the previous example, adding a unit transport demand $f_{2|1}$ of type τ_2 at the origin location is shown in Eq. 4.15–4.16.

$$r_{in}(r) = \{(\tau_1, q_1), (\tau_2, q_1 \cdot f_{2|1})\} \tag{4.15}$$

$$r_{out}(r) = \{(\tau_1, q_1 \cdot \eta), (\tau_2, 0)\} \tag{4.16}$$

Applications of transport demand factor $f_{i|j}$ include modeling pumping energy for fluid distribution or other variable operations expenses.

The resource transforming behavior changes input to output resource types. It is a function of an element $e \in \mathbf{E}$, input resources r_{in}, and output resources r_{out} with state changes specified in Eq. 4.17.

$$\begin{aligned} \mathcal{R}_{transform} &= \mathcal{R}_{transform}(e, r_{in}, r_{out}) \\ \mathcal{F}_{out}(\mathcal{L}(e)) &\leftarrow \mathcal{F}_{out}(\mathcal{L}(e)) \cup r_{in} \\ \mathcal{F}_{in}(\mathcal{L}(e)) &\leftarrow \mathcal{F}_{in}(\mathcal{L}(e)) \cup r_{out} \end{aligned} \tag{4.17}$$

Similar to the transporting behavior, input and output resources may be functionally defined. For example, a transformation from resources of type τ_1 to resource $r = \{(\tau_2, q_2)\}$ with unit transform demand factor $f_{1|2}$ is represented in Eq. 4.18–4.19.

$$r_{in}(r) = \{(\tau_1, q_2 \cdot f_{1|2}), (\tau_2, 0)\} \tag{4.18}$$

$$r_{out}(r) = \{(\tau_1, 0), (\tau_2, q_2)\} \tag{4.19}$$

Applications of transform demand factor $f_{i|j}$ include modeling electrical energy required to desalinate water, fuel to generate electricity, or general resource production or consumption.

The resource exchanging behavior moves resources across systems boundaries. It is a function of two elements in different system models with a paired destination-origin $e_{ij}, e_{kl} \in \mathbf{E} : i \neq k, \mathcal{L}_d(e_{ij}) = \mathcal{L}_o(e_{kl})$, and resources to exchange r_{ik} with state changes shown in Eq. 4.20.

$$\begin{aligned} \mathcal{R}_{exchange} &= \mathcal{R}_{exchange}(e_{ij}, e_{kl}, r_{ik}) \\ \mathcal{F}_{out}(\mathcal{L}_d(e_{ij})) &\leftarrow \mathcal{F}_{out}(\mathcal{L}_d(e_{ij})) \cup r_{ik} \\ \mathcal{F}_{in}(\mathcal{L}_o(e_{kl})) &\leftarrow \mathcal{F}_{in}(\mathcal{L}_o(e_{kl})) \cup r_{ik} \end{aligned} \tag{4.20}$$

Exchanging is differentiated from transporting and transforming by splitting the state changes between system models. Following Eq. 4.20, system i processes the $\mathcal{F}_{out}$ state changes and system k processes the $\mathcal{F}_{in}$ state changes.

System Model Flow Validity

Resource behaviors affect the flow of resources within and between locations. As the units of spatial aggregation with no resource storage, there must be zero net flow across each location's control boundary at each period of simulation time within a system model. In other words, all resources flowing into or out of a location control boundary must be accounted for in a resource behavior and the total resource flow out $\mathcal{F}_{out}$ equals the total resource flow in $\mathcal{F}_{in}$, as shown in Eq. 4.21.

$$\mathcal{F}_{in}(l) = \mathcal{F}_{out}(l) \; \forall \, l \in \mathbf{L} \tag{4.21}$$

The aggregation of resource flows at locations sets the interface between systems at the location level which, in the limiting cases, may encompass all elements at one location (similar to a system dynamics model), or each element at a separate location (similar to a flow-network model).

System-of-Systems Model Flow Validity

While system models are constrained by resource flows at each location, the system-of-systems model relies on constraining resource exchanging behaviors to consistent values across system models. Consider a resource exchange of r_{ik} between systems i and k. System i processes an exchange behavior of $\mathcal{R}_{exchange}(e_{ij}, e_{kl}, r_{ik})$ and system k processes $\mathcal{R}_{exchange}(e_{kl}, e_{ij}, r_{ki})$. Assuming both system models maintain local validity conditions, the system-of-systems model is valid if and only if $r_{ik} = -r_{ki}$.

4.3.2 Element Behaviors

Element behaviors modify attribute values in correspondence to life-cycle activities. The three element behaviors are storing, transforming, and transporting. Although not presently implemented in the ISoS framework, element exchanging behaviors may be an area of future investigation. This behavior would switch control over element operation between system models, requiring additional model-sharing or interfaces between systems.

The element storing behavior stores elements inside other elements, interpreted as literal cargo or functional attachment. It is a function of two co-located elements $e_i, e_j \in \mathbf{E} : \mathcal{L}(e_i) = \mathcal{L}(e_j)$ with state changes shown in Eq. 4.22.

$$\begin{aligned} \mathcal{E}_{store} &= \mathcal{E}_{store}(e_i, e_j) \\ \mathcal{P}(e_i) &\leftarrow e_j \end{aligned} \tag{4.22}$$

The element transforming behavior changes an element's operational parameters. It is a function of an element $e \in \mathbf{E}$ and one of its allowable states $s \in S_e$ with state changes shown in Eq. 4.23.

$$\begin{aligned} \mathcal{E}_{transform} &= \mathcal{E}_{transform}(e, s) \\ \mathcal{S}(e) &\leftarrow s \end{aligned} \tag{4.23}$$

Transformations between two particular states have special meaning. A transformation from the empty state s_0 to an operational state s represents element commissioning and a transformation from an operational state s to the null state $s_\varnothing$ represents decommissioning.

The element transporting behavior allows mobile elements to change locations, consequently also moving any nested (stored) elements. It is a function of an element which is not stored in another element, i.e. $e \in \mathbf{E} : \mathcal{P}(e) = e$ and an allowable location $l \in \mathbf{L}$, with state changes shown in Eq. 4.24.

$$\begin{aligned} \mathcal{E}_{transport} &= \mathcal{E}_{transport}(e, l) \\ \mathcal{L}(c) &\leftarrow l \ \forall \ c : \mathcal{P}^h(c) = e, h > 0 \end{aligned} \tag{4.24}$$

The recursive definition in Eq. 4.24 updates the location of all elements having a parent (at some level) equal to the element being transported.

4.3.3 Interoperability Interface

Among the ISoS framework concepts presented, only the resource exchanging behavior interacts across infrastructure systems. This behavior forms the core of an interoperability interface which requires coordination of three items:

1. Establish a consistent context model including nodes **N**, locations **L**, and resource types **T**.
2. Send and receive the location $\mathcal{L}(e)$ of all elements capable of resource exchanging behaviors.
3. Send and receive coordinated resource exchange behaviors $\mathcal{R}_{exchange}(e_{ij}, e_{kl}, r_{ik})$ where elements e_{ij} and e_{kl} are controlled by systems $i \neq k$.

The resource exchanging behavior may operate either on physical resources or other information. For example, if one element's behavior relies on the quantity of physical resources stored in another element, this data could be expressed as an information exchanging behavior between the two elements.

Two potential challenges exist for system model interoperability as presented. First, sending and receiving resource exchanging behaviors requires a non-zero amount of real time, preventing instantaneous resource exchange which is often assumed in system-of-system models with shared state. Second, resource exchange between system models having cyclic dependencies can cause inconsistencies due to limits on the number of data exchange periods allowed. These challenges are addressed in the software implementation of an ISoS model discussed in Chapter 5.

4.4 Application Use Cases for Evaluation

This section presents four application cases to evaluate the generality and applicability of the ISoS framework across a range of infrastructure systems. The first case retrospectively studies the assembly and resupply of the International Space Station. The second case study prospectively studies partnerships in future exploration of Mars. The third case introspectively studies resource management in Burkina Faso. Finally, the fourth case prospectively studies national infrastructure planning in Saudi Arabia.

4.4.1 International Space Station Assembly and Resupply

The International Space Station (ISS) is a habitable laboratory environment in low Earth orbit. Although not typically considered an infrastructure system-of-systems due to strong physical dependencies between modules, the encompassing system of assembly and resupply missions managed by international partners and commercial firms can be considered one.

ISS assembly started in November 1998 with the launch of the *Zarya* module by the Russian Federal Space Agency (Roscosmos). Additional components and modules were added by international partners including the U.S. National Aeronautics and Space Administration (NASA), European Space Agency (ESA), Japanese Space Agency (JAXA), and Canadian Space Agency (CSA) over the following 15 years.

Besides component delivery and assembly, missions also rotate crew and resupply critical resources and components. Although environmental control and life-support systems provide some degree of habitat loop closure, regular resupply is required to sustain crew life and operations. NASA's Space Transportation System ("shuttle") served as the primary resupply vehicle up until its retirement in 2010. Today, Roscosmos is responsible for all crew transportation (Soyuz spacecraft) and some resupply missions (Progress spacecraft), ESA and JAXA are responsible for some resupply missions (ATV and HTV spacecraft, respectively), and SpaceX and Orbital Sciences hold contracts for commercial orbital transportation service (COTS) resupply missions (Dragon and Cygnus spacecraft, respectively).

This case applies the ISoS modeling framework to describe historical ISS assembly and resupply missions with the objective of representing the structure and behavior of participating infrastructure including ISS modules and launch vehicles.

Structural Models

The nodes defined in Eq. 4.25 include Cape Canaveral (*CC*, NASA and SpaceX launch site), Mid-Atlantic Regional Spaceport (*MARS*, Orbital launch site), Baikonur Cosmodrome (*BCD*, Roscosmos launch site), Guiana Space Center (*GSC*, ESA launch site), and Tanegashima Space Center (*TSC*, JAXA launch site), the static Space Station Orbit (*SSO*), and the Return Landing Zone (*RLZ*) as an aggregated area for reentry and Earth return.

$$\mathbf{N} = \{CC, MARS, BCD, GSC, TSC, SSO, RLZ\} \tag{4.25}$$

The edge locations illustrated as dashed lines in Fig. 4-5 include the trajectories between launch sites and the space station orbit and between the space station orbit and the landing zone.

Resource types considered in this application are drawn from a logistics classification system in Shull et al. (2006). Resources defined in Eq. 4.26 include classes 101 (cryogenic fuel), 102 (hypergolic fuel), 201 (water consumables), 202 (food consumables), 203 (gas consumables), 401 (spare parts), 501 (cargo containers), 601 (science equipment), 701 (waste), and 703 (failed parts).

$$\mathbf{T} = \{\tau_{101}, \tau_{102}, \tau_{201}, \tau_{202}, \tau_{203}, \tau_{401}, \tau_{501}, \tau_{601}, \tau_{701}, \tau_{703}\} \tag{4.26}$$

Table 4.3 describes ISS element models at two snapshots in time. The first set is during the December 1998 STS-88 mission to deliver and assemble the *Unity* module. The elements include a Proton-K launch vehicle (e_{R1}) and the *Zarya* module (e_{R2}) controlled by Roscosmos (system

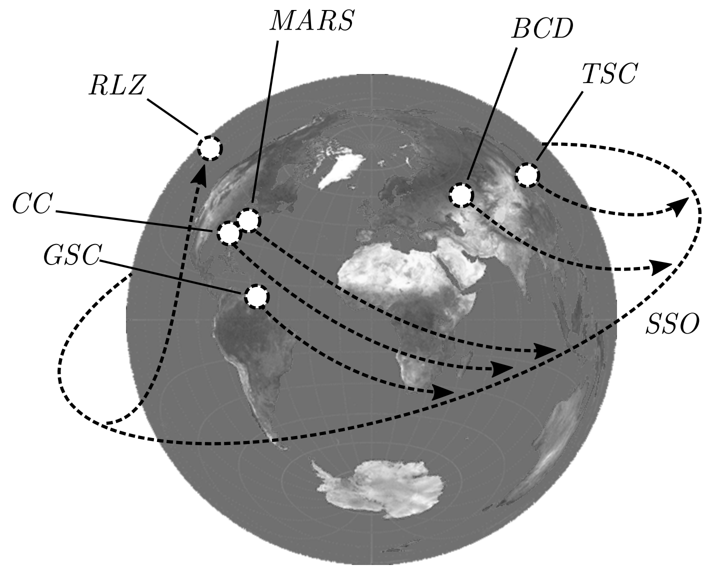

Figure 4-5: ISS network model includes trajectories from each launch site to the station orbit (*SSO*) and the general return trajectory to the landing zone (*RLZ*). (Earth azimuthal projection source images are from *NASA Visual Earth*)

R) and the *Endeavour* orbiter (e_{N1}), *Unity* (e_{N2}) module, and STS-88 crew (e_{N3}) controlled by NASA (system N). Note the Proton-K vehicle is in a null state, identifying it has been staged during launch. The Solid Rocket Boosters (SRBs) and External Tank (ET) are omitted from the *Endeavour* Space Launch System for brevity.

The second set of elements is during the March 2013 SpaceX CRS-2 resupply mission. The new elements include the *Destiny* (e_{N4}) and *Harmony* (e_{N5}) modules controlled by NASA and a Falcon 9 launch vehicle (e_{S1}) and Dragon capsule (e_{S2}) controlled by SpaceX (system S). The other ISS modules and components including the crew are omitted for brevity.

Behavioral Models

Table 4.4 describes behaviors for delivery and assembly of the *Unity* module during the STS-88 mission in December 1998. First, the *Endeavour* orbiter becomes operational (4.27), retrieves the required quantity of fuel (4.28), and burns it (4.29) for transportation to the launch trajectory (4.30). Later, the orbiter arrives at the ISS orbital location (4.31) and commences orbital operations (4.32). While on orbit, the crew perform operations to retrieve (4.33) and consume (4.34) resources, producing produce waste which is stored (4.35) aboard *Endeavour*. The crew resource transformation can be approximated with constant consumption rates for each type (water, gases, and food) and a constant production rate for waste, multiplied by the duration of the mission to determine the values of r_{201}, r_{202}, r_{203}, and r_{701}. Finally, the *Unity* module docks with *Zarya* to complete the assembly (4.36).

Table 4.5 describes behaviors for resource resupply during the SpaceX CRS-2 mission in May 2013. First, the Falcon 9 launch vehicle becomes operational (4.37), retrieves the required quantity of fuel (4.38), and burns it (4.39) to achieve the launch trajectory (4.40). During launch, the

Table 4.3: ISS element instantiations and properties at two snapshots in time

Element	e_{ij}	Contents $\mathcal{C}(e)$	Location $\mathcal{L}(e)$	Parent $\mathcal{P}(e)$	State $\mathcal{S}(e)$
Proton-K	e_{R1}	$\{r_{102}\}$	(BCD, SSO)	e_{R1}	$s_{\varnothing}$
Zarya	e_{R2}	$\{r_{102}, r_{401}\}$	(SSO, SSO)	e_{R2}	$s_{operating}$
Endeavour	e_{N1}	$\{r_{101}, r_{201}, r_{202}, r_{203}, r_{401}, r_{501}, r_{701}\}$	(CC, CC)	e_{N1}	$s_{default}$
Unity	e_{N2}	$\{\}$	(CC, CC)	e_{N1}	$s_{default}$
STS-88 Crew	e_{N3}	$\{\}$	(CC, CC)	e_{N1}	$s_{default}$
⋮	⋮	⋮	⋮	⋮	⋮
Destiny	e_{N4}	$\{r_{201}, r_{202}, r_{203}, r_{501}, r_{701}, r_{703}\}$	(CC, CC)	e_{N2}	$s_{operating}$
Harmony	e_{N5}	$\{\}$	(CC, CC)	e_{N4}	$s_{operating}$
Falcon 9	e_{S1}	$\{r_{101}\}$	(CC, CC)	e_{S1}	$s_{default}$
Dragon	e_{S2}	$\{r_{201}, r_{202}, r_{401}, r_{601}\}$	(CC, CC)	e_{S1}	$s_{default}$

Table 4.4: *Unity* assembly behaviors in December 1998

Behavior	Description	Item
$\mathcal{E}_{transform}(e_{N1}, s_{launching})$	Transform *Endeavour* to launching state	(4.27)
$\mathcal{R}_{store}(e_{N1}, \{\}, \{(\tau_{101}, b)\})$	Retrieves fuel for *Endeavour* burn	(4.28)
$\mathcal{R}_{transform}(e_{N1}, \{(\tau_{101}, b)\}, \{\})$	Transforms fuel for *Endeavour* burn	(4.29)
$\mathcal{E}_{transport}(e_{N1}, (CC, SSO))$	Move *Endeavour* to launch trajectory	(4.30)
$\mathcal{E}_{transport}(e_{N1}, (SSO, SSO))$	Move *Endeavour* to station orbit	(4.31)
$\mathcal{E}_{transform}(e_{N2}, s_{operating})$	Transform *Endeavour* to operating state	(4.32)
$\mathcal{R}_{store}(e_{N1}, \{\}, \{r_{201}, r_{202}, r_{203}\})$	Retrieve consumables from *Endeavour*	(4.33)
$\mathcal{R}_{transform}(e_{N3}, \{r_{201}, r_{202}, r_{203}\}, \{r_{701}\})$	Transform consumable resources to waste with STS-88 Crew	(4.34)
$\mathcal{R}_{store}(e_{N1}, \{r_{701}\}, \{\})$	Store crew waste in *Endeavour*	(4.35)
$\mathcal{E}_{store}(e_{N2}, e_{R2})$	Dock *Unity* with *Zarya*	(4.36)

Table 4.5: Dragon resupply behaviors in May 2013

Behavior	Description	Item
$\mathcal{E}_{transform}(e_{S1}, s_{launching})$	Transform Falcon 9 to launching state	(4.37)
$\mathcal{R}_{store}(e_{S1}, \{\}, \{(\tau_{101}, b)\})$	Retrieve fuel for Falcon 9 burn	(4.38)
$\mathcal{R}_{transform}(e_{S1}, \{(\tau_{101}, b)\}, \{\})$	Transform fuel for Falcon 9 burn	(4.39)
$\mathcal{E}_{transport}(e_{S1}, (CC, SSO))$	Move Falcon 9 to launch trajectory	(4.40)
$\mathcal{E}_{store}(e_{S2}, e_{S2})$	Separate Dragon from Falcon 9	(4.41)
$\mathcal{E}_{transform}(e_{S1}, s_{\varnothing})$	Stage (discard) Falcon 9	(4.42)
$\mathcal{E}_{transport}(e_{S2}, (SSO, SSO))$	Move Dragon to station orbit	(4.43)
$\mathcal{E}_{store}(e_{S2}, e_{N5})$	Dock Dragon with *Harmony*	(4.44)
$\mathcal{E}_{transform}(e_{S2}, s_{operating})$	Transform Dragon to operating state	(4.45)
$\mathcal{R}_{store}(e_{S2}, \{\}, \{r_{201}, r_{202}, r_{203}, r_{401}, r_{601}\})$	Retrieve resupply resources from Dragon	(4.46)
$\mathcal{R}_{exchange}(e_{S2}, e_{N4}, \{r_{201}, r_{202}, r_{203}, r_{401}, r_{601}\})$	Exchange resupply resources with *Destiny*	(4.47)
$\mathcal{R}_{exchange}(e_{N4}, e_{S2}, \{r_{601}, r_{703}\})$	Exchange return resources to Dragon	(4.48)
$\mathcal{R}_{store}(e_{S2}, \{r_{601}, r_{703}\}, \{\})$	Store return resources in Dragon	(4.49)

Dragon capsule separates from the Falcon 9 (4.41), which is then staged (4.42). When the Dragon arrives on orbit (4.43), it docks with *Harmony* (4.44) and exchanges resupply resources including water, food, gases, spare parts, and science equipment with the *Destiny* module (4.47). As the CRS-2 mission also includes return mass capability, the *Destiny* module exchanges resources such as science equipment and failed parts to Dragon (4.48) to be stored for return (4.49).

4.4.2 Partnerships for Mars Space Exploration

Future space missions seek to explore distant locations such as near-Earth asteroids and Mars. Underlying the technical challenges of vehicle and mission design there are two key logistics challenges. First, travel to distant locations requires large amounts of propellant per unit mass carried due to the physics of rocket propulsion.[2] Second, the travel happens over long durations with limited opportunities for resupply, potentially requiring large quantities of contingency resources to be carried along at great expense.

One strategy to improve exploration performance is to close the resource loop and achieve higher self-sufficiency of remote operations. Advanced life support systems, *in-situ* resource production, and storage depots reduce reliance on resupply but also introduce additional interdependencies between elements and missions. Furthermore, there is active interest in enabling multi-national and commercial enterprises supporting future space exploration (Griffin 2011). A core principle of the U.S. National Space Policy states that "a robust and competitive commercial space sector is vital to continued progress in space" (United States 2010). In other words, future space exploration may involve an infrastructure system-of-systems with interdependencies between elements.

This case applies the ISoS modeling framework to describe structural and behavioral models for a conceptual mission to Mars including both NASA and commercial participation. A future

[2] Estimates of the "propellant-to-mass" ratio for Mars exploration using conventional chemical propulsion require more than 200 kilograms of propellant for every kilogram of mass returned to Earth (Grogan and de Weck 2012).

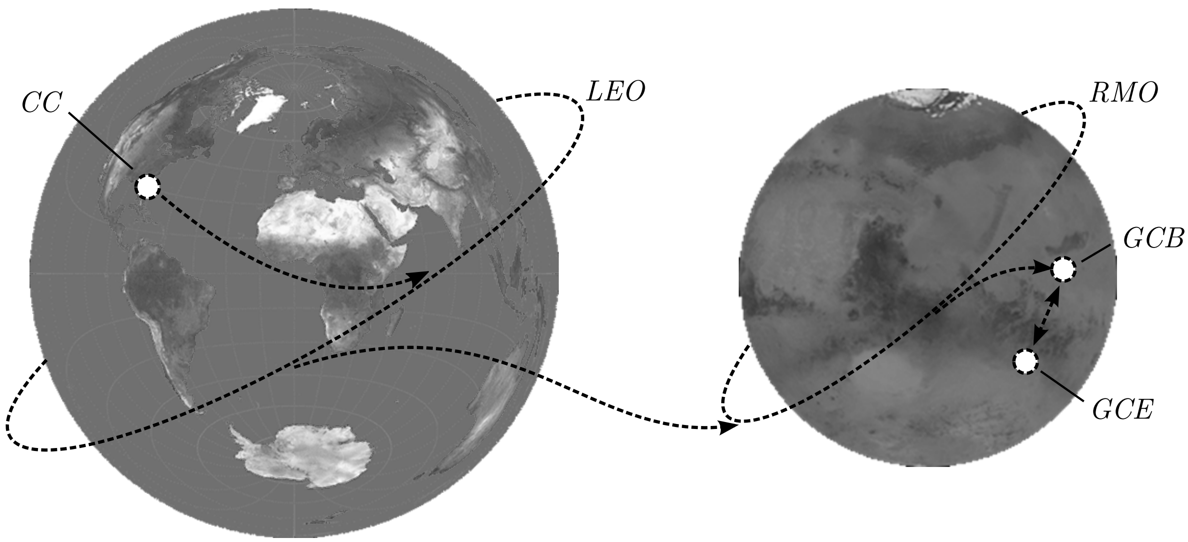

Figure 4-6: The Mars exploration network model includes Cape Canaveral (CC), Low Earth Orbit (LEO), Reference Mars Orbit (RMO), Gale Crater Base (GCB) and Gale Crater Excavation (GCE; location not to scale) with edges based on trajectories and surface paths. (Earth and Mars azimuthal projection source images are from *NASA Visual Earth* and NASA/JPL-Caltech)

model implementation may help plan long-duration campaigns with high element re-use and identify opportunities of commercial partnerships for resource exchange.

Structural Models

The nodes of interest defined in Eq. 4.50 include surface areas such as Cape Canaveral (CC), Gale Crater Base (GCB), and Gale Crater Excavation (GCE) and stable locations in space such Low Earth Orbit (LEO) and Reference Mars Orbit (RMO).

$$\mathbf{N} = \{CC, LEO, RMO, GCB, GCE\} \tag{4.50}$$

Figure 4-6 illustrates the set of allowable locations. Edge locations (CC, LEO), (LEO, RMO), and (RMO, GCB) are propulsive trajectories for space transportation and (GCB, GCE) and (GCE, GCB) are paths for surface transportation.

The resource types considered in this application are drawn from a logistics classification system in Shull et al. (2006). Resources defined in Eq. 4.51 include classes 101 (cryogenic fuel), 102 (hypergolic fuel), 201 (water consumables), 202 (food consumables), 203 (gas consumables), 401 (spare parts), 603 (exploration samples), 701 (waste), and 703 (failed parts).

$$\mathbf{T} = \{\tau_{101}, \tau_{102}, \tau_{201}, \tau_{202}, \tau_{203}, \tau_{401}, \tau_{603}, \tau_{701}, \tau_{703}\} \tag{4.51}$$

Figure 4-7 illustrates element models at the Gale Crater exploration site. The elements include a transfer vehicle (e_{A1}), lander (e_{A2}), habitat (e_{A3}), and astronaut (e_{A4}) controlled by a national space agency (system A) and a resource plant (e_{C1}) and *in-situ* regolith (e_{C2}) controlled by a

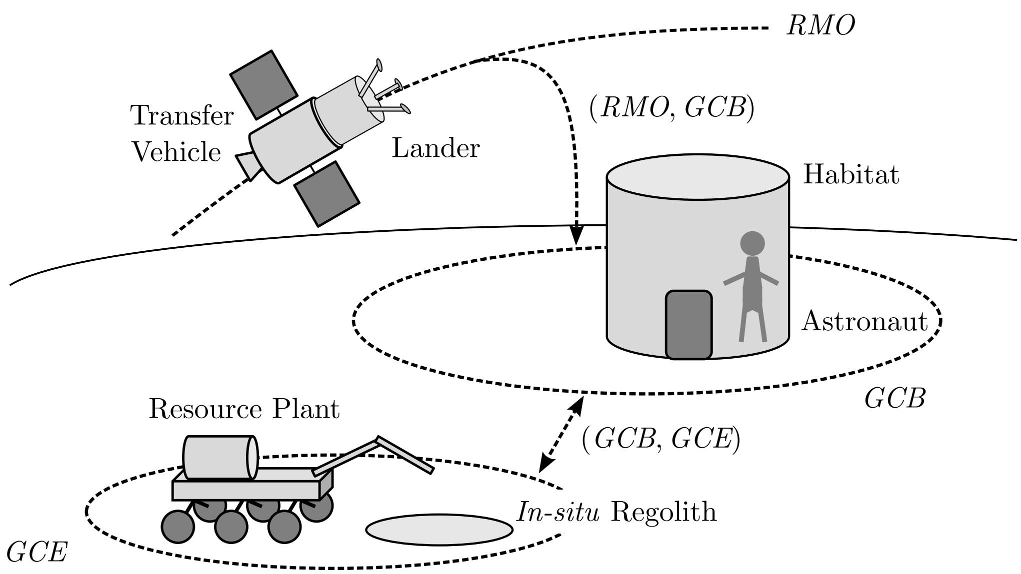

Figure 4-7: Mars exploration elements include vehicles in orbit and habitats, astronauts, resource plants, and natural *in-situ* regolith at surface sites

Table 4.6: Mars exploration element instantiations and properties

Element	e_{ij}	Contents $\mathcal{C}(e)$	Location $\mathcal{L}(e)$	Parent $\mathcal{P}(e)$	State $\mathcal{S}(e)$
Transfer Vehicle	e_{A1}	$\{r_{101}\}$	(RMO, RMO)	e_{A1}	$s_{default}$
Lander	e_{A2}	$\{r_{102}, r_{201}, r_{202}\}$	(RMO, RMO)	e_{A1}	$s_{waiting}$
Habitat	e_{A3}	$\{r_{201}, r_{202}, r_{203},$ $r_{401}, r_{701}, r_{703}\}$	(GCB, GCB)	e_{A3}	$s_{default}$
Astronaut	e_{A4}	$\{\}$	(GCB, GCB)	e_{A3}	$s_{default}$
Resource Plant	e_{C1}	$\{r_{203}\}$	(GCE, GCE)	e_{C1}	$s_{processing}$
In-situ Regolith	e_{C2}	$\{r_{603}\}$	(GCE, GCE)	e_{C2}	$s_{default}$

commercial firm (system C). Table 4.6 illustrates the elements' properties in detail. Each element except the astronaut stores at least one type of resource. For example, the lander contains hypergolic propellant (r_{102}) for landing and water (r_{201}) and food (r_{202}) consumables for resupply. Note the lander is functionally stored inside the transfer vehicle and the astronaut actually inside the habitat.

Behavioral Models

Table 4.7 describes the landing sequence of the lander vehicle. First, the lander un-docks from the transfer vehicle (4.52) which provided transportation from Earth and enters a landing state (4.53). Next, the lander retrieves stored fuel (4.54) and burns it (4.55) for propulsion to move from a stationary orbit RMO to a descent trajectory (RMO, GCB). The quantity of fuel required, b_1, depends on the total mass of the lander and its propulsion system efficiency. Finally, the lander retrieves (4.57) and burns (4.58) more fuel to complete the landing maneuver and arrive (4.59) at the base site GCB. The quantity of fuel required, b_2, depends on the remaining mass of the lander.

Table 4.7: Mars landing sequence behaviors

Behavior	Description	Item
$\mathcal{E}_{store}(e_{A2}, e_{A2})$	Un-dock lander from transfer vehicle	(4.52)
$\mathcal{E}_{transform}(e_{A2}, s_{landing})$	Transform lander to landing state	(4.53)
$\mathcal{R}_{store}(e_{A2}, \{\}, \{(\tau_{102}, b_1)\})$	Retrieve fuel for 1st burn	(4.54)
$\mathcal{R}_{transform}(e_{A2}, \{(\tau_{102}, b_1)\}, \{\})$	Transform fuel for 1st burn with lander	(4.55)
$\mathcal{E}_{transport}(e_{A2}, (RMO, GCB))$	Move lander to landing trajectory	(4.56)
$\mathcal{R}_{store}(e_{A2}, \{\}, \{(\tau_{102}, b_2)\})$	Retrieve fuel for 2nd burn	(4.57)
$\mathcal{R}_{transform}(e_{A2}, \{(\tau_{102}, b_2)\}, \{\})$	Transform fuel for 2nd burn with lander	(4.58)
$\mathcal{E}_{transport}(e_{A2}, (GCB, GCB))$	Move lander to base	(4.59)
$\mathcal{E}_{transform}(e_{A2}, s_{default})$	Transform lander to default state	(4.60)

Table 4.8: Mars resource production and delivery behaviors

Behavior	Description	Item
$\mathcal{R}_{store}(e_{C2}, \{\}, \{(\tau_{203}, q_{603})\})$	Retrieve samples from *in-situ* regolith	(4.61)
$\mathcal{R}_{transform}(e_{C1}, \{(\tau_{603}, q_{603})\}, \{(\tau_{203}, q_{203})\})$	Transform samples to consumable gases with resource plant	(4.62)
$\mathcal{R}_{store}(e_{C1}, \{(\tau_{203}, q_{203})\}, \{\})$	Store gases in resource plant	(4.63)
$\mathcal{E}_{transform}(e_{C1}, s_{moving})$	Transform plant to transport state	(4.64)
$\mathcal{E}_{transport}(e_{C1}, (GCE, GCB))$	Move plant to the base path	(4.65)
$\mathcal{E}_{transport}(e_{C1}, (GCE, GCE))$	Move plant to the base	(4.66)
$\mathcal{E}_{transform}(e_{C1}, s_{default})$	Transform plant to default state	(4.67)
$\mathcal{R}_{exchange}(e_{C1}, e_{A3}, \{(\tau_{203}, q_{203})\})$	Exchange gases with habitat	(4.68)

Table 4.8 describes the resource production and delivery behaviors of the resource plant. First, samples are retrieved from the *in-situ* regolith (4.61) and are transformed to consumable gases (4.62) by the resource plant. The transformation ratio may be approximated by a factor $f_{603|203}$ such that $q_{603} = f_{603|203} \cdot q_{203}$. Next, the plant stores the gases (4.63) and moves (4.65) from the excavation site GCE to the path to the base (GCE, GCB). Finally, the resource plant arrives at the base GCB (4.66) and exchanges gases with the habitat (4.68).

4.4.3 Resource Management in Burkina Faso

Burkina Faso is a landlocked country in western Africa bordering the Sahara desert to the north and a fertile region to the south. It was the subject of a past case study by Hermann et al. (2012) to investigate inker-linkages between climate, land, energy, and water for national policy which forms the basis of this application. Burkina Faso is an impoverished nation with a population that doubled between 1985 and 2011 and is currently growing at 3% per year. Without sea access, it survives on limited natural resources and trade with nearby countries.

Agriculture in Burkina Faso is a large part of the economy with cereals as the staple diet and cotton as the primary export crop. While the majority of people work in agriculture, productivity is low due to lack of mechanization and limited use of irrigation and fertilizer. Securing nourishment for the populace is a continuing challenge while agricultural land is starting to encroach on forested land due to low output.

The majority of energy consumption in Burkina Faso is derived from biomass such as wood for residential use. Electrification of the country is sparse and improvement faces significant capacity limitations. Most electricity is generated from thermal power plants using imported fossil fuels, although there are also hydro power plants. Interconnecting Burkina Faso's grid with the West African Power Pool (WAPP) has recently enabled direct electricity import.

Water is plentiful at times via the numerous rivers, although there are frequent droughts and floods which may be exasperated by climate change. The southern regions receive nearly twice as much rainfall as the northern regions. Most water is drawn from surface reservoirs from rivers. A large portion of the country under-utilizes groundwater at rates much lower than the natural recharge rate.

This case applies the ISoS modeling framework to describe present natural and artificial infrastructure in Burkina Faso with an emphasis on modeling the structure and behavior of agriculture, water, and energy elements.

Structural Models

The nodes of interest for Burkina Faso defined in Eq. 4.69 are centered on water resources to include the five river basins: Comoe Basin (CMB), Mouhoun Basin (MHB), Nakambe Basin (NKB), Niger Basin (NGB), and Oti Basin (OTB).

$$\mathbf{N} = \{CMB, MHB, NKB, NGB, OTB\} \tag{4.69}$$

The river basins are selected as nodes due to their shared water properties, however more detailed network models may also consider urban areas or climate patterns like rainfall to create additional nodes. Figure 4-8 illustrates the set of allowable locations including edges between adjacent river basins.

Resource types defined in Eq. 4.70 include biomass fuel (τ_b), fossil fuel (τ_{ff}), electrical energy (τ_e), water (τ_w), food cereals (τ_{fc}), and cotton (τ_c).

$$\mathbf{T} = \{\tau_b, \tau_{ff}, \tau_e, \tau_w, \tau_{fc}, \tau_c\} \tag{4.70}$$

Table 4.9 describes fixed elements at Nakambe Basin as an example of spatial infrastructure within one river basin. Additional infrastructure elements could describe resource transportation between basin regions or provide more detailed models of the aggregated systems, for example to illustrate rainfall variations in different cities within Nakambe Basin.

The agriculture sector (system A) includes natural forests (e_{A1}) as a store of biomass resources and combined cereals and cotton farms (e_{A2}) to aggregate all farms in the basin. The water sector (system W) includes two aggregated elements as water sources: surface water reservoirs (e_{W1}) from rivers and groundwater aquifers (e_{W2}). The energy sector (system E) includes aggregated elements for fossil-fuel thermal power plants (e_{E1}) and renewable hydro power plants (e_{E2}). The public (system P) includes an aggregated element to represent the societies within the basin. Finally, an international sector (system I) includes a global market element to facilitate resource import

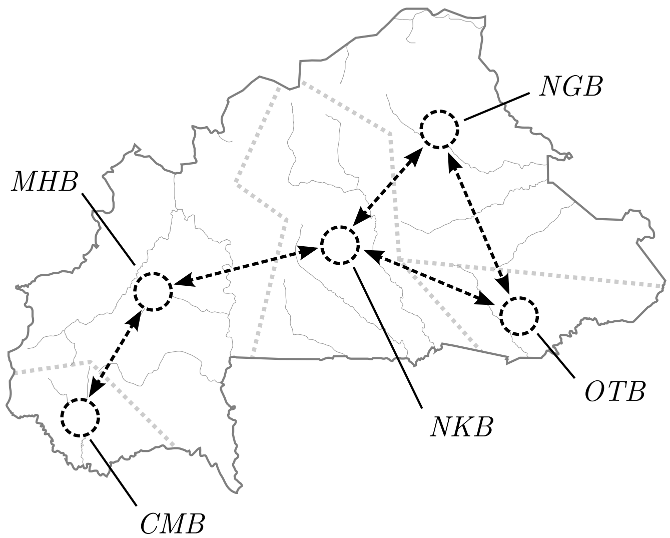

Figure 4-8: The Burkina Faso network model includes the five river basins and edge locations based on physical adjacency. Dotted gray lines indicate notional basin boundaries and solid gray lines represent rivers. Map credit: d-maps.com accessed at `http://d-maps.com/m/africa/burkina/burkina46.pdf`

and export. Although omitted in this example, currency flows would be involved in bi-directional resource exchanges with the global market.

Behavioral Models

Table 4.10 describes typical behaviors for the agriculture system. First, water is exchanged from surface reservoirs (4.71). The irrigation water is transformed to produce food cereals and cotton (4.72). The food cereals are exchanged with the societies (4.73) as food while the cotton resources are exchanged with the global market (4.74) for export. Biomass is also retrieved from natural forests (4.75) and exchanged with the societies (4.76). The natural forests regrow (4.77) a fraction β of the harvested biomass which is stored (4.78) as new biomass.

Table 4.9: Nakambe Basin element instantiations and properties

Element	e_{ij}	Contents $\mathcal{C}(e)$	Location $\mathcal{L}(e)$	Parent $\mathcal{P}(e)$	State $\mathcal{S}(e)$
Natural Forests	e_{A1}	$\{r_b\}$	(NKB, NKB)	e_{A1}	$s_{default}$
Cereals & Cotton Farms	e_{A2}	$\{\}$	(NKB, NKB)	e_{A2}	$s_{operating}$
Surface Water Reservoirs	e_{W1}	$\{r_w\}$	(NKB, NKB)	e_{W1}	$s_{default}$
Groundwater Aquifer	e_{W2}	$\{r_w\}$	(NKB, NKB)	e_{W2}	$s_{default}$
Thermal Power Plants	e_{E1}	$\{\}$	(NKB, NKB)	e_{E1}	$s_{operating}$
Hydro Power Plants	e_{E2}	$\{\}$	(NKB, NKB)	e_{E2}	$s_{operating}$
Nakambe Basin Societies	e_{P1}	$\{\}$	(NKB, NKB)	e_{P1}	$s_{default}$
Global Market	e_{I1}	$\{\}$	(NKB, NKB)	e_{I1}	$s_{default}$

Table 4.10: Nakambe Basin agriculture production behaviors

Behavior	Description	Item
$\mathcal{R}_{exchange}(e_{W1}, e_{A2}, \{\}, \{(\tau_w, q_w)\})$	Exchange water withdrawn from reservoirs	(4.71)
$\mathcal{R}_{transform}(e_{A2}, \{(\tau_w, q_w)\}, \{(\tau_{fc}, q_{fc}), (\tau_c, q_c)\})$	Transform water to cereals and cotton	(4.72)
$\mathcal{R}_{exchange}(e_{A2}, e_{P1}, \{(\tau_{fc}, q_{fc})\})$	Exchange food cereals to societies	(4.73)
$\mathcal{R}_{exchange}(e_{A2}, e_{I1}, \{(\tau_c, q_c)\})$	Exchange cotton to market	(4.74)
$\mathcal{R}_{store}(e_{A1}, \{\}, \{(\tau_b, q_b)\},$	Retrieve biomass from forests	(4.75)
$\mathcal{R}_{exchange}(e_{A1}, e_{P1}, \{(\tau_b, q_b)\})$	Exchange biomass to societies	(4.76)
$\mathcal{R}_{transform}(e_{A1}, \{\}, \{(\tau_b, q_b \cdot \beta)\})$	Transform to regrow biomass	(4.77)
$\mathcal{R}_{store}(e_{A1}, \{(\tau_b, q_b \cdot \beta)\}, \{\})$	Store new biomass in forests	(4.78)

Table 4.11: Nakambe Basin energy production and import behaviors

Behavior	Description	Item
$\mathcal{R}_{exchange}(e_{I1}, e_{E1}, \{\}, \{(\tau_s, q_s)\})$	Exchange imported petroleum fuel from market	(4.79)
$\mathcal{R}_{transform}(e_{E1}, \{(\tau_s, q_s)\}, \{(\tau_e, q_{e1}))\})$	Transform petroleum to electricity from thermal plants	(4.80)
$\mathcal{R}_{transform}(e_{E2}, \{\}, \{(\tau_e, q_{e2})\})$	Transform electricity from hydro power plants	(4.81)
$\mathcal{R}_{exchange}(e_{I1}, e_{E1}, \{(\tau_e, q_{e3})\})$	Exchange imported electricity from market	(4.82)
$\mathcal{R}_{exchange}(e_{E1}, e_{P1}, \{(\tau_e, q_e)\})$	Exchange net electricity to societies	(4.83)

Table 4.11 describes typical behaviors for the energy system. Petroleum fuels are exchanged from the global market (4.79) as import. The fuels are transformed by thermal power plants (4.80) to produce electricity. Hydro plants also transform electricity (4.81) as a second source. Any shortfall in electricity is exchanged from the global market (4.82) as import. Finally, the total electricity $q_e = q_{e1} + q_{e2} + q_{e3}$ is exchanged to the societies (4.83) for use.

4.4.4 National Infrastructure Planning in Saudi Arabia

The Arabian Peninsula has a hot, arid climate which historically limited the expanse of human societies. Since the 1950s the Kingdom of Saudi Arabia has experienced strong economic growth driven in part by use of large natural petroleum reserves. Cities such as Riyadh have grown from around 50,000 inhabitants in the 1950s to over 6.2 million in 2009 and is currently growing at 2.7% annually (KSA 2010). At the same time, per capita demands for basic resources such as potable water, electricity for cooling, and food are increasing as individuals reach higher standards of living.

There are strong couplings between infrastructure based on geographical, technical, and social factors in Saudi Arabia. For example, Riyadh is positioned at 600 meters in elevation in the center of the peninsula, 400 kilometers from the Persian (Arabian) Gulf and 900 kilometers from the Red Sea. Facing a shortage of potable water from nonrenewable aquifers, its population relies in part on desalinated sea water pumped from the coast, an energy-intensive production and distribution method. With limited electrical generation and distribution capacity and low domestic petroleum prices, some areas directly burn crude oil for electricity which is both environmentally-damaging and reduces export capacity.

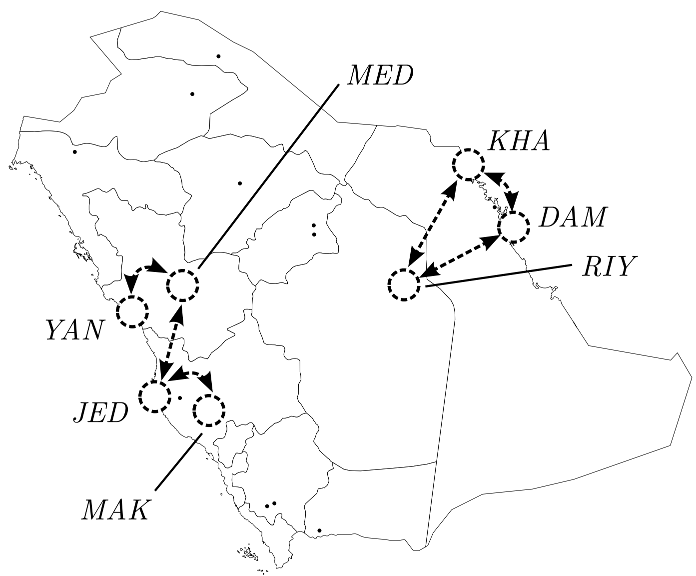

Figure 4-9: The Saudi Arabia network model includes nodes for the major cities of Khafji, Dammam, Riyadh, Makkah, Medina, Yanbu, and Jeddah and edges based on physical adjacency. Gray lines identify regional provincial boundaries and black dots are large cities. Map credit: d-maps.com, accessed at `http://d-maps.com/m/arabie/arabie20.pdf`.

To meet growing demands for basic resources, the Ninth Development Plan seeks to appropriate 1.4 trillion SAR ($370 billion USD) for development between 2010 and 2014 (KSA 2010). High-level objectives call for improving the standard of living and quality of life, achieving balanced development among regions, and diversifying the economic base beyond petroleum products. Although Saudi Arabia has a centrally-managed government, national infrastructure planning still involves multiple government ministries and regional administrative divisions acting as a system-of-systems.

This case applies the ISoS modeling framework to describe national infrastructure in Saudi Arabia with emphasis on the energy and water systems. A future model implementation could be used for collaborative development of national infrastructure plans across government ministries.

Structural Models

The nodes of interest for the context of Saudi Arabia include the major geo-spatial areas and supporting infrastructure regions. The nodes defined in Eq. 4.84 target coastal cities with desalination operations such as Jeddah (JED), Yanbu (YAN), Dammam (DAM), and Khafji (KHA) and interior cities receiving pumped water such as Makkah (MAK), Medina (MED), and Riyadh (RIY).

$$\mathbf{N} = \{JED, MAK, MED, YAN, RIY, DAM, KHA\} \tag{4.84}$$

The set of allowable locations shown in Fig. 4-9 includes edges for physically adjacent nodes where distribution lines may feasibly exist.

Resource types defined in Eq. 4.85 include oil (τ_o), electrical energy (τ_e), water (τ_w), people (τ_p), and currency (τ_c).

$$\mathbf{T} = \{\tau_o, \tau_e, \tau_w, \tau_p, \tau_c\} \tag{4.85}$$

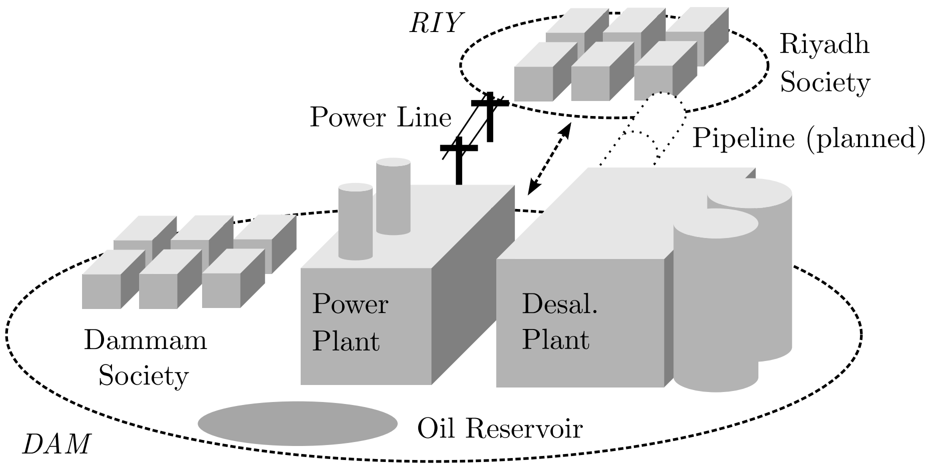

Figure 4-10: Saudi infrastructure elements include natural oil reservoirs, power and water plants with fixed distribution lines, and public societies.

Table 4.12: Saudi Arabia element instantiations and properties

Element	e_{ij}	Contents $\mathcal{C}(e)$	Location $\mathcal{L}(e)$	Parent $\mathcal{P}(e)$	State $\mathcal{S}(e)$
Oil Reservoir	e_{E1}	$\{r_o\}$	(DAM, DAM)	e_{E1}	$s_{default}$
Power Plant	e_{E2}	$\{\}$	(DAM, DAM)	e_{E2}	$s_{operating}$
Power Line	e_{E3}	$\{\}$	(DAM, RIY)	e_{E3}	$s_{operating}$
Desalination Plant	e_{W1}	$\{\}$	(DAM, DAM)	e_{W1}	$s_{operating}$
Water Pipeline	e_{W2}	$\{\}$	(DAM, RIY)	e_{W2}	s_0
Dammam Society	e_{P1}	$\{r_p, r_c\}$	(DAM, DAM)	e_{S1}	$s_{default}$
Riyadh Society	e_{P2}	$\{r_p, r_c\}$	(RIY, RIY)	e_{S2}	$s_{default}$

More detailed framings could consider different types of water (potable, wastewater, seawater, etc.), agriculture and food, and carbon dioxide and other pollutants.

Figure 4-10 illustrates element model instantiations focusing on the Dammam-Riyadh region. The elements include an oil reservoir (e_{E1}), power plant (e_{E2}), and power line (e_{E3}) controlled by an energy ministry (system E), a desalination plant (e_{W1}) and pipeline (e_{W2}) controlled by a water ministry (system W), and societies of Dammam (e_{P1}) and Riyadh (e_{P2}) which are organized under the public (system P). Here, societies aggregate the residential, commercial, and industrial activities which consume resources and generate socio-economic activity.

Table 4.12 illustrates the elements' properties in detail. Due to the large scale of aggregation, no elements are stored within others and only the oil reservoir and two societies contain resources—all other resources are transported, transformed, and exchanged. The power line and pipeline elements are fixed at the (DAM, RIY) location, allowing transportation of resources between the two nodes. Note the water pipeline state is empty (s_0), meaning it has not yet become operational.

Table 4.13: Saudi Arabia energy generation and distribution behaviors

Behavior	Description	Item
$\mathcal{R}_{store}(e_{E1}, \{\}, \{(\tau_o, q_o)\})$	Retrieve oil from reservoir	(4.86)
$\mathcal{R}_{exchange}(e_{W2}, e_{E2}, \{(\tau_w, q_w)\})$	Exchange cooling water to power plant	(4.87)
$\mathcal{R}_{transform}(e_{E2}, \{(\tau_o, q_o), (\tau_w, q_w)\}, \{(\tau_e, q_e)\})$	Transform oil and water to electricity with plant	(4.88)
$\mathcal{R}_{exchange}(e_{E2}, e_{P1}, \{(\tau_e, q_{e1})\})$	Exchange electricity with Dammam society	(4.89)
$\mathcal{R}_{transport}(e_{E3}, \{(\tau_e, q_{e2})\}, \{(\tau_e, q_{e2} \cdot \eta_e)\})$	Transport electricity to Riyadh with power line	(4.90)
$\mathcal{R}_{exchange}(e_{E3}, e_{P2}, \{(\tau_e, q_{e2} \cdot \eta_e)\})$	Exchange electricity to Riyadh society	(4.91)

Table 4.14: Saudi Arabia water production and distribution behaviors

Behavior	Description	Item
$\mathcal{R}_{exchange}(e_{E2}, e_{W1}, \{(\tau_e, q_{e1})\})$	Exchange electricity to desal. plant	(4.92)
$\mathcal{R}_{transform}(e_{W1}, \{(\tau_e, q_{e1})\}, \{(\tau_w, q_w)\})$	Transform electricity into water	(4.93)
$\mathcal{R}_{exchange}(e_{W1}, e_{P1}, \{(\tau_w, q_{w1})\})$	Exchange water with Dammam society	(4.94)
$\mathcal{E}_{transform}(e_{W2}, s_{operating})$	Transform pipeline to operational	(4.95)
$\mathcal{R}_{exchange}(e_{E2}, e_{W2}, \{(\tau_e, q_{e2})\})$	Exchange electricity with pipeline	(4.96)
$\mathcal{R}_{transport}(e_{W2}, \{(\tau_w, q_{w2}), (\tau_e, q_{e2})\}, \{(\tau_w, q_{w2} \cdot \eta_w)\})$	Transport water to Riyadh with pipeline	(4.97)
$\mathcal{R}_{exchange}(e_{W2}, e_{P2}, \{(\tau_w, q_{w2} \cdot \eta_w)\})$	Exchange water to Riyadh society	(4.98)

Behavioral Models

Table 4.13 describes the behaviors for electricity generation and distribution. First, oil is retrieved from the reservoir (4.86) and the desalination plant exchanges cooling water with the power plant (4.87). The power plant transforms oil as fuel and water for cooling to electricity (4.88). The quantities of oil and water required can be estimated with constant factors $f_{o|e}$ and $f_{w|e}$ respectively. Next, the power plant exchanges (4.89) a portion of its generated electricity q_{e1} with the local society (Dammam) to satisfy demands and produce socio-economic activity. The remaining electricity $q_{e2} = q_e - q_{e1}$ is transported (4.90) to Riyadh with efficiency η_e due to resistive losses. Finally, the power line exchanges (4.91) electricity to the Riyadh society.

Table 4.14 describes the behaviors for water production and distribution. First, the power plant exchanges electricity to the desalination plant (4.92). Next, the desalination plant transforms electricity into water (4.93). The amount of electrical energy required can be approximated with a constant factor $f_{e1|w1}$. Next, the desalination plant exchanges water with the local (Dammam) society (4.94). At this time, the pipeline becomes operational (4.95). The power plant exchanges electricity with the pipeline (4.96) for pumping energy to transport water (4.97) to Riyadh. The amount of electricity here can also be estimated with a constant factor $f_{e2|w2}$, and the efficiency of water transport after leakage can be estimated at η_w. Finally, the pipeline exchanges water to the Riyadh society (4.98).

References for Chapter 4

Bellman, R. (1958). “On a routing problem”. In: *Quarterly of Applied Mathmatics* 16, pp. 87–90.

Berger, E., H. Boulay, and B. Zisk (1970). “Simulation and the City: A Critical Overview”. In: *Simulation & Gaming* 1.4, pp. 411–428. DOI: 10.1177/104687817000100404.

Board, D. S. (1996). *Information Warfare – Defense.* ADA319571. Department of Defense. URL: http://handle.dtic.mil/100.2/ADA319571 (visited on 08/14/2013).

Bush, B., L. Dauelsberg, R. LeClaire, D. Powell, S. DeLand, and M. Samsa (2005). “Critical Infrastructure Protection Decision Support System (CIP/DSS) Project Overview”. In: *2005 International System Dynamics Conference*. LA-UR-05-1870. Boston, Massachusetts.

Dantzig, G. B. (1963). *Linear Programming and Extensions.* Priceton, New Jersey: Priceton University Press.

de Weck, O. L., D. Roos, and C. Magee (2012). *Engineering Systems: Meeting Human Needs in a Complex Technological World*. Cambridge, Massachusetts: MIT Press.

Dijkstra, E. W. (1959). “A note on two problems in connexion with graphs”. In: *Numerische Mathematik* 1, pp. 269–271. DOI: 10.1007/BF01386390.

Dudenhoeffer, D. D., M. R. Permann, and M. Manic (2006). “CIMS: A Framework for Infrastructure Interdependency Modeling and Analysis”. In: *Proceedings of the 2006 Winter Simulation Conference*.

Edwards, J. and R. M. Karp (1972). “Theoretical improvements in algorithmic efficiency for network flow problems”. In: *Journal of the ACM* 19.2, pp. 248–264. DOI: 10.1145/321694.321699.

Ellison, J. (2006). “Modeling the US Natural Gas Network”. In: *IIE Annual Conference and Exposition*. Orlando, Florida.

Ford, L. R. (1956). *Network Flow Theory.* P-923. Santa Monica, California: The RAND Corporation.

Griffin, M. D. (2011). “Enabling Complementary Commercial and Government Enterprises in Space”. In: *62nd International Astronautical Congress*. IAC-11.E3.4.6. Cape Town, South Africa.

Grogan, P. T. and O. L. de Weck (2012). “Federated Simulation and Gaming Framework for a Decentralized Space-based Resource Economy”. In: *ASCE Earth and Space 2012 Conference*. Pasadena, California.

Hermann, S., M. Welsch, R. E. Segerstrom, M. I. Howells, C. Young, T. Alfstad, H.-H. Rogner, and P. Steduto (2012). “Climate, land, energy and water (CLEW) interlinkages in Burkina Faso: An analysis of agricultural intensification and bioenergy production”. In: *Natural Resources Forum* 36.4, pp. 245–262. DOI: 10.1111/j.1477-8947.2012.01463.x.

Hu, T. C. (1963). “Multi-commodity Network Flows”. In: *Operations Research* 11.3, pp. 344–360. DOI: 10.1287/opre.11.3.344.

IEEE (1998). *IEEE Standard for Distributed Interactive Simulation – Application Protocols*. IEEE Std. 1278.1a-1998.

— (2010). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*. IEEE Std. 1516-2010.

Kingdom of Saudi Arabia (2010). *The Ninth Development Plan.* Riyadh: Ministry of Economy and Planning. URL: http://www.mep.gov.sa/ (visited on 07/14/2013).

Min, H.-S. J., W. Beyeler, T. Brown, Y. J. Son, and A. T. Jones (2007). “Toward modeling and simulation of critical national infrastructure interdependencies”. In: *IIE Transactions* 39.1, pp. 57–71. DOI: 10.1080/07408170600940005.

O'Rourke, T. D. (2007). "Critical Infrastructure, Interdependencies, and Resilience". In: *The Bridge* 37.1, pp. 22–29.

Pederson, P., D. Dudenhoeffer, S. Hartley, and M. Permann (2006). *Critical Infrastructure Interdependency Modeling: A Survey of U.S. and International Research.* Tech. rep. INL/EXT-06-11464. Idaho Falls, Idaho: Idaho National Laboratory.

President's Commission on Critical Infrastructure Protection (1997). *Critical Foundations: Protecting America's Infrastructures.* URL: https://www.fas.org/sgp/library/pccip.pdf (visited on 03/11/2013).

Rinaldi, S. M., J. P. Peerenboom, and T. K. Kelly (2001). "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies". In: *IEEE Control Systems Magazine*, pp. 11–25.

Shull, S., E. Gralla, O. de Weck, and R. Shishko (2006). "The Future of Asset Management for Human Space Exploration: Supply Classification and an Interplanetary Supply Chain Management Database". In: *AIAA Space 2006 Conference and Exhibition.* AIAA-2006-7232. San Jose, California.

Tolone, W. J., E. W. Johnson, S.-W. Lee, W.-N. Xiang, L. Marsha, C. Yeager, and J. Blackwell (2008). "Enabling System of Systems Analysis of Critical Infrastructure Behaviors". In: *Proceedings of the Third International Workshop on Critical Infrastructure Security.* Rome.

United States (2010). *National Space Policy of the United States of America.* URL: http://www.whitehouse.gov/sites/default/files/national_space_policy_6-28-10.pdf (visited on 08/29/2012).

van Wyk, R. J. (1988). "Management of Technology: New Frameworks". In: *Technovation* 7, pp. 341–351.

Zhang, P. and S. Peeta (2011). "A generalized modeling framework to analyze interdependencies among infrastructure systems". In: *Transportation Research Part B* 45.3, pp. 553–579. DOI: 10.1016/j.trb.2010.10.001.

Zhang, P., S. Peeta, and T. Friesz (2005). "Dynamic Game Theoretic Model of Multi-Layer Infrastructure Networks". In: *Networks and Spatial Economic* 5.2, pp. 147–178. DOI: 10.1007/s11067-005-2627-0.

Chapter 5

Software Implementation of Distributed ISoS Simulations

> "Another type of war game—and one favored by writers of science fiction—is the game played on a high-speed computer. In such a game, the opposing commanders might, for example, sit at their control panels and play out a tactical air war, turning knobs to indicate their allocation of aircraft and weapons to various targets."
>
> Robert D. Specht in *War Games* (1957)

The previous chapter presented the ISoS modeling framework as an approach for formulating infrastructure simulation models with an emphasis on interoperability. Model instantiations represent the structure and behavior of infrastructure elements operating on resource flows, although no specific implementation form is specified. The interoperability interface for interaction between system models included the following requirements:

1. Establish a consistent context of nodes $\mathbf{N}$, locations $\mathbf{L}$, and resource types $\mathbf{T}$,

2. Send/receive the location $\mathcal{L}(e)$ of all elements capable of exchanging resources, and

3. Send/receive resource exchanges $\mathcal{R}_{exchange}(e_{ij}, e_{kl}, r_{ik})$ where elements e_{ij} and e_{kl} are controlled by systems $i \neq k$ and resources r_{ik} are exchanged from system i to k.

This chapter presents an approach to implement interoperable ISoS simulation models as distributed software applications. First, Section 5.1 discusses the motivations and technical challenges in distributed simulation. Next, Section 5.2 compares several potential software architectures for distributed simulation and explains strengths and weaknesses of the High Level Architecture (HLA) selected for this dissertation. Section 5.3 provides an overview of the HLA for those unfamiliar with its history, structure, or current applications. Section 5.4 applies the HLA to the ISoS modeling framework by defining a generic federation object model and operational agreement among federates. Section 5.5 illustrates a sample federate implementation using for generic ISoS application using the Java programming language. Finally, Section 5.6 provides practical insights for developing distributed simulations.

5.1 Fundamentals of Distributed Simulation

Most software applications run a sequence of instructions on a central processing unit (CPU) as a single logical process (LP). There is an implicit guarantee that an LP executes program instructions in a particular order. Even modern computers with multi-core processors rely on operating systems to run these programs as a single LP. Many modeling activities follow the single-LP approach whereby all components are integrated in a single, centralized application. The implicit guarantee of instruction ordering allows even large simulation models to use a shared memory space to reliably communicate information between model components.

Integration in a single model or tool, however, may not be feasible in systems problems where information and models are distributed among a team of decision-makers. For example, the application in Bush et al. (2005) uses a custom model-linker to compile member models into an integrated model. Not only are all participants required to use the same software tool (Vensim) and modeling formalism (system dynamics), but significant coordination is required between every pair of models to specify endogenous/exogenous variables, their representation (units, etc.), and other assumptions (e.g. naming conventions, bounds on validity) to ensure interoperability. Jacobs (2005, p. 6) describes this approach as the *1-1-1* paradigm where "only one actor can use a simulation model, designed by one simulation model designer, to carry out one experiment." Most integrated simulation environments "only support one model formalism, one reporting format and one framework for animation" and it "supports one operating system, one hardware platform, one processor, and thus one concurrent physical location." A new N_n-N_m-N_o modeling paradigm should "support a synthetic, interdisciplinary approach to problem solving" with multiple decision-makers, multiple client side operating systems and hardware platforms, and multiple composite models (Jacobs 2005, p. 17).

Parallel and distributed simulations (PADS) use multiple LPs to evaluate parts of a model application simultaneously. Parallel simulations often use LPs within a machine (e.g. running as separate threads on a multi-core CPU) or within a computing cluster optimized for low communication latencies. Distributed simulation usually deals with heterogeneous machines over larger spatial distances using networking technologies for communication. Fujimoto (2000, pp. 4–5) identifies four main motivations for parallel and distributed simulation:

1. Reduced execution time through parallelization of computation,
2. Fault tolerance through redundancy,
3. Geographical distribution of simulation models and participants, and
4. Interoperability of simulators executing on different machines.

Whereas parallel computing generally emphasizes items 1 and 2, distributed simulation emphasizes items 3 and 4. Applied to the ISoS modeling problem, distributed simulation allows decentralized development and control over the component models and a method of integration based on interoperability.

5.1.1 History and Current Applications

Fujimoto (2000, pp. 8–11) describes historical development of PADS from the contributions of three communities. First, the high-performance computing (HPC) community developed parallel simu-

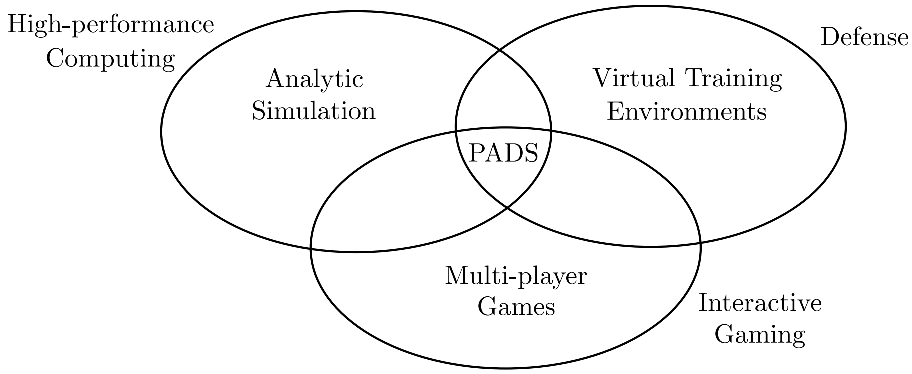

Figure 5-1: Parallel and distributed simulation (PADS) methods and theory developed during the 1970s and 1980s built on contributions from the high-performance computing, defense, and gaming communities.

lation technologies to support analytic studies in the late 1970s and throughout the 1980s. Second, the defense community used distributed simulation to create virtual environments to reduce training expenses compared to field exercises starting in the 1980s. Third, the interactive gaming and Internet communities developed distributed software for multi-player games starting in the early 1980s including the popular MultiUser Dungeon (MUD) game. The three communities, illustrated in Figure 5-1, emphasized different dimensions of PADS. The high-performance computing community used parallelism as a method to accelerate execution of large simulations. The defense community used distributed systems to integrate hardware, software, and humans in a realistic virtual space. Finally, the gaming community used distributed systems to enable player-player interaction in fictional virtual space.

Development and innovation of PADS has widened its contemporary applications (Fujimoto 2000, pp. 12–16). The military uses distributed simulation for wargaming simulations to study attack and defense strategies, training environments for pilots and tank operators, and testing and evaluation (T&E) to evaluate the effectiveness of new devices. Entertainment applications include a wide range of multi-player and massively-multiplayer online games (MMOGs). Business collaborations use distributed simulations as a virtual environment to host social interactions and they have also been applied to educational and training programs for non-military purposes. Finally, parallel simulations are used to simulate large networks with many entities such as the Internet, air, and road transportation systems and to accelerate verification of logic circuits used in digital electronic devices.

5.1.2 Technical Challenges

The fundamental challenge to distributed simulation is the synchronization problem of how to maintain consistent shared state across multiple processors over time. Here, the concept of time is distinguished between that within the model (simulation time) and that in the real world (wallclock time). Distributed simulation handles time in one of several modes which determines the relationship between simulation and wallclock time (Fujimoto 2000, pp. 27–30). Real-time simulation runs simulation time in parallel to wallclock time and is most useful for operational analyses such as studying the actions of human operators or interfacing with hardware. Scaled real-time simulation runs simulation time at a faster or slower (but constant) rate as compared to wallclock time. Finally, as-fast-as-possible simulation runs simulation time at the maximum possible rate which need not be constant and is most useful for analytic studies.

There are a two main problems regarding synchronization. First, network communications between LPs may introduce delays between the times when messages are sent and received. Especially in real-time simulations where simulation and wallclock time are equivalent, the delays may cause events to arrive late, potentially violating causality. For example, a tank destroyed by a missile strike may incorrectly take actions until the destruction event is properly received. To further complicate the issue, network communication delays are not predictable, leading to non-deterministic and non-repeatable simulations if no control is exerted. Second, differences in machine clock, processing rate, or model computational complexity may cause scaled real-time and as-fast-as-possible simulations to drift out-of-sync. For example, a machine requiring more wallclock time to process a simulation time step than another machine will fall behind in a simulation execution without synchronization. The main tradeoff in most approaches is between consistency of synchronized data across components and responsiveness between action and effect (Smed et al. 2002). Whereas analytic simulations must maintain consistency, entertainment applications such as in games relax it to improve responsiveness at the cost of allowing minor causality violations.

Algorithms to address the synchronization problem ensure a distributed simulation produces exactly the same results as a sequential execution running on a single processor (Fujimoto 2000, p. 52). The first class of conservative algorithms obey the local causality constraint whereby "each LP processes events in nondecreasing time stamp order" where the time stamp is the simulation time assigned to each event (Fujimoto 2000, p. 52). The main challenge of conservative algorithms is to determine when it is safe to advance time, i.e. when no messages will be received such that they would violate the nondecreasing order required by the local causality constraint.

Conservative algorithms must first prevent a condition called deadlock in which each member of a distributed simulation is waiting on another, causing the simulation to halt execution (Fujimoto 2000, pp. 54–55). The first solution to the deadlock condition was the Chandy/Misra/Bryant algorithm establishing the concept of null messages to more frequently update time information (Bryant 1977; Chandy and Misra 1978). This algorithm establishes the lookahead time as the minimum duration during which no new events can be scheduled. Under some conditions, however, the overhead of additional communication hurts performance. While newer, more efficient algorithms have been created, conservative algorithms will never be able to fully exploit concurrency due to their inherent overhead of only proceeding when absolutely safe (Fujimoto 2000, pp. 92–94).

Given the inherent efficiency limitations of conservative algorithms, a second class of optimistic algorithms have emerged. Rather than obeying the local causality constraint outright, optimistic algorithms process events assuming no causality errors (Fujimoto 2000, pp. 97–98). The Time Warp algorithm, for example, has the ability to undo and reprocess events in a revised order when a late message arrives (Jefferson 1985). There are some interesting implications of the Time Warp algorithm. For example, a process must reclaim memory from saved states when safe to do so using a process called "fossil collection." In addition, there must be sophisticated handling to allow for transient error conditions while waiting for late events. While optimistic algorithms can be more efficient than conservative ones and are generally simpler to implement an application, they are more complex to implement the simulation executive, consume larger amounts of memory to save state, and may suffer performance problems while rolling back state (Fujimoto 2000, p. 172).

Either conservative or optimistic algorithms are applicable to the ISoS simulation case. As an analytic simulation of infrastructure behaviors, it has a strong consistency and limited responsive-

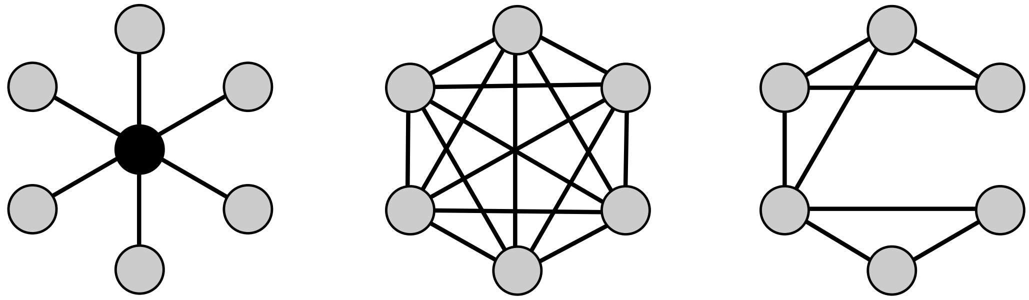

Figure 5-2: Distributed simulation architecture topologies include 1) centralized client-server, 2) distributed peer-to-peer with complete connectivity, and 3) distributed peer-to-peer with partial connectivity.

ness requirement. Rather than evaluating one or several algorithms, this research identifies existing implementations usable for synchronizing distributed infrastructure system models. The next section discusses several alternative software architectures and implementations for this application.

5.2 Software Architectures for Distributed Simulation

Several distributed simulation approaches exist for different types of applications. This section reviews two main topologies for distributed simulation illustrated in Figure 5-2: a centralized client-server (C-S) architecture and a decentralized peer-to-peer (P-P) architecture with partial or complete connectivity. The objective of this section is to identify an existing approach applicable to interoperable simulation gaming and outline its architectural implications in developing model implementations.

5.2.1 Centralized Client-server Architecture

A centralized client-server (C-S) architecture follows a star topology where multiple clients connect to a single server. Each client application has responsibility to "perform local simulation computations pertaining to entities 'owned' by the client" while the central server "maintains global state of the simulation ... and is responsible for notifying each client simulator whenever some portion of the virtual world relevant to the client has changed" (Fujimoto 2000, p. 197). The C-S architecture avoids a portion of the technical synchronization challenges with centralized model state storage (i.e. there is only one state and it is, by definition, correct) while still allowing distributed control of state changes through client-server message transactions.

C-S architectures are prevalent in modern multi-player entertainment games, realized either with dedicated commercial online servers or with one client "host" acting as a server for other homogeneous clients (Smed et al. 2002). Many games design custom C-S implementations using low-level networking libraries in general-purpose programming languages. Some commercial packages (e.g. Valve Developer Community 2012; RakNet 2012; Photon 2013) are also available to provide a higher level of abstraction and implement useful features such as "host migration" which selects and migrates to a new client host if the current one disconnects.

C-S architectures are common in entertainment games for three reasons. First, it provides greater simplicity of implementation as compared to P-P architectures. The separation of logic between server (model state) and client (state change requests) eases testing and debugging, allowing both components to be developed independently while maintaining a common interface. The server ultimately operates as a sequential, single-LP simulation and internal synchronization is limited to managing the concurrency of receiving and responding to client requests. Also, as the role of the client is limited to send state update requests to the server, automated testing clients can also be created to rapidly and reliably execute test programs.

Second, C-S architectures exert central control over the model state which allows for relaxed consistency or synchronization for a more responsive application. In entertainment games, for example, while player inputs usually remain active, delayed messages may cause "lag" in the client display compared to the real server state. Central control over the state also provides a level of security for applications which may be subject to exploits. For example, an unauthorized modification in a competitive game client holding the state of a player could override the true value of a state parameter such as a player's health or power to gain an advantage. The client-server architecture allows all messages requesting state changes to be processed by the trusted server application.

Finally, C-S architectures alleviate some bandwidth concerns by routing all interactions between clients through the central server node rather than passing all messages to all clients in a fully connected network. Ideally, the server would be equipped with high-bandwidth network infrastructure and located at a proximate spatial location for minimal latency to clients. In applications with client hosts (rather than dedicated servers), the host can be dynamically selected among clients on the basis of maximum bandwidth and minimum latency.

The central control over the model state in a C-S architecture also introduces some limitations. First, the server contains the extent of all information in an integrated model. No portions can be contributed by third parties without explicit integration, and no model state is considered private or inaccessible to other portions of the server application. Second, the central control of the model state creates a bottleneck for bandwidth and latency if there are a large number of clients. Some approaches (e.g. Cronin et al. 2001) create server mirrors to partially alleviate this concern.

In addition to entertainment game applications, C-S architectures are also common in web applications and services. A simple web server maintains the state of its content (e.g. HTML documents, images, CSS style sheets, etc.) which is be sent on request via HTTP to clients. More advanced web servers may use server-side programming languages (PHP, ASP.NET C#, Python, etc.) to execute a program and send its output to clients or use a centralized databases (Postgres, MySQL, CouchDB, etc.) to store content and state information.

Similar to the bottleneck in the entertainment game application, there is also a limit to the scale of C-S architectures in web services. As popular web services reach levels of serving hundreds of millions of clients, active research and development over the past ten years has focused primarily on scalability. Most approaches emphasize server replication, efficient response-handling, and/or data caching. Techniques such as the MapReduce programming model, implemented in the Hadoop software framework, mirror the state of databases across geographic regions to manage the scale of client requests to central servers (Dean and Ghamawat 2008; Apache Software Foundation 2013). Server frameworks such as Twisted and Node.js seek to better implement concurrency to handle client requests more efficiently (Twisted Matrix Labs 2013; Joyent, Inc. 2013). REST (representational state transfer) is another architectural style for modern web services (Fielding and Taylor 2002). It emphasizes a stateless client-server architecture, i.e. there is no mutable state on the server-side. This requires all state to be stored client-side, and allows server requests to be cached as, without state, the same inputs will always produce the same outputs.

5.2.2 Distributed Peer-to-peer Architecture

A decentralized peer-to-peer (P-P) architecture follows a partially or fully connected topology where each component connects to one or more other components. Components are responsible for updating local state and no single component has visibility of the entire global state.

Some entertainment games, namely older first-person shooter (FPS, e.g. Gautier and Diot 1997) and real-time strategy (RTS, e.g. Bettner and Terrano 2001) use parallel simulation as a form of a distributed server-less architecture whereby every client simulates the entire game world. These applications use a fully-connected topology where each peer sends its actions (state changes) within a time step to others. Extra considerations must be made to manage performance differences between peers such that the entire game runs in lock-step synchronization. Parallel simulation is usually only scalable to small numbers of players (fewer than ten) due to quadratic bandwidth scaling for transactions between every pair of peers.

Extensions of the basic lock-stepped P-P architecture allow wider distribution of game state across multiple servers. For example, Bharambe et al. (2005) describe an architecture which uses a single copy consistency model to improve responsiveness at the cost of weaker consistency and a pre-fetch process to use locality and predictability to improve the performance of logic computations. While the architecture of Bharambe et al. (2005) is technically a P-P, similar to parallel simulation it assumes the possible states of each peer are still controlled by a central authority (i.e. the game studio). Less centralized control of the application makes it vulnerable to cheating. Bharambe et al. address these security concerns by "carefully selecting the owners of primary objects ... to limit the damage malicious players or nodes can inflict on others." P-P architectures are more common in other domains where there may not be a central developer or simulation application to control a server and there is a purely collaborative agreement between peer applications (but not necessarily actions within applications).

Distributed Interactive Simulation (DIS) is an architecture and interface specification (protocol) originally developed for real-time military simulations (IEEE 1998). It specifies a networking protocol to communicate state information across distributed simulation models without centralized control. Each member application is responsible for maintaining synchronization of its state. DIS uses a periodic broadcast communication method represented as a fully-connected network topology, and as such, has bandwidth limitations for large numbers of members. To limit data exchanged, "dead-reckoning" is used to interpolate state, for example position changes of an entity.

The High Level Architecture (HLA) is an architecture for military simulations viewed as a successor to DIS. Whereas DIS specifies the format of line-level messages between simulation applications, the HLA only specifies an application programming interface (API) to another software component, the runtime infrastructure (RTI). An RTI implementation includes the synchronization algorithms required to run as-fast-as-possible and scaled time simulations in addition to real-time simulations. The RTI also acts as a centralized router for message-passing among federates to reduce bandwidth use. Thus, while the HLA logically follows a distributed architecture with partial connectivity, it sometimes follows a centralized architecture with the central RTI component (CRC) coordinating messages between local RTI components (LRCs).

While DIS and HLA are common in military and defense domains, distributed simulation ap-

Table 5.1: Comparison of distributed simulation architectures and approaches

Architecture	Topology	Interface	Decentralization	Synchronization	Usability	Standardization	Accessibility
Monolithic	n/a	API	−	+	+	−	+
Custom Game	C-S	Commercial network library, other TCP/IP	0	0	0	−	+
Web-based	C-S	HTTP, JSON, XML	0	0	+	0	+
Parallel Simulation	P-P	Custom TCP/IP	0	+	0	−	+
DIS	P-P	Standard TCP/IP	+	−	0	+	+
HLA	P-P	Standard API	+	+	−	+	0
Homespun	P-P	Proprietary COM, CORBA, WinSock, other TCP/IP	+	−	0	−	+

plications in industry tend to use other "homespun" protocols such as WinSock, CORBA, and COM to facilitate communication between peers (Boer 2005, pp. 43–44). While applications of true distributed simulation are limited, there is often interoperability between modeling tools and other systems such as databases, spreadsheets, optimization tools, statistical software, and enterprise resource planning (ERP) systems (Boer 2005, p. 43).

5.2.3 Architectural Comparison

Table 5.1 compares the above approaches in consideration for an architecture for interoperable simulation gaming. Each approach is qualitatively evaluated on the basis of five categories.

Decentralization evaluates the ability to distribute control over model structure and behavior. The monolithic architecture provides no decentralized control over the model state or changes as all models must be integrated into a single application (–). Custom and web-based C-S architectures and the parallel simulation P-P approach only provide decentralized control over state changes (0). DIS, HLA, and homespun P-P architectures allow full decentralized control over both model state and state changes (+).

Synchronization evaluates the ability to maintain consistency across models. As a real-time simulation architecture, DIS does not include synchronization (–). Homespun P-P architectures must also implement algorithms to maintain synchronization of the distributed state (–). Custom and web-based C-S architectures require synchronization of the server state (0). Monolithic architectures do no need synchronization (+), parallel simulation inherently synchronizes state across clients (+), and the HLA RTI implements synchronization algorithms (+).

The simulation architecture should provide a high usability for development and testing. The HLA is a complex standard with dozens of API calls which causes a steep learning curve (–). DIS and homespun architectures require additional implementation of algorithms or other approaches to ensure synchronization (0). Parallel simulation simplifies the synchronization issue with duplication of a single simulation model, however message transactions must still be configured (0). Web-based architectures are typically simpler and easier to use than those in general programming languages

(+) and monolithic architectures are the most traditional and easiest to implement (+).

Standardized approaches are preferred over proprietary or custom ones to ensure wider acceptance and use. Monolithic, custom C-S, parallel simulation, and homespun architectures all require non-standardized interfaces through APIs or TCP/IP protocols (–). Due to its wide use, web-based C-S architectures are more standardized, but still require an API (0). DIS and HLA include specified protocols and APIs as published IEEE standards (+).

The simulation architecture should also be accessible for widespread use by different organizations. All architectures except the HLA do not require specific licensed software (+). The HLA requires an RTI implementation which may be open source (e.g. partial implementations include Open HLA 2013; The Portico Project 2013) or commercially-licensed (0).

5.2.4 Summary

The HLA is the only existing architecture meeting the requirements of distributed control over model state and providing strong consistency and non-real-time simulation with a standard interface. It is limited, however, by difficult implementation (due to its complexity) and limited accessibility due to a licensing expense for a commercial RTI implementation. Despite these limitations, the HLA is selected as the architectural basis of the ISoS framework software implementation.

The ideal architecture, however, is not achieved by any existing approach, suggesting there is an opportunity for future research and development. In particular, combining the simplicity and ease of access for modern web-based services and extended to a P-P architecture may provide a significantly improved approach. Such an architecture could use an API similar to that of the HLA, but rather than operating in a software application on an OS, it would operate in a browser. While styles like REST may not be fully applicable due to the inherent state-storing nature of simulations, its architectural principles of simplicity could be helpful in reducing the complexity of developing distributed simulation applications. Indeed, the inclusion of web services in the HLA-Evolved standard takes a few steps in this direction (IEEE 2010c). The web services define a protocol using simple object access protocol (SOAP) for communication between a federate and the RTI. A future extension with a browser-based JavaScript RTI implementation may allow federates to run completely in the browser.

5.3 Overview of the High Level Architecture (HLA)

As introduced in the previous section, the High Level Architecture (HLA) is the culmination of distributed simulation research and development driven primarily from military cases, but applicable to generalized simulation. It is logically a distributed peer-to-peer architecture using an application programming interface (API) to communicate between member models using common RTI. This section is intended for readers unfamiliar with the HLA to discuss its history and applications, structure and services, and recommended development processes.

5.3.1 History and Current Applications

The HLA has its roots in the military community. Early efforts to develop distributed simulations for virtual environments were sponsored by the Defense Advanced Research Projects Agency

(DARPA). SIMNET (SIMulator NETworking) was a prototype wide area network vehicle simulator project from 1983–1989 (Fujimoto 2000, p. 9). A follow-on program to SIMNET developed Distributed Interactive Simulation (DIS), “an infrastructure for linking simulations of various types at multiple locations to create realistic, complex, virtual ‘worlds’ for the simulation of highly interactive activities” (DIS Steering Committee 1994). DIS specifies a transport layer protocol for conducting simulations across multiple computers with improved interoperability, later standardized as IEEE Std. 1278 (1998). Applied mostly in training exercises, the DIS does not include coordination to advance simulation time and uses dead reckoning algorithms to extrapolate state information between updates.

The Aggregate Level Simulation Protocol (ALSP) developed by MITRE in the early 1990s applied interoperability approaches formulated under DIS to aggregate-level combat simulations. Rather than focusing on operational-level training with individual vehicles and troops as under DIS, the objective of ALSP was to enable joint military wargames across the Army, Air Force, and Navy using groups of abstracted force objects (Wilson and Weatherly 1994). The ALSP introduced an infrastructure software (AIS) as a separate module to manage the distributed simulation and added several key improvements including time and data management.

The contributions of ALSP and DIS were unified the late-1990s under the High Level Architecture (HLA), a standard software architecture for federated simulation (Dahmann et al. 1997). Federated in this context means a distributed set of heterogeneous simulation models, thus emphasizing the interoperability focus of the standard. The most recent iteration of the standard is IEEE Std. 1516-2010 (HLA-Evolved), which “evolved” HLA from the previous release to include new features including additional XML support, an API for web services (WSDL), modular FOMs, encoding helpers, and standardized time representations (IEEE 2010c). Similar to ALSP’s AIS, the HLA specifies a runtime infrastructure (RTI) to support data and time management processes.

Since its initial publication, the HLA has been applied growing number of non-military (civilian) domains seeking to implement distributed simulation (Straßburger 2001). Some prominent examples include space exploration (e.g. Reid 2000), aviation and air traffic management (e.g. Sweet et al. 2002; Simons et al. 2013), grid computing (e.g. Zajac et al. 2004), electrical and communications systems (e.g. Hopkinson et al. 2006; Sztipanovits et al. 2012), traffic systems (e.g. Schulze et al. 1999), infrastructure interdependencies (e.g. Eusgeld et al. 2011; Nan and Eusgeld 2011), manufacturing and supply chains (e.g. Hibino et al. 2002; Bruzzone et al. 2005), health care (e.g. Brailsford et al. 2006), and emergency response (e.g. Jain and McLean 2003; Fiedrich 2006; Liu et al. 2007).

While the HLA has been previously identified as a potential architecture to simulate infrastructure systems, its application in practice thus far have been limited. One particular challenge deals with differing time-scales and model resolutions of component models (Pederson et al. 2006). For example:

> Many models and computer simulations exist for aspects of individual infrastructure (e.g., load flow and stability programs for electric power networks, connectivity and hydraulic analyses for pipeline systems, traffic management models for transportation networks), but simulation frameworks that allow the coupling of multiple interdependent infrastructures to address infrastructure protection, mitigation, response, and recovery issues are only beginning to emerge. This problem is exacerbated by the variety of classes of models in use: physics-based models, nodal analysis models, agent-based models, stocks-and-flows models, and more. ... simply ‘hooking’ several existing infrastructure models together generally does not work: every model has its own unique assumptions, data, and numerical requirements (such as time-step sizes, scaling limita-

tions, or computational algorithms) that may not be compatible with other models. (Rinaldi et al. 2001)

Thus, it appears a common modeling framework which aligns the temporal and spatial resolution of component models (such as that presented in Chapter 4) is necessary to enable application of the HLA.

The limited use of the HLA is not unique to the infrastructure domain. Though commonplace in defense and growing in frequency as applied in research, distributed simulation is still in its infancy in industrial applications. Boer et al. (2009) conducted a questionnaire of commercial off-the-shelf (COTS) simulation vendors, industrial simulation practitioners, HLA designers and developers, HLA simulation practitioners in defense, and commercial HLA vendors to study this issue. They found the main reasons for limited industrial applications to include:

- An emphasis on inexpensive, limited, and disposable industrial models rather than credible, reusable, and expensive models in defense,
- Industry use of COTS packages compared to general-purpose programming languages in defense, and
- High RTI cost, high complexity, and few experts in industry.

The complexity of the HLA standard has led others to develop simpler service-oriented architectures for infrastructure modeling with centralized event processing and significantly reduced functionality (Tolone et al. 2008).

5.3.2 Structure and Services

The HLA includes three main components. First, ten rules in IEEE Std. 1516 (2010) set requirements for the federation and each federate. Second, an object model template (OMT) in IEEE Std. 1516.2 (2010) defines the structure of information produced or exchanged in a simulation. Finally, a runtime infrastructure (RTI) in IEEE Std. 1516.1 (2010) specifies how "simulations to connect to one another, exchange data, and coordinate activities during a distributed runtime execution" (IEEE 2010c).

The ten rules include five for the federation as a whole and five for each federate (IEEE 2010c, pp. 20–22). In summary, all federation simulation objects must be documented in a federation object model (FOM), exist in the federate application state, and exchange data via the RTI using standard services. Federates must be able to manage local time and their simulation objects must be documented in a simulation object model (SOM) which determines the attributes and/or interactions to be sent, received, or transferred to another owner.

The OMT specifies the data structures of objects and is applied to create the SOM for each federate and FOM for each federation.[1] The FOM is used as an interface control document to specify a common language between federates during the simulation and is foremost used to encode and decode data. While the HLA does not specify a network-level protocol, the data is encoded to a serialized (byte array) form when sent to the RTI, and must be decoded when received from the RTI. Explicit data formats are required as heterogeneous machines and programming languages may implement different standards by default.

[1] The FOM is composed of each federate's SOM to allow for heterogeneity.

The FOM is often presented in a tabular format for visibility, but is formatted as an XML document for a federation. The major elements of a FOM specify the hierarchy of object classes (persistent objects) and their attributes, interaction classes (transient objects) and their parameters, and the corresponding data types used for attributes and parameters. All complex data types must be specified in the FOM as compositions of basic data types (e.g. boolean, short, integer, float, Unicode string, etc.) using standard or custom structures.

The RTI provides services pertaining to the following seven categories:

1. Federation management: creating, destroying, synchronizing, storing, and restoring federations and joining and resigning of federates,
2. Declaration management: specifying data sent or received,
3. Object management: registering, discovering, deleting, and removing objects, updating and reflecting attribute values, and sending and receiving interactions,
4. Ownership management: specifying authority for updating attribute values
5. Time management: enabling and disabling time-related modes and advancing time,
6. Data distribution management: filtering data sent or received, and
7. Support services: other miscellaneous functions.

Some RTI implementations (e.g. Pitch pRTI, MÄK RTI, and Open HLA) include a central RTI component (CRC) to manage federations and local RTI components (LRCs) to pass communication between the federate and the CRC. The LRC includes an application programming interface (API) for integration into the simulation program using a general-purpose programming language.[2] While separate components, the CRC and LRC(s) may actually be running on the same physical machine. Other RTI implementations (e.g. Portico) use a truly distributed architecture with each federate accessing a separate RTI component.

5.3.3 Recommended Development Processes

In addition to the HLA, IEEE Std. 1730 (2010) specifies a recommended practice for a distributed simulation engineering and execution process (DSEEP). The DSEEP is structured as a waterfall-style development process with specific output products from activities used as inputs to successive activities. It breaks down development of a distributed simulation into seven steps:

1. Define simulation environment objectives,
2. Perform conceptual analysis,
3. Design simulation environment,
4. Develop simulation environment,
5. Integrate and test simulation environment,

[2] IEEE Std. 1516-1 provides API code for Java, C++, and Web Services (SOAP HTTP), however the specific language bindings supported depend on the RTI implementation.

6. Execute simulation, and

7. Analyze data and evaluate results.

Each step includes a number of activities, each with specified inputs, outputs, and recommended tasks.

This discussion focuses on step 4 to develop the simulation environment, assuming the simulation objectives, conceptual analysis, and preliminary design of the simulation environment are completed in advance. As a generalization for other distributed simulation architectures including DIS and TENA (Test and Training Enabling Architecture), some DSEEP activity products overlap with HLA requirements. For example, activity 4.1 develops the simulation data exchange model which corresponds to the HLA FOM. Additional products, however, coordinate other development activities. In particular, activity 4.2 establishes simulation environment agreements to identify runtime interactions between models.

The simulation environment agreement (called the federation agreement as applied to HLA) specifies how runtime interaction will take place between federates. It adds the context of the particular scenarios under investigation to the abstract nature of the HLA to specify:

1. Time management agreements,

2. Data management and distribution agreements,

3. Synchronization and initialization procedures,

4. Saving and restoring strategies, and

5. Publishing and subscribing responsibilities, among others.

Completing each of these tasks involves creating requirements using the standard HLA services.

5.4 ISoS Federation Implementation

This section discusses the implementation of the infrastructure system-of-systems model discussed in the previous chapter as a federation using the HLA. A generic federation object model defines the common data structures and a federation agreement specifies the operational behaviors required of each federate.

5.4.1 Federation Object Model

The federation object model (FOM) defines data structures to meet the interoperability interface required in the ISoS modeling framework. Specifically, it:

1. Represents nodes, locations, resource types, resources, and resource sets as data types,

2. Provides a mechanism for sending and receiving elements' location, and

3. Provides a mechanism for sending and receiving resource exchanges between elements.

Table 5.2: ISoS simple data types

Name	Representation	Units	Resolution	Accuracy	Semantics
ISOSnode	HLAunicodeString	NA	NA	NA	Node name.
ISOSresourceType	HLAunicodeString	NA	NA	NA	Resource type name.

Table 5.3: ISoS enumerated data types as an alternative to simple data types

Name	Representation	Enumerator	Values	Semantics
ISOSnode	HLAinteger32BE	Node1	1	The first enumerated node.
		Node2	2	The second enumerated node.
		⋮	⋮	⋮
ISOSresourceType	HLAinteger32BE	Type1	1	The first enumerated resource type.
		Type2	2	The second enumerated resource type.
		⋮	⋮	⋮

The simple data types in Table 5.2 represent nodes and resource types as `ISOSnode` and `ISOSresourceType` data types, each using the default Unicode string encoding `HLAunicodeString`. This data type requires the list of possible node names and resource type names to be defined in the federation agreement. Table 5.3 presents an alternative representation for nodes and resources as enumerated data types using the default HLA 32-bit big-endian integer encoding `HLAinteger32BE`. This option requires the list of possible nodes and resource types to be specified in the FOM itself. While the simple data type option is more flexible to accommodate new nodes and resource types, the enumerated data type is more tightly controlled to only accept certain pre-specified values.

The simple or enumerated data types are used in composite fixed record and array data types. The fixed record data type `ISOSresource` in Table 5.4 represents resources as a combination of a resource type and a quantitative amount with units defined in the federation agreement. The `ResourceType` field uses the existing `ISOSresourceType` data type (either simple or enumerated) while the `Amount` field uses the default HLA 64-bit big-endian floating-point data type `HLAfloat64BE`. The array data types in Table 5.5 represent pairs of nodes as locations and sets of resources. The `ISOSlocation` data type uses the `HLAfixedArray` encoding to combine two `ISOSnode` data types as a fixed array where the first node is the origin and the second node is the destination. Similar to the simple data type option for nodes, a list of possible locations must be specified in the federation agreement. The `ISOSresourceSet` data type uses the `HLAvariableArray` encoding to combine any number of `ISOSresource` data types.

Table 5.6 defines `ISOSelement` as the base object class representing elements for an ISoS framework. It has two required attributes. The `Name` attribute uses the `HLAunicodeString` data type to identify its unique name. The `Location` attribute uses the `ISOSlocation` data type to identify its location. While the name is not expected to change (i.e. it is static), the location attribute will update conditionally during an element transport event. Neither attribute can be divested (D) or acquired (A), and both can be published (P) and subscribed (S). The attributes do not use dimensions and are transported using reliable networking protocols. While updates to the name attribute can be receive-ordered, the location attribute should be time stamp-ordered to preserve synchronization during an execution.

While the object class ISOSelement presented is the minimum required to adhere to the ISoS interoperability interface, future extensions could be proposed to share more data under the ISoS framework. The three remaining element attributes include the resource contents, parent element, and element state. These attributes are included in Table 5.6 as Contents using the ISOSresourceSet data type to identify the resource contents, Parent using the HLAunicodeString data type to identify the parent element name, and State using the HLAunicodeString data type to identify the current operational state name. All three extended attributes are also conditionally updated when the associated behaviors take place.

Table 5.7 defines ISOSresourceExchange as the interaction class to exchange resources between elements. It has three required parameters. The SendingElement parameter uses the HLAunicodeString data type to identify the element sending resources by name. Likewise, the ReceivingElement parameter uses the HLAunicodeString data type to identify the element receiving resources by name. Finally, the ResourcesExchanged parameter uses the ISOSresourceSet to define the set of resources exchanged between the two elements. The ISOSresourceExchange interaction does not use dimensions, should use reliable transport, and is time stamp-ordered for synchronization during a simulation.

As an alternative to using resource exchange interactions, two additional attributes can be added to the ISOSelement object. The ExchangeInputs attribute specifies a list of resources to be sent to target elements and the ExchangeOutputs attribute specifies a list of resources to be received from target elements. To meet resource flow consistency conditions, the inputs from a sending element should equal the outputs from the receiving element. Both attributes use the ISOSdirectedResourcesSet array data type, which in turn uses the ISOSdirectedResources fixed record to store pairs of ISOSresourceSet and HLAunicodeString data elements. Using attributes rather than interactions is more desirable for regular continuous resource exchanges as the attributes can be conditionally updated using standard processes. Alternatively, resource exchange interactions are more desirable for discrete resource flows. Both methods can be combined in a single federation if desired.

Table 5.4: ISoS fixed record data types

Record Name	Field Name	Field Type	Field Semantics	Encoding	Semantics
ISOSresource	ResourceType	ISOSresourceType	Type of resource	HLAfixedRecord	Resource type
	Amount	HLAfloat64BE	Amount of resource		and amount.
ISOSdirectedResources†	Resources	ISOSresourceSet	Resources to be exchanged	HLAfixedRecord	Resources exchanged
	TargetName	HLAunicodeString	Name of target element		and target element.

† Required for attribute-based alternative to using resource exchanging interactions.

Table 5.5: ISoS array data types

Name	Element Type	Cardinality	Encoding	Semantics
ISOSlocation	ISOSnode	2	HLAfixedArray	Pair of nodes.
ISOSresourceSet	ISOSresource	Dynamic	HLAvariableArray	Set of resources.
ISOSdirectedResourcesSet†	ISOSdirectedResources	Dynamic	HLAvariableArray	Set of directed resources.

† Required for attribute-based alternative to using resource exchanging interactions.

Table 5.6: ISoS object attributes

Object	Attribute	Data Types	Update Type	Update Condition	P/S	Transport	Order
ISOSelement	Name	HLAunicodeString	Static	NA	PS	Reliable	Receive
	Location	ISOSlocation	Conditional	On transporting.	PS	Reliable	TimeStamp
	Contents*	ISOSresourceSet	Conditional	On resource storing.	PS	Reliable	TimeStamp
	Parent*	HLAunicodeString	Conditional	On element storing.	PS	Reliable	TimeStamp
	State*	HLAunicodeString	Conditional	On transforming.	PS	Reliable	TimeStamp
	ExchangeInputs†	ISOSdirectedResourcesSet	Conditional	On exchange sent.	PS	Reliable	TimeStamp
	ExchangeOutputs†	ISOSdirectedResourcesSet	Conditional	On exchange received.	PS	Reliable	TimeStamp

* Optional extensions to share data beyond the immediate ISoS interoperability interface.
† Alternative attributes to using resource exchanging interactions.

Table 5.7: ISoS interaction parameters

Interaction	Parameter	Data Types	Dim.	Transport	Order
ISOSresourceExchange	SendingElementName	HLAunicodeString	NA	Reliable	TimeStamp
	ReceivingElementName	HLAunicodeString			
	ResourcesExchanged	ISOSresourceSet			

5.4.2 Federation Agreement and Required Activities

The federation agreement establishes two key items. First, it specifies any additional constraints, invariant conditions, or other assumptions about the FOM and its use in federates. Second, it specifies required processes and behaviors of federates to correctly participate in the federation.

If using the simple data types for nodes and resource types, allowable values must be specified in the federation agreement. Next, the measurement units of each resource type must be identified for proper quantification. For example, resource type "water" may be measured in cubic meters and resource type "electricity" in kilowatt hours. An element naming scheme should be devised such that each element receives a unique name. Finally, there should be an agreement as to how and when the resource exchange interaction can be used. For example, the sending and receiving elements should, at minimum, have matching destination and origin nodes respectively.

In addition to the FOM elements specified in the previous section, the federation agreement also establishes timing information. Data types for time and duration are selected and entered into the time representation section of the FOM. For ISoS simulations, federates must agree on a uniform initial time t_0, ending time t_{end}, and time-step Δt for advancing the simulation. Synchronization point labels including "initialized" and "reset" used in this application are also defined in the FOM.

During a simulation execution, federates must use specific HLA services to interact with the federation. The activity diagram in Figure 5-3 includes start-up, advance, reset, and shut-down activities as components of a federate life-cycle. The federate performs the start-up activity which initializes the federate to the initial time t_o. Next, the advance activity increments time by Δt as long as the simulation should continue. After the simulation achieves the ending time t_{end} or is halted, the federate can either reset to initial conditions or shut-down.

The following sections describe each activity in detail. HLA services are identified by section number in the standard (e.g. *Connect* is 4.2). Services for communication to the RTI are illustrated as rounded rectangles (actions) and callback services for communication from the RTI are illustrated as labels on transitions (conditions).

Start-up Activity

The start-up activity diagrammed in Figure 5-4 connects to a federation and configures the federate. It uses service 4.2 to connect to the RTI. If the federation does not yet exist, one is created using service 4.5 and the federate joins the federation using service 4.9. Next, the federate enables asynchronous messages (i.e. outside of the time-advancing state) using service 8.14 and enables time constraint and regulation (using Δt as the look-ahead) using services 8.5 and 8.2. Callback services 8.6 and 8.3 confirm time constraint and regulation is enabled.

The federate attempts to register the "initialized" synchronization point for all federates participating in a simulation using service 4.11. If the callback service 4.12 is successful the federate

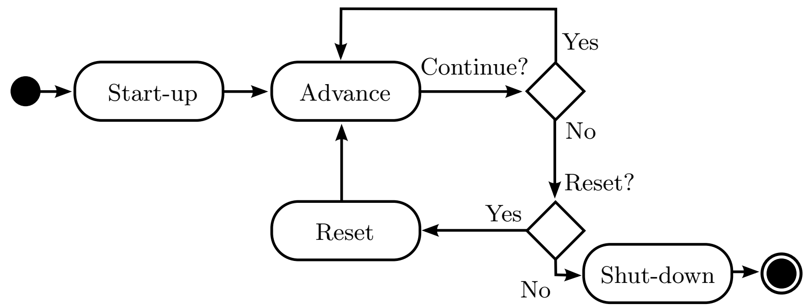

Figure 5-3: A federate life-cycle includes start-up, advance, reset, and shut-down activities.

will initiate a later save. The federate advances time to an initial value of $t_0 - 2\Delta t$ and waits for the callback service 4.13 to announce the "initialized" synchronization point.[3] Next, the federate publishes and subscribes object class attributes using services 5.2 and 5.6 and registers object instances of the local infrastructure systems using service 6.8. The federate uses service 4.14 to notify the synchronization point was achieved, and waits for the callback service 4.15 to notify that other federates are also synchronized.

The federates advance time to $t_0 - \Delta t$ to update and reflect attribute values. The federate which successfully registered the synchronization point requests a federation save using service 4.16. Other federates wait for the callback service 4.17 to initiate the federation save. The federate uses services 4.18 to begin the save and 4.19 to complete the save, waiting for callback service 4.20 to notify other federates have also completed saving. Finally, the federate advances time to t_0.

Advance Activity

The advance activity diagrammed in Figure 5-5 advances time by Δt and allows for timestamped callback services. To accommodate dependencies among federates, however, the time step is divided into $\gamma \geq 1$ pseudo-steps, each of duration $\xi = \Delta t / \gamma$. During each pseudo-step, federates are allowed to exchange interdependent attributes and interactions without advancing internal simulation time. For example, if using a time-step of $\Delta t = 1.0$ year and $\gamma = 5$, each sub-step would advance time by $\xi = 1/\gamma = 0.2$ years of simulated federation time, however, only the last would advance the simulated federate time by $\Delta t = 1.0$ year.[4] To completely capture all dependencies, ξ must exceed the longest dependency path ρ, i.e. $\xi > \rho$. In cases with cyclic dependencies, no value of ξ is sufficient to result in zero error. In practice, however, gains between systems are typically low enough such that values such as $\xi = 2\rho^\star$ may produce negligible error where $\rho^\star$ is the longest acyclic dependency path.

To complete the advance activity, a federate first updates any modified attribute values using service 6.10 and sends interactions using service 6.12. Next, it requests a time advance to $t+\xi$ using service 8.8 and waits on the grant callback service 8.13. Callback services for reflecting attribute

[3]Federates first initialize to time $t_0 - 2\Delta t$ to allow for sufficient advance periods to receive state information at time t_0 from other federates. The advance to $t_0 - \Delta t$ discovers objects and reflects attributes before the federation is saved. Additional updates are processed after the federation restoration and during the advance to t_0.

[4]The method of federation pseudo-steps which adds more frequent federation time steps is in contrast to the typical approach of using more frequent internal federate updates (e.g. for state variable integration) to improve model fidelity while limiting federation update rates.

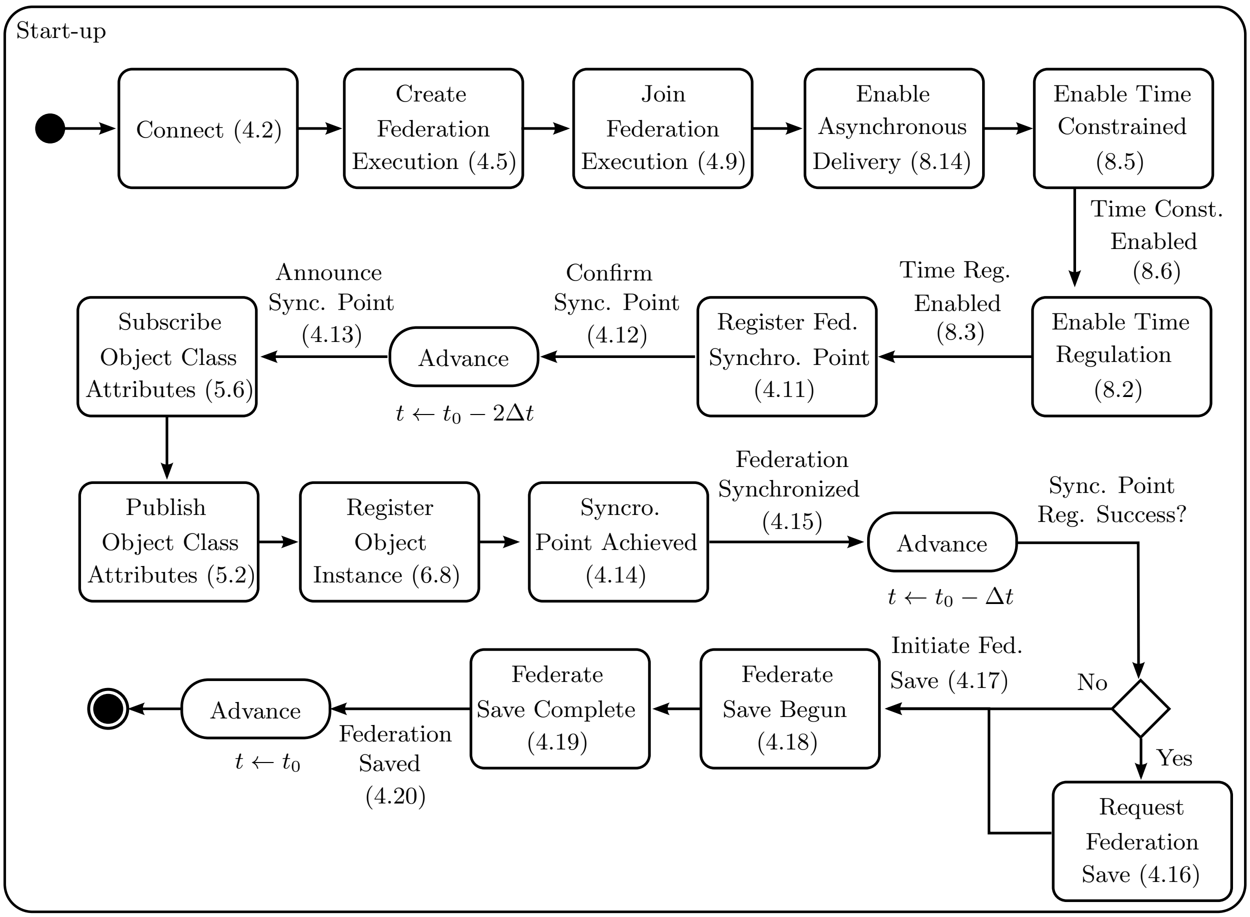

Figure 5-4: The start-up activity connects to a federation and configures a time-constrained and time-regulating federate. It uses a synchronization point to ensure all federates are connected before registering objects and sending attribute updates. Finally, a federation save is requested by the first federate to store the initial state before ticking to the initial time and executing the simulation.

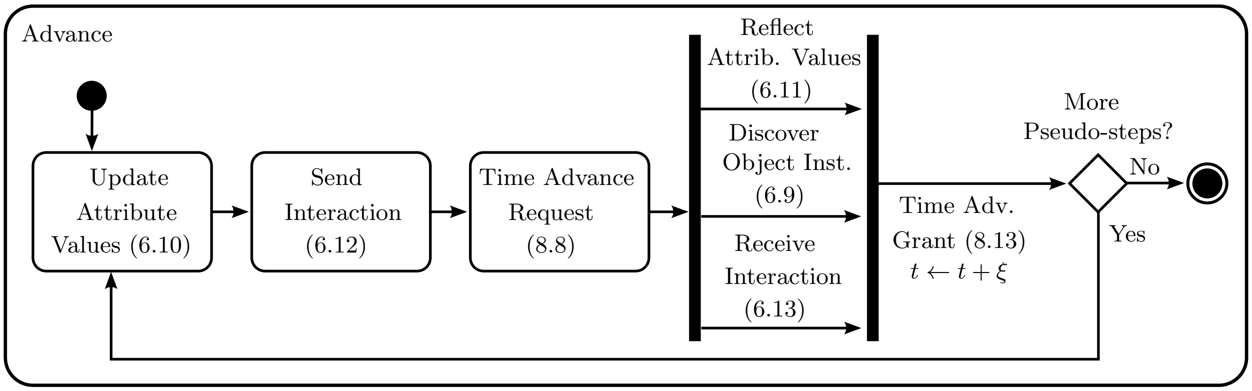

Figure 5-5: The advance activity sends attribute updates and interactions, advances time by one step, and responds to callbacks. These steps are repeated, if necessary, to accommodate interdependent effects.

updates (service 6.11), receiving interactions (service 6.13), and object instance discovery (service 6.9) can be received while waiting for service 8.13. As mentioned, the process repeats γ times to achieve the entire time step of Δt.

Reset Activity

The reset activity diagrammed in Figure 5-6 restores a federation to the initial conditions at time t_0. First, the federate registers the "reset" synchronization point for all federates participating in a simulation using service 4.11. If the callback service 4.12 is successful the federate will initiate the restore. The federate waits for the callback service 4.13 to announce the "reset" synchronization point, uses service 4.14 to notify the synchronization point was achieved, and waits for the callback service 4.15 to notify that other federates are also synchronized. The federate which successfully registered the synchronization point requests the federation restore using service 4.24 which is confirmed by callback service 4.28. The federate waits on callback services to notify the restore has begun 4.26 and should be initiated 4.27. Once restored, the federate is at time $t = t_0 - \Delta t$, uses service 4.28 to notify it is complete, and waits on the callback service 4.29 that the rest of the federates are also restored. Finally, federate advances to time t_0.

Shut-down Activity

The shut-down activity in Figure 5-7 configures and disconnects a federate. It disables time constraint and regulation with services 8.7 and 8.4 and resigns the federate using service 4.10. In case it is the last federate to resign, it attempts to destroy the federation execution using service 4.6 and disconnects from the RTI with service 4.3.

5.4.3 Network and Physical Infrastructure

The simulation environment uses a computer network for communication. For a small number of federates, a local area network (LAN) with wired connections to a common network switch or hub is optimal, however a wide area network connection (WAN) will also work provided compatible RTI communication protocol, adequate bandwidth, and low latency. Figure 5-8 diagrams a notional

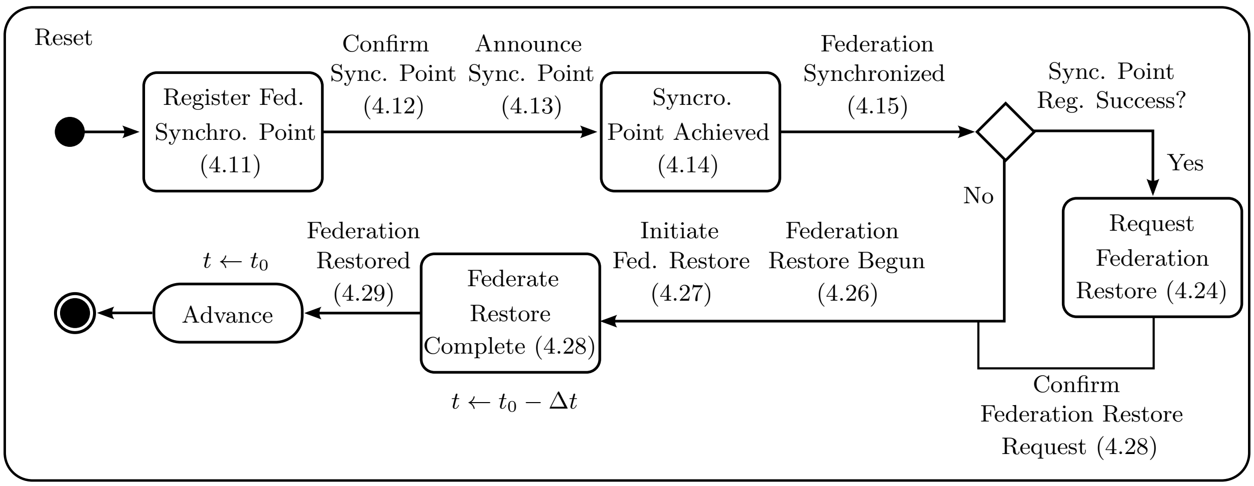

Figure 5-6: The reset activity restores a federation to the initial state. It uses a synchronization point to ensure that all federates request a reset before restoring the initial state saved during the start-up process. Finally, it advances to the initial time.

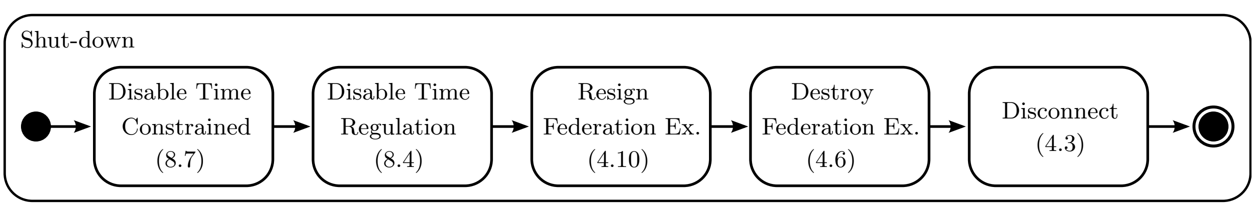

Figure 5-7: The shut-down activity disconnects a federate from the federation.

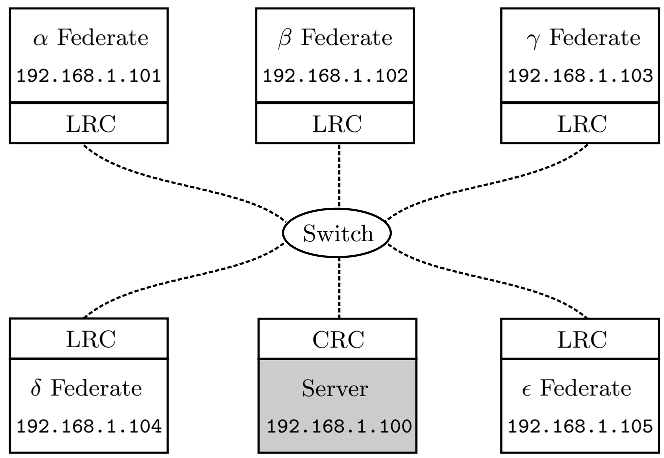

Figure 5-8: The network diagram for five federates $(\alpha, \beta, \gamma, \delta, \epsilon)$ each at separate (notional) IP addresses requires a switch with a connection to the CRC (if required) at a specified IP address.

LAN consisting of six computers connected with a switch, each assigned a private IP address. If required by the RTI implementation, one computer is designated to host the central runtime component (CRC) and all connecting federates must configure their local runtime components (LRC) to connect to the IP address of the CRC (`192.168.1.100` in this case). While the CRC need not be a stand-alone server, it may be desirable to do so for configuration and performance reasons. Following the federated architecture, each federate (α through ϵ) runs a simulation model which communicates within the federation using its LRC. Some federations may assign federates to be passive (e.g. to only display information on a large-format screen) or non-interactive (e.g. to manage time advancement or provide common federation data services).

All federates must also use a common RTI implementation. In addition to the network, software, and hardware infrastructure, the simulation environment also requires sufficient power for each computer and any displays including projectors. Especially when using desktop computers, large displays, or projectors, it is critical to verify there is sufficient load capacity on the electrical network to power all computers and displays.

5.5 ISoS Federate Implementation

A federate implementation of the ISoS framework is a software application capable of participating in a federation. This section introduces a generic federate implemented in the Java programming language designed to be extensible to future ISoS application cases. As a particular implementation, the proposed structure is by no means required or optimal. Alternative implementations may use different programming languages, data structures, or overall organization to interoperate with any federate implementation adhering to the federation object model and federation agreement.

Figure 5-9 illustrates a simplified object class diagram of the core modules of the generic federate implementation. Color schemes illustrate three modules including core ISoS model interfaces (blue boxes), default model implementations (green boxes), and HLA model implementations with related services (orange boxes). Default model implementations represent locally-controlled simulation

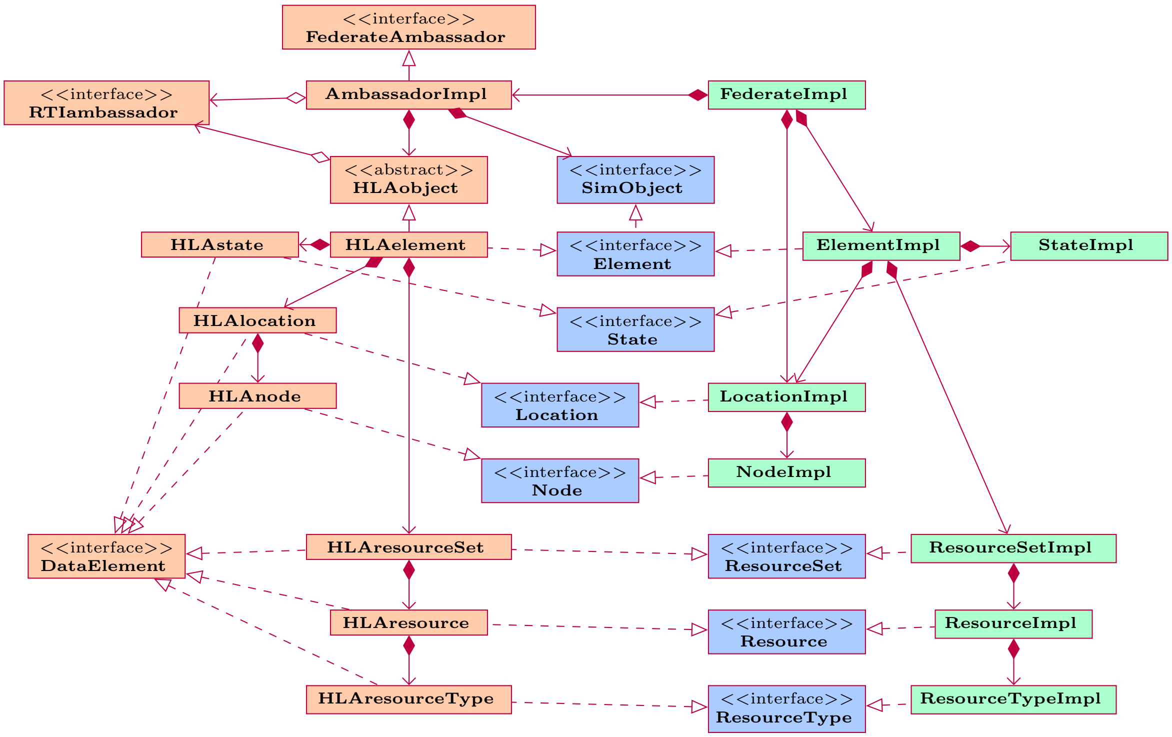

Figure 5-9: Simplified object class diagram for a generic federate implementation. Blue boxes define core ISoS interfaces, green boxes implement default models, and orange boxes implement HLA models. Resource exchanging interactions and attributes are omitted for clarity.

objects while HLA model implementations represent remotely-controlled simulation objects.

5.5.1 Core ISoS Interface

The core ISoS interfaces in Figure 5-10 define a common structure for local model implementations and HLA model implementations. The classes `SimObject` and `SimInteraction` provide basic interfaces to any objects requiring time-stepped simulation and time-based message-passing. `SimObject` defines three methods to initialize simulation objects, compute (`tick`), and commit (`tock`) state changes. The two-part state change computes all state changes using consistent information to eliminate update order dependencies for local objects.

The `Element` interface extends the `SimObject` interface to include methods to access a name (for reference) and dependent properties from the ISoS modeling framework including spatial location, resource contents, operational state, and parent element. The `State` interface defines a method to access the operational state name. The `Location` interface defines methods to access the origin and destination nodes and the `Node` interface defines a method to access a text-based name. The `ResourceSet` interface defines a method to access component resources as a collection and particular resources by type, the `Resource` interface defines methods to access its associated type and floating point amount, and the `ResourceType` interface defines a method to access a text-based name.

Both attribute-based and interaction-based resource exchanging behaviors are specified. The `Element` interface includes methods to access inputs and outputs from an exchanging behavior with

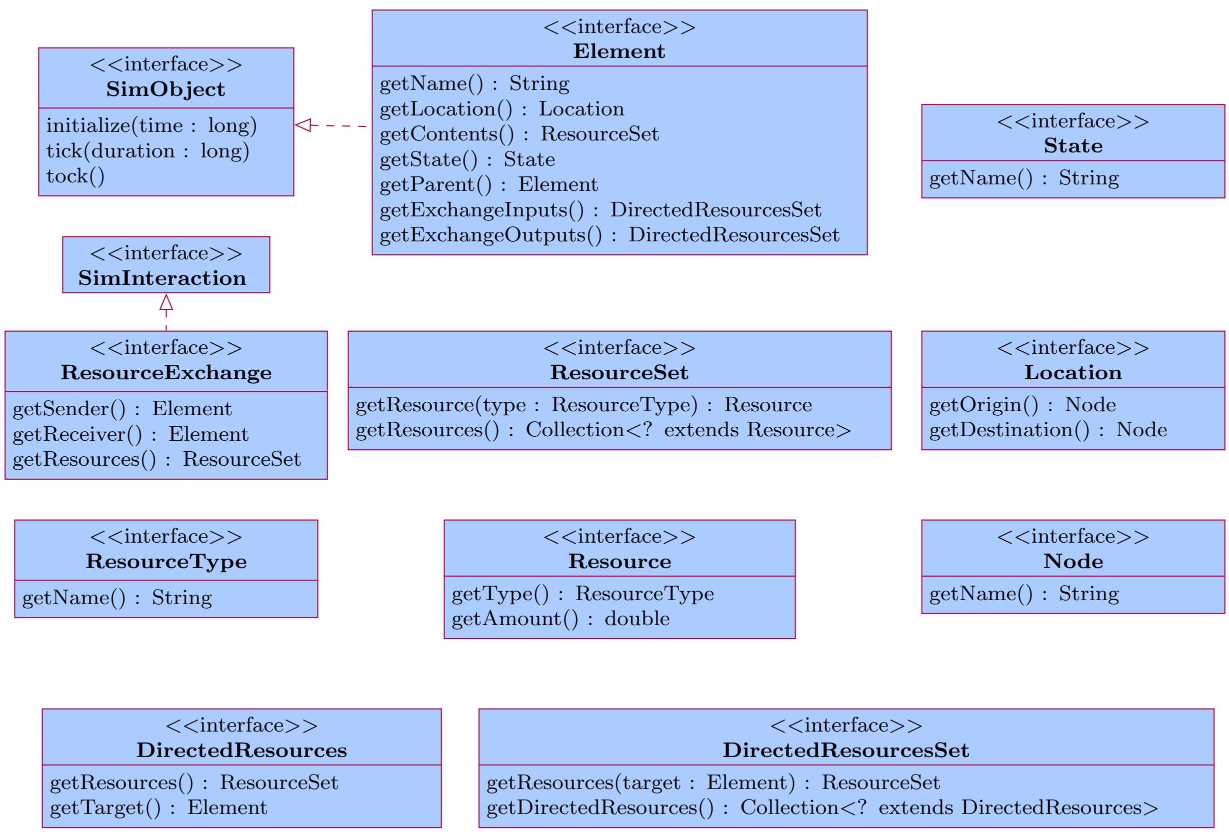

Figure 5-10: Object class diagram for core ISoS model interfaces. Node and ResourceType classes use name-based identification associated with string data types. Includes both interaction-based and attribute-based resource exchanging behaviors.

associated `DirectedResourcesSet` and `DirectedResources` data types. The `ResourceExchange` interface implements the `SimInteraction` interface and includes methods to access the sending and receiving elements and the sent resources.

5.5.2 Default Model Implementation

The default model implementations in Figure 5-11 identified with a `*Impl` suffix specify object classes adhering to the core interfaces with minimal functionality. These object classes could be used as the basis for application-specific model implementations to reduce development effort.

The embodiment of a federate is the `FederateImpl` class which contains all local model instantiations and manages a simulation execution. Its data members store allowable locations, instantiated elements, and the associated federate ambassador implementation to coordinate HLA services. It also includes methods to add, update, and remove objects, send interactions, and execute a simulation by specifying the initial and final times, time step, and pseudo time step. The simulation formalism is best described as an agent-based network as each object (elements in ISoS) independently updates itself via the associated `tick` and `tock` methods. Alternative federate implementations could use other formalisms such as discrete event simulation (i.e. the federate contains a list of events to process in a centralized manner) or system dynamics (i.e. the federate contains a stock-and-flow model specified by a centralized system of equations).

The `ElementImpl` class includes data members to store its location, state, parent element, contents, exchange inputs and outputs, and a set of allowable states using local model implementations. Additional data members with a `next*` prefix temporarily store values computed during the `tick` method before overwriting the associated data members during the `tock` method call.

The `ResourceExchangeImpl` class includes data members for the sending and receiving elements and the associated resources. It avoids implementation-specific element data members to allow for any element (local or remote) to be targeted with a resource exchange interaction.

The `DirectedResourcesSetImpl` class is backed by an array list of directed resources which are implemented in class `DirectedResourcesSetImpl` with data members for the target element and resources. Unlike the resource exchange interaction, the directed resources set uses default model implementations to specify the resources.

The `ResourceImpl` class stores resource amounts using a primitive double data member. While a simple implementation, it is also limited in certain cases by the inability for default encoding to exactly express all numbers. More precise applications may consider alternative implementations such as the `BigDecimal` Java data type. The `ResourceSetImpl` class is backed by an array list. Alternative implementations may use a set or array of resources resource or even a map between resource types and double quantities. The `ResourceSetImpl` and `ResourceImpl` classes introduce new methods to add, subtract, and multiply resources to serve as convenience methods.

The `ResourceTypeImpl`, `NodeImpl`, and `StateImpl` classes are minimal implementations which only store the associated name as a String data member. Classes including `ResourceSetImpl`, `ResourceImpl`, `ResourceTypeImpl`, `LocationImpl`, `NodeImpl`, `DirectedResourcesSetImpl`, and `DirectedResourcesImpl` are implemented as immutable, or unchangeable, objects. In other words, their data members cannot be modified once an object is instantiated. Thus, for resource operations such as addition, subtraction, and multiplication and for methods returning a collection, a new object instance is returned rather than modifying an existing one.

Several classes require methods to override default behavior for `hashCode` and `equals` Java

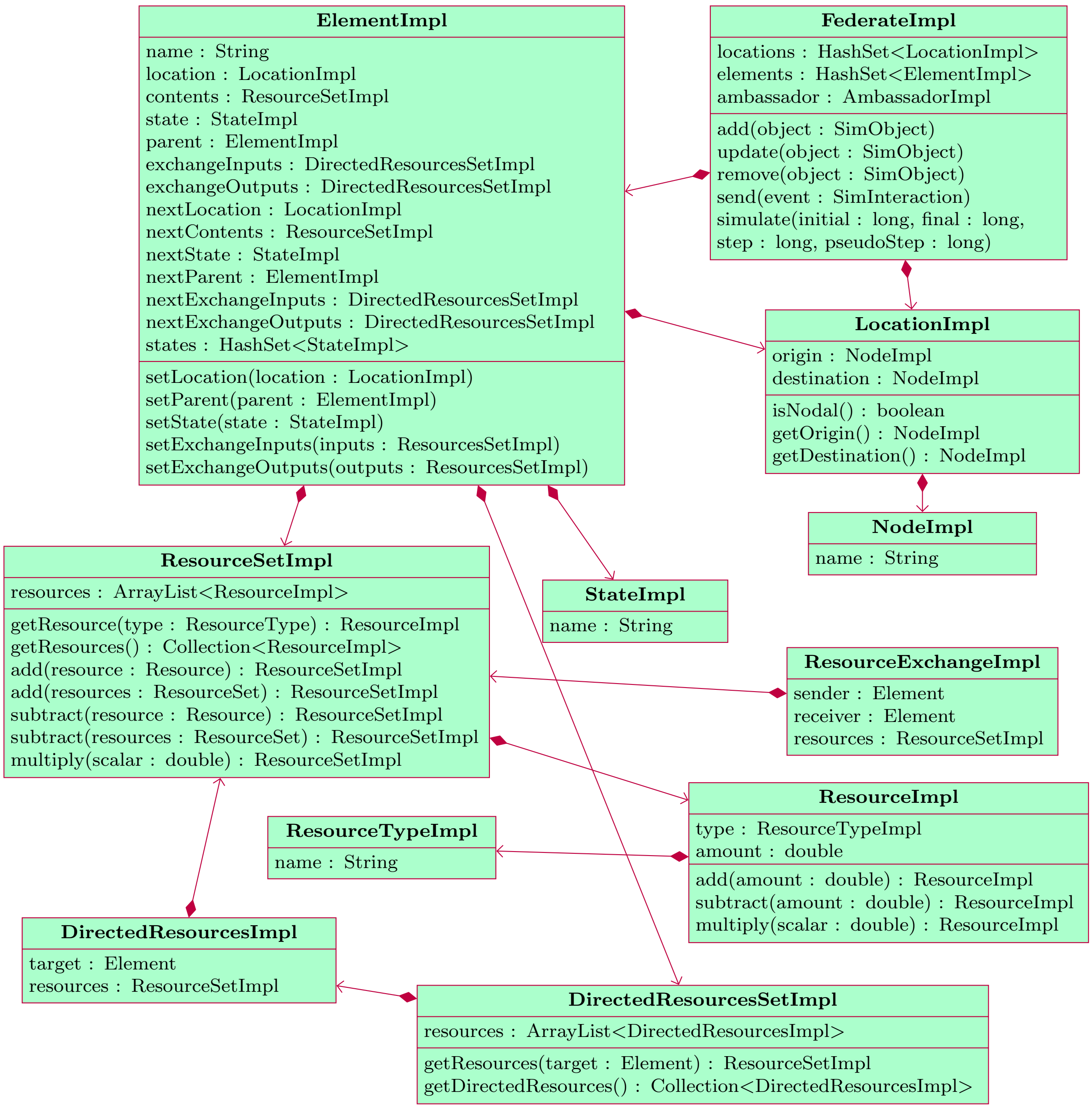

Figure 5-11: Object class diagram for a default model implementation. Resource amounts are implemented as a primitive double data type.

methods. For example, two Node objects, two ResourceType objects, or two State objects shall be equal if they have a common name. Similarly, two Location objects shall be equal if they have equal origin nodes and destination nodes. Resource objects shall be equal if they have equal resource types and amounts. Two ResourceSet objects shall be equal if they contain equal sets of resources with non-zero amounts. Two DirectedResources objects shall be equal if they contain equal resource sets and target elements. Finally, two DirectedResourcesSet objects shall be equal if they contain equal sets of directed resources.

5.5.3 HLA Model Implementation

The HLA model implementations in Figure 5-12 identified with a HLA* prefix specify object classes adhering to the core interfaces and HLA-related interfaces. As a complete description of the federation interface, these classes need not be modified for any federate implementation using the Java programming language.

The class AmbassadorImpl implements the HLA standard FederateAmbassador interface to interact with the RTI. Service requests are issued via an RTIambassador data member and callback services are received via interface methods. Additionally, the federate ambassador provides methods for the required processes detailed in the federation agreement including start-up, advance, reset, and shut-down. Additional methods register, update, and delete objects and send interactions using local model implementations. Although not discussed here, an observer pattern such as the EventListener and EventObject can be used to notify the federate of HLA services.

While the entire HLA functionality could be included in a FederateAmbassador implementation, a modular structure presented leverages abstraction for future extensibility. Abstract object classes HLAobject and HLAinteraction support the federate ambassador to provide low-level functionality. HLAobject contains data members for the RTI ambassador, HLA class, attribute, and object handles, the assigned instance name, and a boolean determining if the object is local (i.e. corresponding to a local model implementation) or remote. Methods provide an interface to delete local or remote objects, request attribute value updates of remote objects, set attributes from local objects or from HLA attribute handle value maps, or to update local attributes. Similarly, HLAinteraction contains data members for the RTI ambassador and the HLA class handle. Methods provide an interface to set parameters from local interactions or from HLA parameter handle value maps, and to send local interactions.

The HLAelement and HLAresourceExchange classes extend the abstract classes to add ISoS interface-specific data members. The HLAelement class corresponds to the ISOSelement object in the FOM and includes data elements for the name, location, contents, state, and parent element. The HLAresourceExchange class corresponds to the ISOSresourceExchange interaction in the FOM and includes data elements for the names and object references of sending and receiving elements and the set of exchanged resources. Whereas the element names are assigned via HLA, the object references are assigned by the federate ambassador when a remote interaction is received.

The data structure classes HLAstate, HLAlocation, HLAnode, HLAresourceSet, HLAresource, HLAresourceType, HLAdirectedResources, and HLAdirectedResourcesSet correspond to ISOS* data type entries in the FOM and implement the standard HLA interface DataElement to act as HLA-compatible data types. At the lowest level, each class uses standard HLA data types such as HLAvariableArray, HLAfixedArray, and HLAfixedRecord for composite data members and HLAunicodeString and HLAfloat64BE for primitive data members.

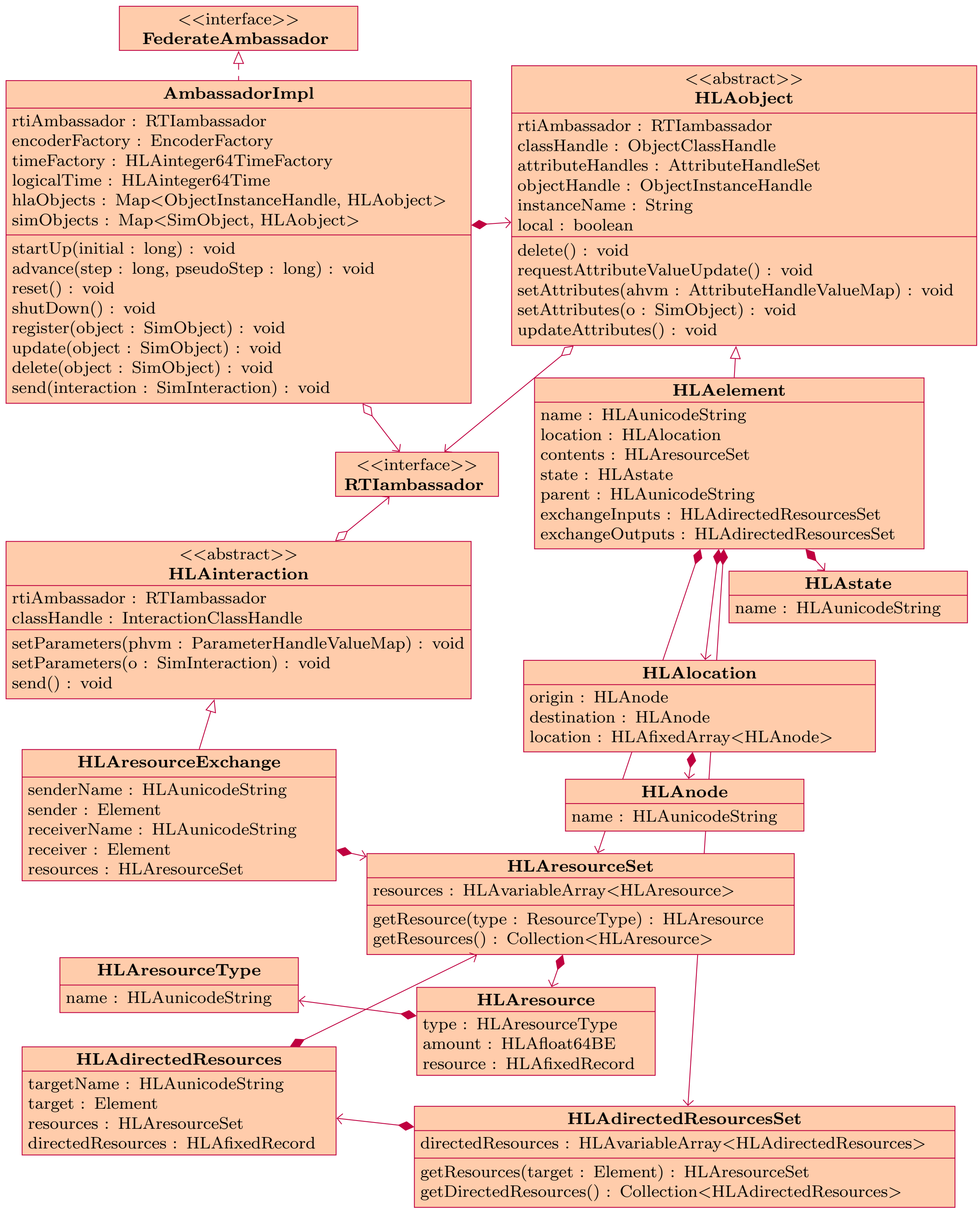

Figure 5-12: Object class diagram of HLA model implementation.

5.6 Practical Implications for Distributed Simulation

This chapter closes with a brief discussion of some practical implications for designing and executing a distributed simulation. These comments are specific to the HLA standard; however similar challenges would be expected in other software architectures as well. Experiences are drawn both from the research presented and from participation in the *Simulation Smackdown*[5] outreach event organized by the Simulation Interoperability Standards Organization (SISO) during the 2010, 2011, and 2012 academic years. This event allowed student teams from around the world to create federates for a lunar exploration distributed simulation. For more information on the MIT team federates and applications, please see Essilfie-Conduah et al. (2011).

5.6.1 On Software Complexity

Distributed simulation using the HLA requires more complex software as compared to a sequential simulation. While some challenges are due to the parallelism of concurrent model execution, others are due to the additional layers of software involved and the wide scope of the HLA standard.

While the HLA ensures federation synchronization via algorithms implemented in the RTI, each federate must also maintain synchronization of its internal state variables. Especially if using the multi-threaded mode of HLA, the "callback" services (e.g. object discovery and removal, attribute value reflection, and receiving interaction events) are executed by a separate thread from the simulation program. In these conditions, mutable state variables must be carefully managed to prevent unexpected access to non-threadsafe data structures.[6] Even if using the single-threaded callback mode, one must take extra considerations to ensure external model data is robust to missing or delayed updates from third party federates.

In addition to managing the parallel federation execution, there are extra layers of software involved in the HLA. Most RTI implementations include CRC and LRC applications which must be installed and configured separately from the simulation application. There may be a significant amount of set-up and configuration required to run sample federate programs packaged with the RTI. Once running, tracking down and debugging errors or problems is made more difficult by the layers of applications outside the federate program itself. For example, the MIT team's *Simulation Smackdown* simulation federate was developed in MATLAB using a separate middle-ware application to connect with the LRC's C++ bindings. At one point the teams experienced a problem with data encoding/decoding which could not easily be traced between possible sources in our federate implementation, the MATLAB middle-ware, the LRC implementation, in other federates' implementation, or in other federates' middle-ware.

Finally, the wide scope and generality of the HLA standard makes it difficult to apply to a particular case. Navigating the dozens of potential services and hundreds of API calls for general applications and finding the "minimal" set required to implement a federate is a significant challenge for novice developers. While Section 5.4 identifies the subset of services used in the infrastructure system-of-systems application, novice developers should start with much simpler applications to learn about the HLA services and incrementally add functionality to build experience with object management, data management, and finally time management. In addition to large scope of the

[5]As of the 2013–2014 academic year this event is called the *Simulation Exploration Experience*.

[6]For readers interested in using multi-threaded HLA or GUI-based user interaction, Goetz et al. (2006) is strongly recommended as an overview of concurrency in Java software.

HLA, there are also limited tutorial exercises available for self-learning. As an attempt to produce new educational materials, a short course on modeling and simulation was developed and taught at MIT during January 2012. Most students familiar with object-oriented programming could develop a time-managed HLA federate with one intensive week (20 class hours) of guided tutorials.

5.6.2 On Network Infrastructure

Distributed simulation leverages network infrastructure for communication between federates. In addition to hardware limitations of bandwidth and latency, privacy and security implications in network infrastructure also contribute practical challenges.

Hardware limitations of bandwidth (data throughput) and latency (response time) are important constraints in distributed simulation. Even with modern high-speed networks, one must consider the total federation bandwidth requirements when specifying the quantity and frequency of data to exchange. The challenge grows if remote federates are hosted on low-capacity (e.g. old, residential, hotel, or other public access) or high-latency (geographically distant or noisy wireless) networks. As an example, the *Simulation Smackdown* used a rate of one update per second for simple data types including floating-point position/attitude vectors and short text strings. While testing should identify requirements for federates, runtime congestion may not be predictable in advance for shared networks.

There are additional network components which may interact negatively with distributed simulation. Firewalls are network devices or software applications running within an operating system to prevent malicious or unwanted traffic. Often, default firewall configurations do not recognize and correspondingly block portions of distributed simulation network traffic. Especially in the case of the HLA where no wire-level protocol is specified in the standard, i.e. the implementation can use any method, a recommended practice is to disable software firewalls on the machines running federates. Unfortunately, disabling firewalls also exposes machines to malicious network traffic, requiring additional security considerations. In addition, there may be privacy concerns for the content of the distributed simulation messages which may not be encrypted. One approach to address both problems is to only run distributed simulations on a closed, private network to prevent unauthorized network traffic. Another practical solution is to use a virtual private network (VPN) which can be tunneled through unsecured network connections to provide a comparable level of security and privacy with existing network infrastructure.

5.6.3 On Testing and Debugging

Testing and debugging are major elements of any software development, and distributed simulation complicates the process with its decentralization of software components. During federation development, progressive testing includes standalone, pairwise, and integrated approaches.

Standalone testing involves executing a federate in an isolated manner with the objective of identifying internal problems. It is the least costly, but most limited testing as it can be performed independently from other parties. As other federates are not available in standalone testing, any critical dependencies should be represented with a simple federate providing basic functionality. For example, in the *Simulation Smackdown* federation, a third party federate provides environmental ephemerides (relative position of various celestial bodies) and manages time advancement. For standalone testing of our federates a simple environmental federate was created to provide placeholder data and advance time.

Pairwise testing involves executing two federates concurrently with the objective of identifying interaction problems. This is the first evaluation of the federation agreement which prescribes the expected behavior of each federate. Particular items to investigate include discovery/removal of object instances, updating/reflection of attribute values, and sending/receiving of interactions. For each case, one should verify that data is being encoded and decoded properly and all assumptions in the federation agreement are upheld. Pairwise testing can also identify some network problems such as firewall restrictions and connectivity issues.

Finally, integrated testing involves executing all member federates in a full-up session with the objective of identifying architectural problems. It is the most costly testing as all parties must participate at the same time. Combinations of interactions between federates may give rise to unexpected emergent effects when considering the entire federation. Integrated testing also evaluates architectural network problems such as bandwidth/latency limitations.

5.6.4 On Collaboration

The presence and necessity of collaboration between quasi-independent teams is a major difference between monolithic and federated simulation. While collaboration is ultimately the intent of interoperable simulation gaming, it is not easy to achieve. The first collaborative product should be the federation definition including the FOM and federation agreement. These items should be as simple as possible, yet complete in description and requirements, to allow members to fully implement the required functionality. Fortunately, as applied to the ISoS modeling framework, the required items are somewhat limited to the nodes, resource types, and general assumptions governing each infrastructure system federate.

Collaboration is also required to complete the pair-wise and integrated testing sessions described in the previous section. During the *Simulation Smackdown* events, this was especially challenging due to geographic distribution of teams across time zones and limited communication methods. Our use of a VPN for simulation execution restricted other network traffic, requiring additional infrastructure of conference calls and, later, secure voice-over-IP solutions from a separate computer with a regular Internet connection. Even though federated simulation can operate over WANs, face-to-face contact is strongly encouraged if possible to support collaborative processes.

References for Chapter 5

Apache Software Foundation (2013). *Apache Hadoop*. URL: `http://hadoop.apache.org/` (visited on 09/18/2013).

Bettner, P. and M. Terrano (2001). "1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond". In: *Game Developers Conference 2001*.

Bharambe, A. R., J. Pang, and S. Seshan (2005). *A Distributed Architecture for Interactive Multiplayer Games*. Tech. rep. CMU-CS-05-112. Pittsburgh, Pennsylvania: Carnegie Mellon University. URL: `http://www.cs.cmu.edu/~ashu/papers/cmu-cs-05-112.pdf` (visited on 09/09/2013).

Boer, C., A. de Bruin, and A. Verbraeck (2009). "A survey on distributed simulation in industry". In: *Journal of Simulation* 3, pp. 3–16.

Boer, C. A. (2005). "Distributed Simulation in Industry". PhD thesis. Delft, Netherlands: Technische Universiteit Delft.

Brailsford, S., K. Katsaliaki, N. Mustafee, and S. Taylor (2006). "Modelling very large complex systems using distributed simulation: A pilot study in a healthcare setting". In: *Proceedings of the 2006 Operational Research Society Simulation Workshop*, pp. 257–262. URL: http://bura.brunel.ac.uk/handle/2438/4002.

Bruzzone, A. G., R. Mosca, and R. Revetria (2005). "Agent Directed HLA Simulation for Complex Supply Chain Modeling". In: *Simulation* 81.9, pp. 647–655. DOI: 10.1177/0037549704047602.

Bryant, R. (1977). *Simulation of Packet Communication Architecture Computer Systems*. Cambridge, Massachusetts: Computer Science Laboratory, Massachusetts Institute of Technology.

Bush, B., L. Dauelsberg, R. LeClaire, D. Powell, S. DeLand, and M. Samsa (2005). "Critical Infrastructure Protection Decision Support System (CIP/DSS) Project Overview". In: *2005 International System Dynamics Conference*. LA-UR-05-1870. Boston, Massachusetts.

Chandy, K. and J. Misra (1978). "Distributed simulation: A case study in design and verification of distributed programs". In: *IEEE Transactions on Software Engineering* 5.5, pp. 440–452.

Cronin, E., B. Filstrup, and A. Kurc (2001). *A Distributed Multiplayer Game Server System*. URL: http://warriors.eecs.umich.edu/games/papers/quakefinal.pdf (visited on 09/12/2013).

Dahmann, J. S., R. M. Fujimoto, and R. M. Weatherly (1997). "The Department of Defense High Level Architecture". In: *Proceedings of the 1997 Winter Simulation Conference*. Ed. by S. Andradottir, K. J. Healy, D. H. Withers, and B. L. Nelson, pp. 142–149.

Dean, J. and S. Ghamawat (2008). "MapReduce: Simplified Data Processing on Large Clusters". In: *Communications of the ACM* 51.1, pp. 107–113.

DIS Steering Committee (1994). *The DIS Vision: A Map to the Future of Distributed Simulation*. Tech. rep. IST-SP-94-01. Institute for Simulation & Training.

Essilfie-Conduah, N., P. T. Grogan, P. M. Cunio, R. McLinko, and O. L. de Weck (2011). "A University Perspective on the NASA/SISO Smackdown Modeling and Simulation Outreach Event". In: *2011 Fall Simulation Interoperability Workshop*. 11F-SIW-031. Orlando, Florida.

Eusgeld, I., C. Nan, and S. Dietz (2011). "'System-of-systems' approach for interdependent critical infrastructures". In: *Reliability Engineering and System Safety* 96, pp. 679–686. DOI: 10.1016/j.ress.2010.12.010.

Fiedrich, F. (2006). "An HLA-based multiagent system for optimized resource allocation after strong earthquakes". In: *Proceedings of the 2006 Winter Simulation Conference*. Ed. by L. Perrone, F. Wieland, J. Liu, B. Lawson, D. Nichol, and R. Fujimoto, pp. 486–492. URL: http://dl.acm.org/citation.cfm?id=1218112.1218205.

Fielding, R. T. and R. N. Taylor (2002). "Pricipled Design of the Modern Web Architecture". In: *ACM Transactions on Internet Technology* 2.2, pp. 115–150.

Fujimoto, R. M. (2000). *Parallel and Distributed Simulation Systems*. New York: John Wiley & Sons.

Gautier, L. and C. Diot (1997). *MiMaze: A Multiuser Game on the Internet*. Tech. rep. RR-3248. INRIA. URL: http://hal.inria.fr/inria-00073441/.

Goetz, B., T. Peierls, J. Bloch, J. Bowbeer, D. Holmes, and D. Lea (2006). *Java Concurrency in Practice*. Addison-Wesley Professional.

Hibino, H., Y. Yura, Y. Fukuda, K. Mitsuyuki, and K. Kaneda (2002). "Manufacturing modeling architectures: manufacturing adapter of distributed simulation systems using HLA". In: *Proceedings of the 2002 Winter Simulation Conference*. Ed. by E. Ycesan, C.-H. Chen, J. Snowdon, and J. Charnes, pp. 1099–1107.

Hopkinson, K., X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury (2006). "EPOCHS: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components". In: *IEEE Transactions on Power Systems* 21.2, pp. 548–558. DOI: 10.1109/TPWRS.2006.873129.

IEEE (1998). *IEEE Standard for Distributed Interactive Simulation – Application Protocols*. IEEE Std. 1278.1a-1998.

— (2010a). *IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)*. IEEE Std. 1730-2010.

— (2010b). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification*. IEEE Std. 1516.1-2010.

— (2010c). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*. IEEE Std. 1516-2010.

— (2010d). *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (OMT) Specification*. IEEE Std. 1516.2-2010.

Jacobs, P. H. M. (2005). "The DSOL Simulation Suite: Enabling Multi-formalism Simulation in a Distributed Context". PhD thesis. Delft, Netherlands: Technische Universiteit Delft.

Jain, S. and C. McLean (2003). "Simulation for emergency response: a framework for modeling and simulation for emergency response". In: *Proceedings of the 2003 Winter Simulation Conference*. Ed. by S. Chick, P. Snchez, D. Ferrin, and D. Morrice, pp. 1068–1076. URL: http://dl.acm.org/citation.cfm?id=1030818.1030960.

Jefferson, D. (1985). "Virtual time". In: *ACM Transactions on Programming Languages and Systems* 7.3, pp. 404–425. DOI: 10.1145/3916.3988.

Joyent, Inc. (2013). *Node.js*. URL: http://nodejs.org/ (visited on 09/18/2013).

Liu, K., X. Shen, N. D. Georganas, A. Boukerche, and A. Boukerche (2007). "SimSITE: The HLA/RTI Based Emergency Preparedness and Response Training Simulation". In: *11th IEEE International Symposium on Distributed Simulation and Real-Time Applications*, pp. 59–63. DOI: 10.1109/DS-RT.2007.33.

Nan, C. and I. Eusgeld (2011). "Adopting HLA standard for interdependency study". In: *Reliability Engineering and System Safety* 96.1, 149–159. DOI: 10.1016/j.ress.2010.08.002.

Open HLA (2013). *Open HLA*. URL: http://sourceforge.net/projects/ohla/ (visited on 09/18/2013).

Pederson, P., D. Dudenhoeffer, S. Hartley, and M. Permann (2006). *Critical Infrastructure Interdependency Modeling: A Survey of U.S. and International Research*. Tech. rep. INL/EXT-06-11464. Idaho Falls, Idaho: Idaho National Laboratory.

Photon (2013). *Understanding Photon Server*. URL: http://doc.exitgames.com/photon-server/ (visited on 09/09/2013).

RakNet (2012). *RakNet Manual*. URL: http://www.jenkinssoftware.com/raknet/manual/ (visited on 09/09/2013).

Reid, M. R. (2000). "An Evaluation of the High Level Architecture (HLA) as a Framework for NASA Modeling and Simulation". In: *25th NASA Software Engineering Workshop*. Greenbelt, Maryland. URL: http://hdl.handle.net/2060/20010016107.

Rinaldi, S. M., J. P. Peerenboom, and T. K. Kelly (2001). "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies". In: *IEEE Control Systems Magazine*, pp. 11–25.

Schulze, T., S. Straßburger, and U. Klein (1999). "Migration of HLA into Civil Domains: Solutions and Prototypes for Transportation Applications". In: *Simulation* 73.5, pp. 296–303. DOI: 10.1177/003754979907300506.

Simons, M., S. Stalnaker, and C. Morgan (2013). "A High-level Architecture Framework for Air Traffic Management (ATM) Operations". In: *Proceedings of the 2013 International Systems Conference*, pp. 878–885.

Smed, J., T. Kaukoranta, and H. Hakonen (2002). "Aspects of networking in multiplayer computer games". In: *The Electronic Library* 20.2, pp. 87–97.

Specht, R. D. (1957). *War Games*. Tech. rep. P-1041. Santa Monica, California: The RAND Corporation.

Straßburger, S. (2001). "Distributed Simulation Based on the High Level Architecture in Civilian Application Domains". PhD thesis. Magdeburg, Germany: Otto-von-Guericke-Universität Magdeburg.

Sweet, D. N., V. Manikonda, J. S. Aronson, K. Roth, and M. Blake (2002). "Fast-time Simulation System for Analysis of Advanced Air Transportation Concepts". In: *AIAA Modeling and Simulation Technologies Conference and Exhibit*. AIAA-2002-4593. Monterey, California.

Sztipanovits, J., G. Hemingway, A. Bose, and A. Srivastava (2012). "Model-based integration technology for next generation electric grid simulations". In: *2012 IEEE Power and Energy Society General Meeting*, pp. 1–8. DOI: 10.1109/PESGM.2012.6345323.

The Portico Project (2013). *The Portico Project*. URL: http://www.porticoproject.org/ (visited on 09/18/2013).

Tolone, W. J., E. W. Johnson, S.-W. Lee, W.-N. Xiang, L. Marsha, C. Yeager, and J. Blackwell (2008). "Enabling System of Systems Analysis of Critical Infrastructure Behaviors". In: *Proceedings of the Third International Workshop on Critical Infrastructure Security*. Rome.

Twisted Matrix Labs (2013). *Twisted*. URL: http://twistedmatrix.com/ (visited on 09/18/2013).

Valve Developer Community (2012). *Source Engine*. URL: https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking (visited on 09/18/2013).

Wilson, A. L. and R. M. Weatherly (1994). "The Aggregate Level Simulation Protocol: An Evolving System". In: *Proceedings of the 1994 Winter Simulation Conference*. Ed. by J. Tew, S. Manivannan, D. Sadowski, and A. Seila, pp. 781–787.

Zajac, K., A. Tirado-Ramos, Z. Zhao, P. Sloot, and M. Bubak (2004). "Grid Services for HLA-Based Distributed Simulation Frameworks". In: *Grid Computing*. Ed. by F. F. Rivera, M. Bubak, A. G. Tato, and R. Doallo. Vol. 2970. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 147–154. DOI: 10.1007/978-3-540-24689-3_19.

Chapter 6

The Sustainable Infrastructure Planning Simulation Game (SIPS-G)

> "Public policy-makers deal with difficult problems in our complex society every day. Unfortunately, and seemingly related to the number of facts that compete for attention, these problems are becoming increasingly unmanageable. Solutions, even when they can be formulated, regularly create unimagined new problems. Indeed, complexity challenges the very essence of effective and legitimate control in society today."
>
> Garry D. Brewer in *Politicians, Bureaucrats, and the Consultant* (1973, p. 3)

This chapter uses the Kingdom of Saudi Arabia as a case study in national infrastructure planning using interoperable simulation gaming methods. Saudi Arabia echoes many common tensions with other nations in the Middle East and Northern Africa (MENA) region: rapid population growth and urbanization, limited natural resources, and strong interdependencies between infrastructure sectors (Siddiqi and Anadon 2011). But a global leader in petroleum export, Saudi Arabia is fortunate to have the financial means for massive infrastructure improvements. The infrastructure planning challenge in Saudi Arabia is to efficiently apply its petroleum-based financial wealth to provide long-lasting resource infrastructure for its human capital.

This chapter is organized as follows. First, Section 6.1 introduces the sustainability and complexity challenges to infrastructure system design in Saudi Arabia as background information and motivation for study. Next, Section 6.2 formulates the Sustainable Infrastructure Planning Simulation Game (SIPS-G) using the ISoS modeling framework presented in Chapter 4. Section 6.3 presents quantitative mathematical models to drive behavior of component infrastructure system models. Finally, Section 6.4 presents a detailed instantiation of the simulation models to create a baseline scenario corresponding to Saudi Arabian infrastructure between 1950 and 2010. For further details, Appendix B describes the SIPS-G application implementation using the software architecture presented in Chapter 5.

6.1 Infrastructure in Saudi Arabia

The Arabian Peninsula is a harsh habitat for human societies. The majority of land area is subsumed by a hot, arid desert climate with few accessible resources. Throughout history societies gathered around natural sources of water such as oases and wadis (desert valleys) where water

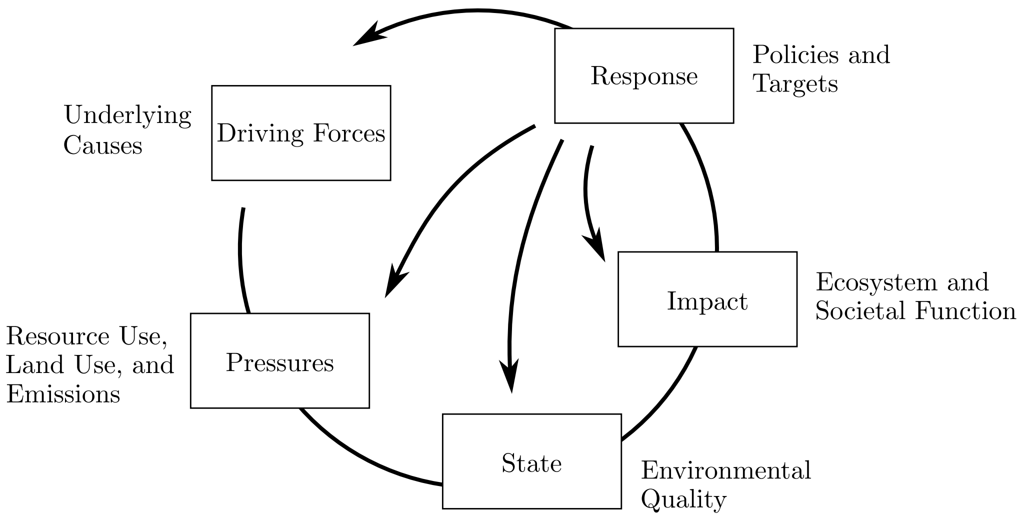

Figure 6-1: The DPSIR framework formulates sustainability through a causal chain of driving forces, environmental pressures, environmental state, impact on human activities, and response.

aquifers reach near the surface and along the coast where humid and cooler air prevail. A dramatic change started to take place in the mid-20th century with the discovery and extraction of large petroleum reserves. Once a resource-starved land, some nations found immense monetary wealth in the export of oil and natural gas to the rest of the world.

The scope of human societies expanded greatly over the past sixty years, driven by rapidly-growing populations and a transition from nomadic tribal to urbanized city life. Large and costly infrastructure systems financed in part through oil export revenue now supply the majority of critical resources. Today, major city inhabitants benefit from comparative luxuries of potable water, paved streets, a wide selection of imported food and goods, and air conditioned buildings.

While prosperity has provided significant benefits, there is concern for the sustainability of current activities and a need for efficient use of limited natural resources. Planning future infrastructure systems requires careful consideration of both technical and social factors over long time periods. The overarching objective of this work is to establish new methods and tools to assist in the planning process, emphasizing the use of modeling and simulation to gain early insights.

6.1.1 Sustainability Challenges through the DPSIR Framework

The DPSIR framework, illustrated in Figure 6-1, explains the causal links between driving forces, pressures, state, impacts, and responses in the context of sustainability (Kristensen 2004). Applied to Saudi Arabia, it describes the items contributing to the response actions including the new infrastructure projects studied in this case.

Driving Forces

The driving forces in Saudi Arabia are common to many other countries around the world; namely, a rapidly growing and urbanizing population. Figure 6-2 illustrates the population in Saudi Arabia growing from 3.1 million in 1950 to over 27 million in 2010 (UNPD 2013), corresponding to a 3.7%

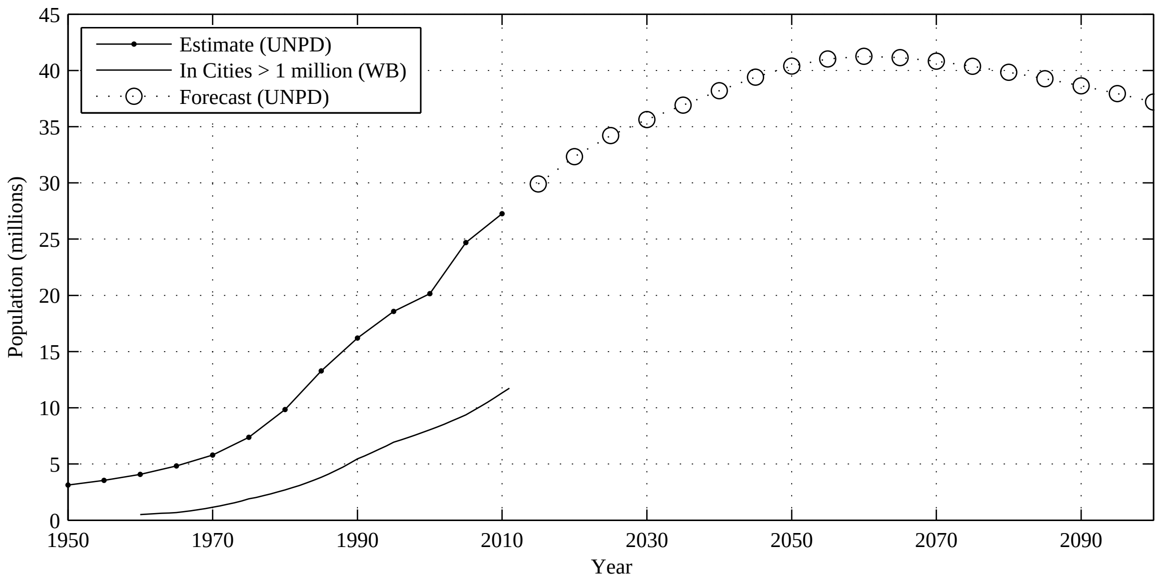

Figure 6-2: The population of Saudi Arabia is expected to exceed 40 million by 2050. Data sources: Estimated, Forecast Population: UNPD 2013; Cities greater than 1 million: World Bank 2012.

annual growth rate (2.6% between 1990 and 2010). Furthermore, median forecasts exceed 40 million by 2050 (UNPD 2013). While population in aggregate is growing, even more people are moving to large cities. Between 1960 and 2010, the number of people living in cities of over 1 million increased at an average of 6.4% per year (World Bank 2012), far out-pacing the population growth rate.

Along with urbanization, consumption of critical resources including water, food, and energy is increasing. Figure 6-3 highlights per-capita use of resources normalized to 1980 values such that constant levels correspond to a 2-3% annual growth rate to match population. Per-capita non-agricultural (municipal and industrial) water use stabilized over the past 20 years at approximately twice that of 1980 (KSA 1990; KSA 1995; KSA 2000; KSA 2005; KSA 2010). Per-capita agricultural water use has fallen in recent years but remains about three times the levels in 1980 when government-sponsored programs spurred agricultural development (KSA 1990; KSA 1995; KSA 2000; KSA 2005; KSA 2010). Per-capita food supply increased nearly 50% since 1980 due to increasing use of feed for animal products (FAO 2013b). By far the most significant increases come from the energy sector: per-capita energy use doubled between 1980 and 2010 while per-capita electricity consumption more than quadrupled over the same time period (World Bank 2012). In 2008 53% of electricity was used for residential purposes (KSA 2010) and some estimate air conditioning consumes 70% of all electricity (Hasnain 1998).

Pressures

Massive infrastructure expansion to meet growing demands impose pressures on the environment. Figure 6-4 highlights environmental pressures of non-renewable water withdrawals, oil production, and carbon dioxide emissions normalized to 1980 values. The most apparent pressure is in withdrawals of non-renewable "fossil" water peaking at nearly 15 billion cubic meters withdrawn in 1994, mostly for agricultural irrigation (KSA 1995). The energy system exerts another main pres-

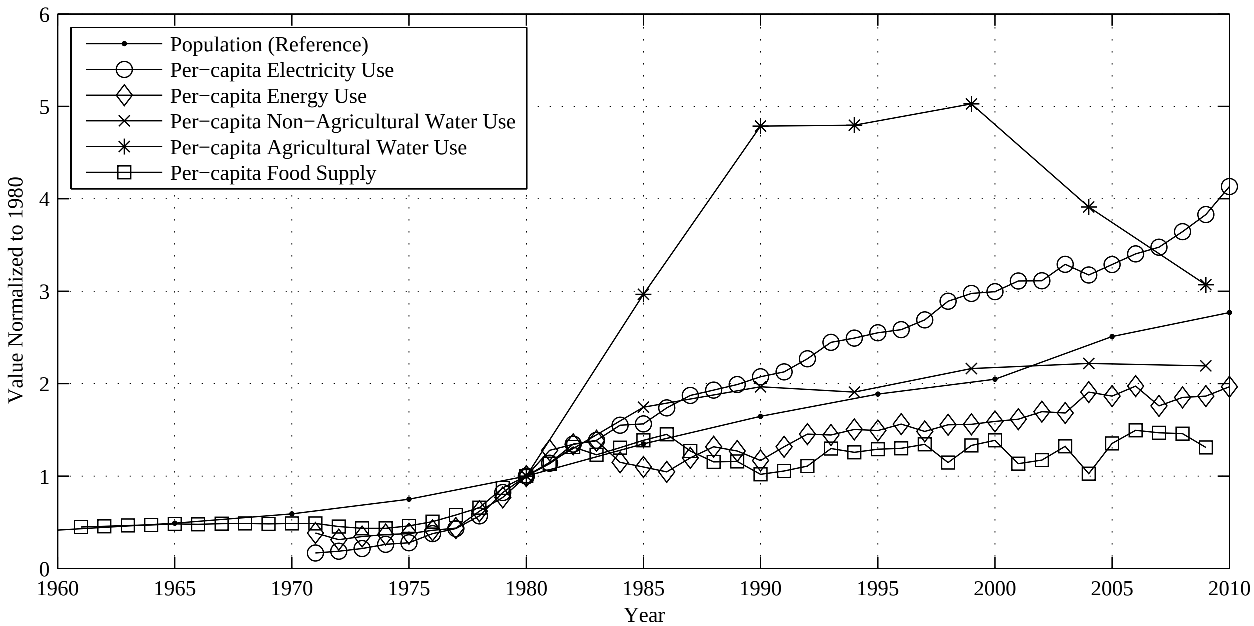

Figure 6-3: Driving forces increase per-capita resource use in Saudi Arabia over time. Values for population, electricity, energy, water, and food use are normalized to their 1980 values for comparison. Data sources: Population: UNPD 2013; Electricity and Energy Use: World Bank 2012; Water Use: KSA 1990; KSA 1995; KSA 2000; KSA 2005; KSA 2010; UNPD 2013; Food Supply (includes food, feed, and seed): FAO 2013b.

sure as Saudi Arabia is among the world's largest oil producers with 547 million metric tons of oil withdrawn from natural reserves in 2012 (BP 2013). Furthermore, as of 2010, 100% of Saudi Arabia's electricity was generated from fossil fuels, contributing to a doubling in carbon dioxide emissions since 1980 (World Bank 2012).

State

Pressures contribute to changes in the environment state. With no recharging capability, use of non-renewable aquifers directly and permanently affects groundwater supply. Various estimates place the non-renewable aquifer volume in Saudi Arabia around the year 2000 at 400-700 billion cubic meters of varying quality (KSA 2005; World Bank 2005). These estimates indicate it is impossible to sustain large withdrawals (2-4% of total volume annually) seen in recent years. As for oil, proven reserves are reported at 265.9 thousand million metric tons (15.9% of world total) at the end of 2012, such that the nominal production extracts 0.2% of the total volume annually (BP 2013). The heavy reliance on fossil fuels for electricity generation and transportation also affects the atmospheric environmental state. The carbon dioxide contribution of Saudi Arabia in 2009 was 16 metric tons per person or 0.75 kilograms per dollar of economic output (World Bank 2012), higher than the global average for developed countries in 2010 of 11 metric tons per person or 0.6 kilograms per dollar of output (UN 2013).

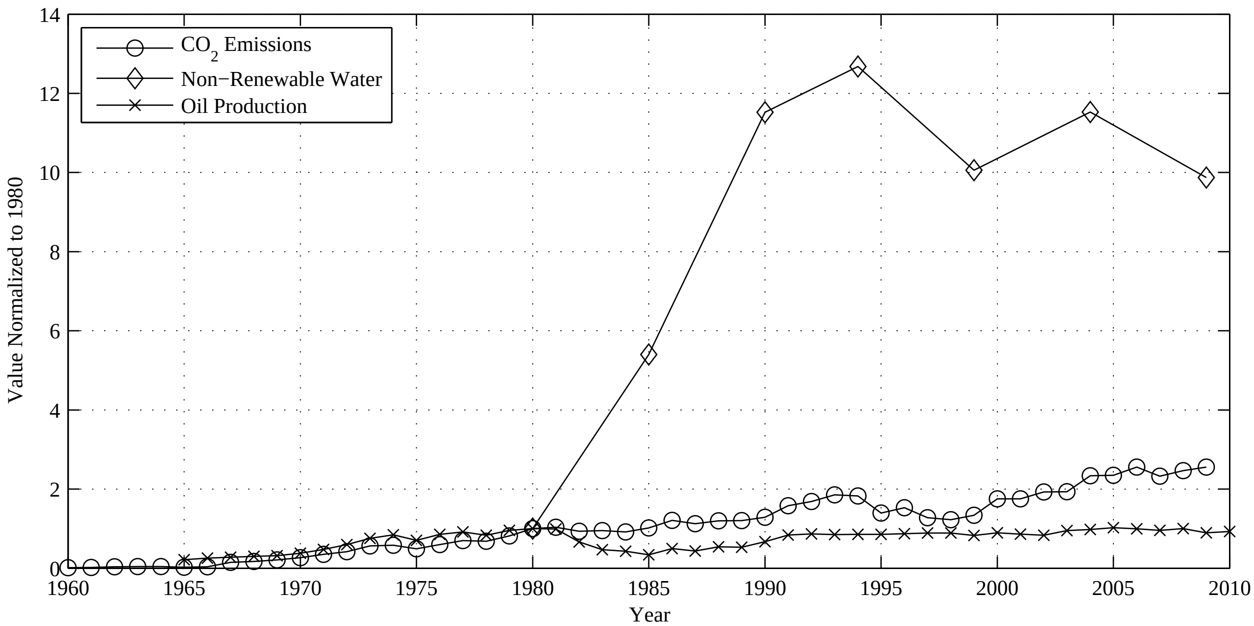

Figure 6-4: Pressures changes the environmental state. Values for carbon dioxide emissions, non-renewable water withdrawals, and oil production are normalized to their 1980 values for comparison. Data sources: CO_2 Emissions: World Bank 2012; Water Withdrawals: KSA 1990; KSA 1995; KSA 2000; KSA 2005; KSA 2010; UNPD 2013; Oil Production: BP 2013.

Impact

Potential impacts from changes to the environmental state are wide-reaching and significant. Continued depletion of non-renewable water aquifers reduces the quantity *and* quality of available water, increases the pumping cost as aquifer levels fall, and may contribute to an inability to domestically cultivate food. Continued domestic oil consumption both decreases proven reserves and export capability, the main source of government revenue. In 2011, oil revenues surpassed SR1 trillion (US$275 billion) which constituted 92.5% of total government revenues (KSA 2013). In 2012, domestic consumption totaled 23.7% of oil production and if consumption continues to increase at 4% per year (as between 1990-2012), it will exceed the 2012 production level by 2049 (BP 2013).

Response

The government of Saudi Arabia has initiated several major programs in response to sustainability challenges. The Ninth Development Plan (KSA 2010) identifies high-level objectives such as accelerating economic growth, balancing regional development, raising the standard of living, diversifying and moving towards a knowledge-based economy, and ensuring rational utilization of natural resources. To support these objectives, the Ninth Development Plan seeks to appropriate SR1.4 trillion (US$370 billion) between 2010 and 2014, of which SR230 billion (US$61 billion) target economic resources development, an increase of 115% over the 8th Development Plan.

6.1.2 Response Challenges: Socio-technical Complexity

This application case focuses on investing in new supply-side infrastructure as a portion of the total sustainability response in Saudi Arabia.[1] This section discusses sources of complexity including technical structure and behavior and socio-political factors which challenge the decision-making process to identify the infrastructure projects in which to invest.

National infrastructure in Saudi Arabia spans sectors including water, electricity, petroleum, transportation, communications, and agriculture, each containing a network of interdependent elements. The internal structure of each element varies greatly from relatively simple (e.g. borehole water wells and pipelines) to complex (desalination and power plants). Each element also relies on input and output resource flows within (e.g. electricity distributed through power lines) and across sectors (e.g. electricity used for desalination). At the national level, however, structural complexity is dominated by the architectural contribution of distributed interdependencies between large numbers of infrastructure components leading to indirect and non-intuitive effects. For example, desalination may be perceived as an effective sustainable response to water demands, however a reliance on electricity generated from burning fossil fuels not only impacts the environmental state but also reduces fuel export revenue.

Infrastructure system behaviors take place on multiple timescales. Operational decisions on a minutes-to-months timescale optimize the cost efficiency of meeting demands. Feedback effects, non-linearities, and other uncertainties may be accommodated by analyzing observational data to build a prediction of future operations within a certain range of confidence. Longer-term strategic decisions on a years-to-decades timescale identify the set of infrastructure investments to meet sustainability objectives. Context changes and system evolution encompassing wide-ranging topics such as socio-economic consumption patterns, climate change, policy actions, and technology innovation are so numerous that extreme confidence bounds limit the usefulness of probabilistic predictions. One can assume the short-term effects are addressed in an optimal or nearly-optimal way to focus on larger impacts of long-term behaviors described in one or more plausible scenarios.

Socio-political complexity arises from the organizational or institutional space within which the physical systems exist. Rather than focusing on technical feasibility or economic viability, socio-political complexity deals with the desirability of options given goals, interests, and opinions of constituent stakeholders. The government in Saudi Arabia is a monarchy headed by the King who governs with help from the Crown Prince and the Council of Ministers representing 22 ministries, each specializing in a role of government. Today, the ministries associated with hard infrastructure planning include the Ministry of Agriculture (MoA), Ministry of Economy and Planning (MoEP), Ministry of Petroleum and Mineral Resources (MoPMR), and Ministry of Water and Electricity (MoWE). Furthermore, Saudi Arabia is regionally administered by 13 provinces, each having a governor and deputy governor. Thus, while a centrally-managed government, there are a large number of organizational units involved in detailed infrastructure planning, each with potentially-differing objectives.

Aspects of socio-political complexity are highlighted in several objectives of the Ninth Development Plan (KSA 2010). The third and fourth objectives describe national development plans to "achieve sustainable economic and social development" and "balanced development among regions of the Kingdom." The tenth and eleventh objectives discuss resource use and call for actions to "en-

[1] Demand-side efforts such as policies to reduce resource consumption are a crucial component of a complete sustainability response not explicitly considered here.

sure rational utilization of natural resources" and "develop regulations aimed at raising efficiency and improving performance." The objectives of (balanced) economic development and efficient resource use are linked by tensions created in past plans. The 1st Development Plan for 1970–1975 set a national policy to provide subsidies and grants to increase agricultural output (KSA 1970). The policy was greatly expanded throughout the 2nd Development Plan for 1975–1980 and maintained in following plans to improve the well-being of rural people and diversify the economy beyond petroleum products (KSA 1975). These policies directly improved economic development while contributing to balanced development across regions; however they also had a dramatic effect on water consumption, primarily non-renewable stocks of water. Influenced by implications of preliminary water studies, calls to reduce agricultural water consumption and institute "rationalization" of water consumption patterns first appeared the 5th Development Plan for 1990–1995 (KSA 1990) and gained importance as a key issue under the 7th Development Plan for 2000–2005 (KSA 2000). This plan took the first steps to reduce agricultural subsidy programs, leading to declines in water consumption in following years.

Only a few main sources of complexity are brought into focus to bound the scope of this application case. First, the architecture of infrastructure illustrating interdependencies within a distributed system-of-systems is selected as the dominant structural source of complexity. Second, long-term behaviors targeting sustainability objectives are emphasized as the basis for selecting and evolving infrastructure systems. Finally, socio-political factors are captured in the social layer to illustrate the different and potentially-conflicting objectives of actors involved in the planning process. While a simplified reflection of the real-world complexities, the emphasis on architecture, long-term behaviors, and socio-political objectives represent the core challenges in strategic infrastructure planning.

6.2 Simulation Game Formulation

The sustainable infrastructure planning simulation game (SIPS-G) supports strategic responses of infrastructure investment by expressing the socio-technical sources of complexity in Saudi Arabia. Its formulation relies on constructs of the infrastructure system-of-systems (ISoS) modeling framework presented in Chapter 4 to represent the national infrastructure planning problem at a level suitable for simulation and gaming. An instantiation of this framework identifies the context, structure, and behaviors of infrastructure systems in Saudi Arabia comprising the core game mechanics.

In framing SIPS-G, similar to other serious or purposeful games, one most balance reality, meaning, and play for effective design (Harteveld 2011). Reality identifies the link between details presented in the serious game and those of the real world, i.e. external validity. SIPS-G builds reality through contextual similarities in history, geography, and challenges relating to infrastructure planning in Saudi Arabia and leverages fundamental resource interdependencies between infrastructure elements. Meaning is the value-creating aspect of a serious game to achieve something useful in the real world. SIPS-G supports familiarizing players with complex infrastructure planning and accomplishing research objectives in studying collaborative design. Finally, play provides motivation for players through engagement, immersion, and fun. SIPS-G includes simplifications to improve its playability, allowing non-domain experts to participate and complete an integrated planning session in a few hours.

The objective of SIPS-G as a prototype lies at a descriptive level, rather than prescriptive

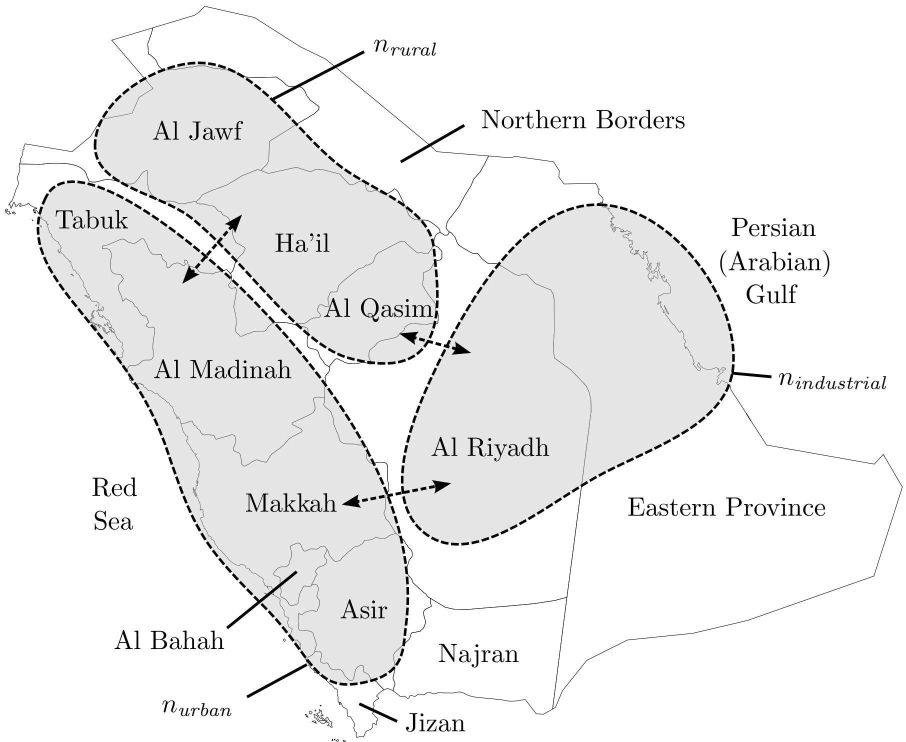

Figure 6-5: The SIPS-G context model aggregates thirteen provinces into three nodes in a fully-connected graph. These nodes represent the unique spatial contexts in industrial, urban, and rural regions. Map credit: d-maps.com, accessed at `http://d-maps.com/m/arabie/arabie20.pdf`.

or normative. Based on historical data, SIPS-G intends to highlight the architectural structure, long-term behaviors, and socio-political factors involved in infrastructure planning. Future work in improving the validation of component models may provide more substantive results capable of supporting active infrastructure planning activities.

6.2.1 Context Model

As mentioned previously, Saudi Arabia is administratively organized as 13 provinces illustrated in Figure 6-5. The Eastern Province includes the coastal area on the Persian (Arabian) Gulf with rich petroleum reserves and the large "Empty Quarter" desert area bordering Oman and the Emirates to the south. The central area, including most of the Al Riyadh province, is a desert plateau containing Riyadh, the largest and capital city. The west-coast provinces Makkah and Al Madinah border the Red Sea and contain the large and historic cities Jeddah, Mecca, and Medina. Finally, the northern provinces such as Al Jawf, Ha'il, and Al Qasim include more plentiful sources of water for agriculture.

The SIPS-G context aggregates the diverse geographic and administrative regions to three distinct nodes defined in Eq. 6.1.

$$\mathbf{N} = \{n_{industrial}, n_{rural}, n_{urban}\} \tag{6.1}$$

The "industrial" node, modeled after Riyadh and the Eastern Province, includes petroleum reservoirs as well as sea access for desalination, a large population with rapid growth, and limited arable land. The "rural" node, modeled after the northern provinces, includes higher quantities of arable land and a small population with moderate growth. Finally, the "urban" node, modeled after Jeddah and the western provinces, includes sea access for desalination, moderate arable land, and a large population with moderate growth.

Given the small graph size, allowable locations for infrastructure elements includes the complete set of node pairs defined in Eq. 6.2, resulting in a total of three nodal locations ($n_i = n_j$) and six edge locations ($n_i \neq n_j$).

$$\mathbf{L} = \{(n_i, n_j)\} \ \forall \ n_i, n_j \in \mathbf{N} \tag{6.2}$$

The set of allowable resource types defined in Eq. 6.3 includes basic resources types for food energy, water, oil, and electrical energy. The agriculture sector uses labor and land resource types to enforce constraints on available workforce and arable land. The water and petroleum sectors use aquifer water and reservoir oil as natural indirectly-accessible resource types which require infrastructure to transform to usable forms.[2] Finally, the social sector uses the people resource type for population and the currency resource type for financial resources.

$$\begin{aligned}\mathbf{T} = \big\{&\tau_{aquifer}, \tau_{currency}, \tau_{elect}, \tau_{food}, \tau_{labor}, \\ &\tau_{land}, \tau_{oil}, \tau_{people}, \tau_{reservoir}, \tau_{water}\big\}\end{aligned} \tag{6.3}$$

6.2.2 Structural Model

The SIPS-G structural model is an infrastructure system-of-systems consisting of a set of elements

$$\mathbf{E} = \{e_{ij}\} \tag{6.4}$$

where element e_{ij} is component j of system i identified by agricultural, water, petroleum, electrical, and social sectors. To simplify interfaces between systems, a system element aggregates the set of elements

$$\mathbf{E}_i^n = \{e_{ij}\} : n \in \mathcal{L}_o(e_{ij}) \tag{6.5}$$

for system i at node n where $\mathcal{L}_o$ gives the element's origin location. In other words, $\mathbf{E}_i^n$ is the set of elements in system i having a common origin node n. The complementary set of elements

$$\mathbf{E}_i^{n\star} = \{e_{ij}\} : n \notin \mathcal{L}_o(e_{ij}) \wedge n \in \mathcal{L}_d(e_{ij}) \tag{6.6}$$

aggregates elements having a common destination, excluding elements at nodal locations. For example, a distribution element $e \in \mathbf{E}_i$ between the industrial to urban nodes with location $\mathcal{L}(e) = (n_{industrial}, n_{urban})$ has origin $\mathcal{L}_o(e) = (n_{industrial}, n_{industrial})$ and destination $\mathcal{L}_d(e) = (n_{urban}, n_{urban})$ such that $e \in \mathbf{E}_i^{industrial}$ and $e \in \mathbf{E}_i^{urban\star}$.

Elements in the agricultural, water, petroleum, and electrical systems produce and distribute basic resources to the social system to meet demands. Infrastructure interdependencies arise from

[2] An alternative formulation avoiding aquifer and reservoir resources could define additional "underground" nodes where inaccessible water and oil stocks exist. Rather than transforming aquifer resources to water, infrastructure would transport water from "underground" to "surface" nodes.

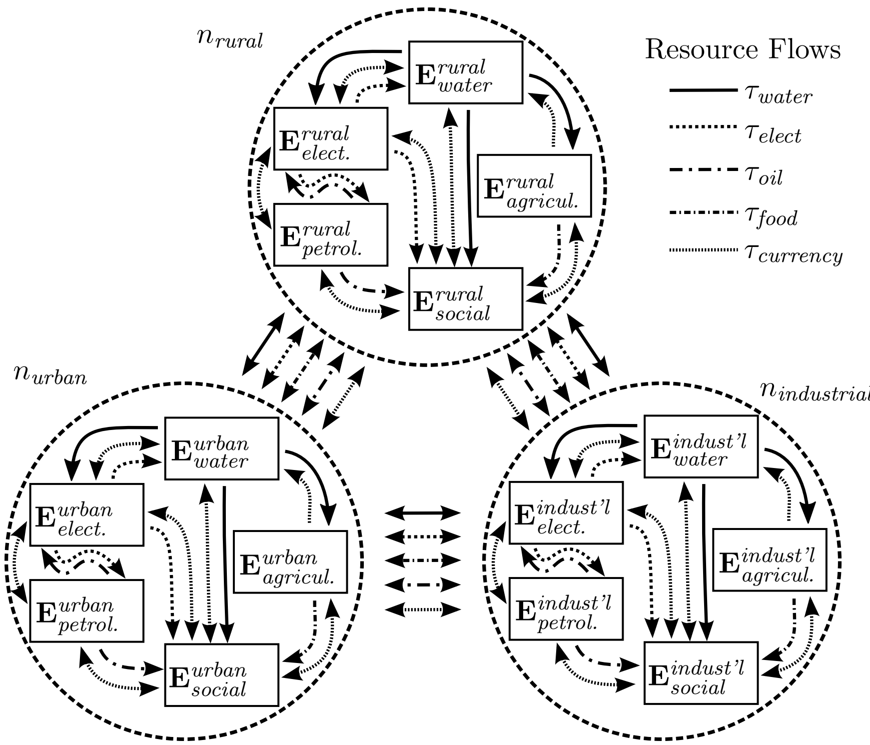

Figure 6-6: The SIPS-G system-of-systems model static dependency network illustrates resource flows between system models of agriculture, water, electricity, and petroleum sectors and the social system.

several basic relationships illustrated in the static dependency network in Figure 6-6. Some agricultural and electrical elements require water for irrigation and cooling respectively. Some water and petroleum production elements require electricity for pumping and other machinery. Thermal power plants require oil as fuel and may require water for cooling. Finally, the social system provides currency for operation of the infrastructure systems and societal demands include water, food, oil, and electricity. In addition to the system interdependencies at each node, spatial dependencies between nodes may also arise from inter-regional distribution.

6.3 Simulation Model Formulation

The simulation model builds on the selected context to identify a mathematical model to express structural and behavioral properties. While tailored to the SIPS-G application, the model is generalizable to other infrastructure systems as well. This section first introduces the social system model to capture non-infrastructure properties such as population and socio-economic resource demand. Next, a generic infrastructure system model represents common functionality between infrastructure sectors. Detailed models are then defined for each of the agriculture, water, petroleum, and electricity systems. Finally, a discussion addresses assumptions and limitations of the selected modeling approach.

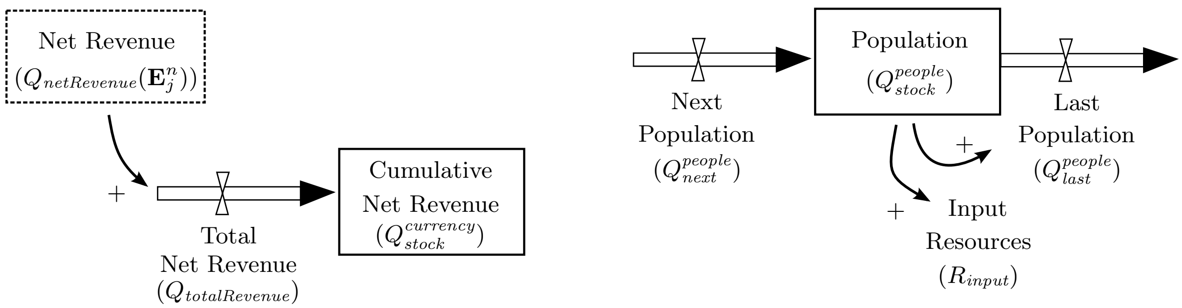

Figure 6-7: The social system system element illustrated as a causal loop diagram includes population and financial balance state variables. Dashed boxes indicate external variables and the dotted arrow indicates an causal loop conditional on demand model used.

Table 6.1: Social system system element behaviors and properties

Behavior	Functional Form and Dependent Properties	
Update Population	$\mathcal{R}_{transform}(\mathbf{E}_{social}^n, \{(\tau_{people}, Q_{last}^{people})\}, \{(\tau_{people}, Q_{next}^{people})\})$	(6.8)
	$\mathcal{R}_{store}(\mathbf{E}_{social}^n, \{(\tau_{people}, Q_{next}^{people})\}, \{(\tau_{people}, Q_{last}^{people})\})$	(6.9)
Add Revenue	$\mathcal{R}_{store}(\mathbf{E}_{social}^n, \{(\tau_{currency}, Q_{totalRevenue})\}, \{\})$	(6.10)
Satisfy Demands	$\mathcal{R}_{exchange}(\mathbf{E}_j^n, \mathbf{E}_{social}^n, \{(\tau_{food}, Q_{input}^{food}), (\tau_{water}, Q_{input}^{water}), (\tau_{oil}, Q_{input}^{oil}), (\tau_{elect}, Q_{input}^{elect})\})$	(6.11)
	$\mathcal{R}_{transform}(\mathbf{E}_{social}^n, \{(\tau_{food}, Q_{input}^{food}), (\tau_{water}, Q_{input}^{water}), (\tau_{oil}, Q_{input}^{oil}), (\tau_{elect}, Q_{input}^{elect})\}, \{\})$	(6.12)

6.3.1 Social System Model

The social system model captures all non-infrastructure activity such as population, cumulative revenue, and resource demands aggregating all residential, commercial, and industrial activity. It includes system elements at each node illustrated in Figure 6-7 as a causal loop diagram. The model stores population and currency stocks and receives net revenue from infrastructure systems. The total net revenue $Q_{totalRevenue}$ sums contributions from each infrastructure system j in Eq. 6.7.

$$Q_{totalRevenue} = \sum_j Q_{netRevenue}(\mathbf{E}_j^n) \tag{6.7}$$

Table 6.1 describes the system element behaviors. The update population behavior in Eq. 6.8–6.9 updates the stored population value from Q_{last}^{people} to Q_{next}^{people}. The add revenue behavior in Eq. 6.10 adds the aggregated revenue to the cumulative balance. Finally, the satisfy demands behavior in Eq. 6.11–6.12 exchanges resources from infrastructure system j to satisfy resource demands.

SIPS-G Population Model

The SIPS-G population model implements a logistic growth model of the form

$$Q_{next}^{people} = \frac{P_{max} \cdot P_0 \cdot e^{r_P \cdot (t-t_0)}}{P_{max} + P_0 \cdot \left(e^{r_P \cdot (t-t_0)} - 1\right)} \tag{6.13}$$

where P_{max} is the maximum population (i.e. carrying capacity), r is the population growth rate, and P_0 is the population at datum time t_0.

SIPS-G Resource Demand Model

The SIPS-G resource demand model implements a per-capita logistic growth model of the form

$$q_{input}^{\tau} = q_{min}^{\tau} + \frac{(q_{max}^{\tau} - q_{min}^{\tau}) \cdot (q_0^{\tau} - d_{min}^{\tau}) \cdot e^{r_\tau \cdot (t-t_0^\tau)}}{(q_{max}^{\tau} - q_{min}^{\tau}) + (q_0^{\tau} - d_{min}^{\tau}) \cdot \left(e^{r_\tau \cdot (t-t_0^\tau)} - 1\right)} \tag{6.14}$$

where q_{max}^{τ} is the maximum per-capita demand, q_{min}^{τ} is the minimum per-capita demand, r_τ is the per-capita demand growth rate, and q_0^{τ} is the demand at datum time t_0^{τ}. This formulation adds a lower bound on per-capita demand whereas the implicit lower bound on a usual logistic growth model for population is 0. When combined with the population model, the total demand for resource type τ is given by

$$Q_{input}^{\tau} = Q_{stock}^{people} \cdot q_{input}^{\tau}. \tag{6.15}$$

The resource demand model is limited to non-decreasing time-varying per-capita demands which are exogenous from other model properties. Future work develop demand models as a function of GDP or other socio-economic activity, resource price, or other actions such as policies, regulations, or other demand reduction programs.

6.3.2 Generic Infrastructure Model

The generic infrastructure model provides a common structure for agriculture, water, petroleum, and electricity systems. It aggregates the behavior of constituent elements to system behaviors and serves as a simplified interface for resource exchanging behaviors. This section introduces the generic infrastructure element model, its lifecycle model implementation in SIPS-G, the generic infrastructure system model, and its resource pricing implementations in SIPS-G.

Generic Infrastructure Element Model

The generic infrastructure element model defines the common structure of resource flows and related expenses for elements in agriculture, water, petroleum, and electricity sectors. Figure 6-8 illustrates potential relationships between element parameters. Variable expenses and input resources depend on resources sent and produced within element behaviors. Resources received are a function of those sent. Stored resources are added to a stock, from which retrieved resources are removed. Finally, other parameters such as capital, decommission, and fixed expense and resource import and export are separate model components.

Element behaviors parameterized in Table 6.2 define dependent variables with implementation-specific functional forms specified by lifecycle and operations models. The commission behavior

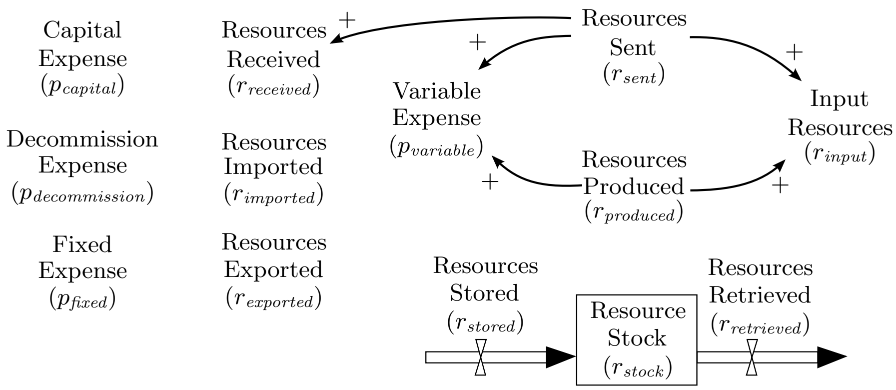

Figure 6-8: The generic infrastructure element model illustrated as a causal loop diagram is parameterized by resources produced, sent, stored, retrieved, imported, and exported.

Table 6.2: Generic infrastructure element model behaviors and properties

Behavior	Functional Form and Dependent Properties	
Commission	$\mathcal{E}_{transform}(e, s_c)$	(6.16)
	$\mathcal{R}_{transform}(e, \{\tau_{currency}, p_{capital}\}, \{\})$	(6.17)
Operate	$\mathcal{E}_{transform}(e, s_o)$	(6.18)
	$\mathcal{R}_{transform}(e, \{\tau_{currency}, p_{fixed}\}, \{\})$	(6.19)
Decommission	$\mathcal{E}_{transform}(e, s_d)$	(6.20)
	$\mathcal{R}_{transform}(e, \{\tau_{currency}, p_{decomm}\}, \{\})$	(6.21)
Produce Resources	$\mathcal{R}_{transform}(e, r_{input} \cup \{\tau_{currency}, p_{variable}\}, r_{produced})$	(6.22)
Distribute Resources	$\mathcal{R}_{transport}(e, r_{sent} \cup r_{input} \cup \{\tau_{currency}, p_{variable}\}, r_{received})$	(6.23)
Store Resources	$\mathcal{R}_{store}(e, r_{stored}, \{\})$	(6.24)
Retrieve Resources	$\mathcal{R}_{store}(e, \{\}, r_{retrieved})$	(6.25)
Import Resources	$\mathcal{R}_{transform}(e, \{\}, r_{imported})$	(6.26)
Export Resources	$\mathcal{R}_{transform}(e, r_{exported}, \{\})$	(6.27)

in Eq. 6.16–6.17 transforms to a commissioning state s_c and consumes capital expenses $p_{capital}$, the operate behavior in Eq. 6.18–6.19 transforms to an operational state s_o and consumes fixed operations expenses p_{fixed}, and the decommission behavior in Eq. 6.20–6.21 transforms to a decommissioning state s_d and consumes decommission expenses p_{decomm}. The produce behavior in Eq. 6.22 consumes input resources r_{input} and variable expenses $p_{variable}$ to produce resources $r_{produced}$. The distribute behavior in Eq. 6.23 sends resources r_{sent} (received as $r_{received}$) by consuming input resources and variable expenses. The store behavior in Eq. 6.24 stores resources r_{stored} and the retrieve behavior in Eq. 6.25 retrieves resources $r_{retrieved}$. Finally, the import and export behaviors in Eq. 6.26–6.27 transform imported and exported resources $r_{imported}$ and $r_{exported}$.

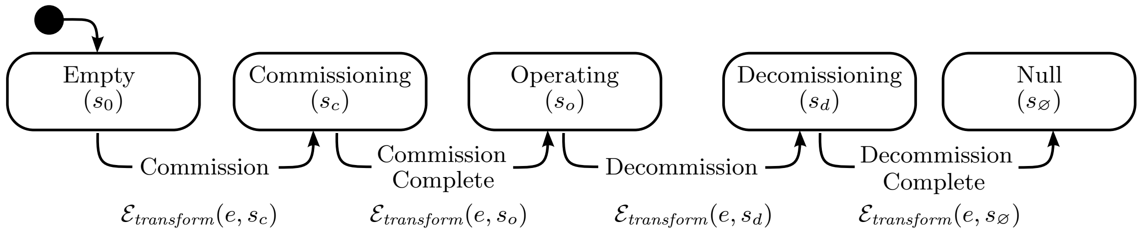

Figure 6-9: The state diagram for an element using the SIPS-G lifecycle model progresses between empty, commissioning, operating, decommissioning, and null states during its life-cycle.

Table 6.3: Element property values under the SIPS-G lifecycle model

State:	Commissioning (s_c)	Operating (s_o)	Decommissioning (s_d)
Time-varying Property	$t_0(e) \leq t < t_0(e) + d_0(e)$	$t_0(e) + d_0(e) \leq t < t_\varnothing$	$t_\varnothing(e) \leq t < t_\varnothing(e) + d_\varnothing(e)$
$p_{capital}(e, t, \Delta t)$	$p_{capital}(e)/d_0(e) \cdot \Delta t$	0	0
$p_{fixed}(e, t, \Delta t)$	0	$p_{fixed}(e) \cdot \Delta t$	0
$p_{variable}(e, t, \Delta t)$	0	$p_{variable}(e) \cdot \Delta t$	0
$q^{\tau,max}_{produced}(e, t, \Delta t)$	0	$q^{\tau,max}_{produced}(e) \cdot \Delta t$	0
$q^{\tau,max}_{sent}(e, t, \Delta t)$	0	$q^{\tau,max}_{sent}(e) \cdot \Delta t$	0
$p_{decomm}(e, t, \Delta t)$	0	0	$p_{decomm}(e)/d_\varnothing(e) \cdot \Delta t$

SIPS-G Element Lifecycle Model

The SIPS-G element lifecycle model consists of five distinct phases which control capital, operations, and decommission expenses and limits resource production and distribution to operational periods. The lifecycle model defines a set of five allowable states

$$S(e) = \{s_0, s_c, s_o, s_d, s_\varnothing\} \tag{6.28}$$

for element e. The states, illustrated in Figure 6-9, include commissioning (s_c), operating (s_o), and decommissioning (s_d) in addition to the empty state (s_0) before commissioning and the null state ($s_\varnothing$) after decommissioning.

The lifecycle model assigns commission time t_0, commission duration d_0, decommission time $t_\varnothing$ and decommission duration $d_\varnothing$ to control element state changes. Time-varying properties such as capital, fixed, variable, and decommission expenses, maximum production capacity, and maximum distribution capacity are evaluated at time t with time-step Δt using Table 6.3 to produce uniformly-distributed expenses over commissioning and decommissioning periods.

Generic Infrastructure System Model

The generic infrastructure system illustrated in Figure 6-10 aggregates the behavior of constituent element models and serves as a simplified interface for resource exchanging behaviors at each node. Upper-case variables (P, V, Q, R) identify aggregated system properties corresponding to element properties with lower-case variables (p, v, r, q). P is used for scalar expenses, V for scalar revenues, and R for vector resource sets with scalar component quantities Q having matching subscripts.

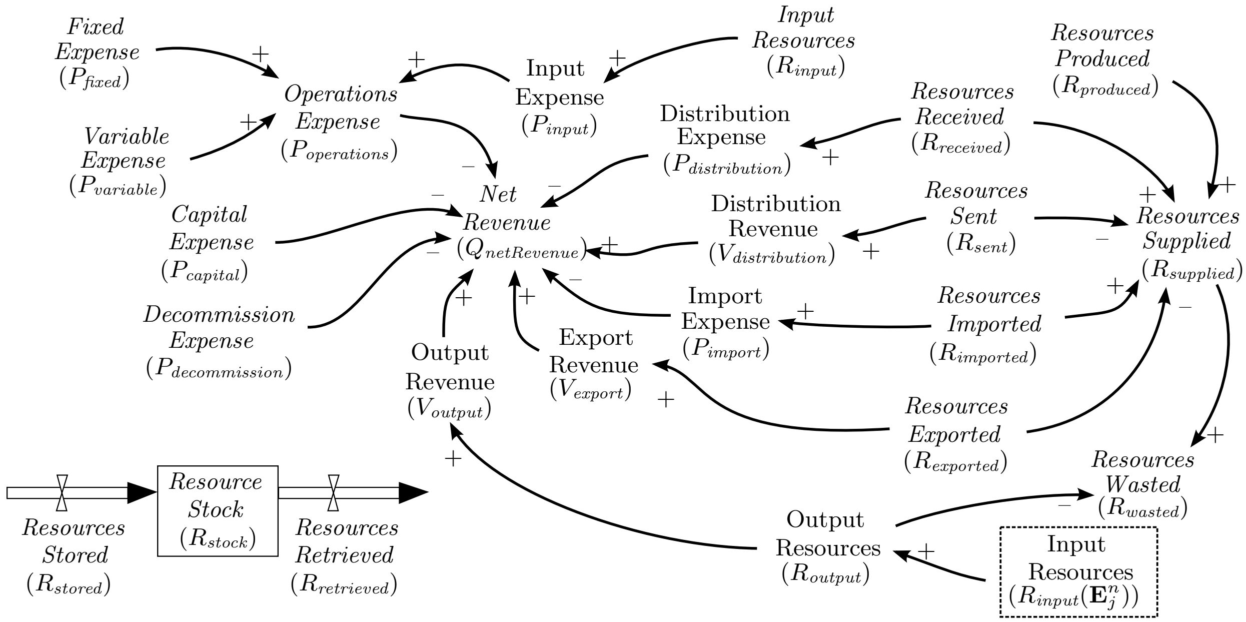

Figure 6-10: The generic infrastructure system model illustrated as a causal loop diagram aggregates element models to italicized variables and external system models to dashed-box variables. Input and output resources are computed for exchange with other infrastructure systems.

For example, $R_{sample} = \{(\tau_{food}, Q^{food}_{sample}), (\tau_{water}, Q^{water}_{sample})\}$ is the "sample" set of food and water resources.

Aggregated properties with italicized labels in Figure 6-10 are directly computed from constituent element models and include scalar and vector types defined in Table 6.4. Scalar properties express quantities related to life-cycle expenses. Those defined in Eq. 6.29–6.32 include capital expenses ($P_{capital}$) to commission infrastructure, fixed (P_{fixed}) and variable ($P_{variable}$) expenses of operating infrastructure, and decommission (P_{decomm}) expenses. The general form aggregates elements having a common origin such that the only one location incurs expenses for edge-located elements such as pipelines. Vector properties defined in Eq. 6.33–6.42 include resources stored (R_{stored}) and retrieved ($R_{retrieved}$) from a stock (R_{stock}), input resources to production and distribution processes (R_{input}), resources produced ($R_{produced}$), resources received ($R_{received}$) and sent (R_{sent}) via distribution, and resources imported ($R_{imported}$), exported ($R_{exported}$), and output (R_{output}) to other systems. Most properties aggregate elements having a common origin with two exceptions. First, resources received in Eq. 6.38 aggregates elements with a common destination for resource distribution. Second, output resources in Eq. 6.42 aggregates the input resources property of other systems to determine output requirements.

Other aggregated properties combine existing system properties. For example, the operations expense ($P_{operations}$) in Eq. 6.43 sums fixed, variable, and resource input expenses. The set of wasted resources (R_{wasted}) in Eq. 6.44 is the difference between resources supplied and those output. The set of supplied resources ($R_{supplied}$) in Eq. 6.45 is the set of resources produced, received and imported less those sent and exported. Finally, the net revenue ($Q_{netRevenue}$) in Eq. 6.46 is the difference between revenues and expenses.

Dependent properties defined in Table 6.5 parameterize infrastructure system behaviors and

Table 6.4: Generic infrastructure system model aggregated properties

Aggregated Property		Aggregated Property	
$P_{capital} = \sum_{e \in \mathbf{E}_i^n} p_{capital}(e)$	(6.29)	$P_{fixed} = \sum_{e \in \mathbf{E}_i^n} p_{fixed}(e)$	(6.30)
$P_{variable} = \sum_{e \in \mathbf{E}_i^n} p_{variable}(e)$	(6.31)	$P_{decomm} = \sum_{e \in \mathbf{E}_i^n} p_{decomm}(e)$	(6.32)
$R_{stored} = \bigcup_{e \in \mathbf{E}_i^n} r_{stored}(e)$	(6.33)	$R_{stock} = \bigcup_{e \in \mathbf{E}_i^n} r_{stock}(e)$	(6.34)
$R_{retrieved} = \bigcup_{e \in \mathbf{E}_i^n} r_{retrieved}(e)$	(6.35)	$R_{input} = \bigcup_{e \in \mathbf{E}_i^n} r_{input}(e)$	(6.36)
$R_{produced} = \bigcup_{e \in \mathbf{E}_i^n} r_{produced}(e)$	(6.37)	$R_{received} = \bigcup_{e \in \mathbf{E}_i^{n\star}} r_{received}(e)$	(6.38)
$R_{sent} = \bigcup_{e \in \mathbf{E}_i^n} r_{sent}(e)$	(6.39)	$R_{imported} = \bigcup_{e \in \mathbf{E}_i^n} r_{imported}(e)$	(6.40)
$R_{exported} = \bigcup_{e \in \mathbf{E}_i^n} r_{exported}(e)$	(6.41)	$R_{output} = \bigcup_j R_{input}(\mathbf{E}_j^n)$	(6.42)
$P_{operations} = P_{fixed} + P_{variable} + P_{input}$	(6.43)	$R_{wasted} = R_{supplied} - R_{output}$	(6.44)
$Q_{netRevenue} = V_{output} + V_{export} + V_{distribution}$ $- P_{capital} - P_{operations} - P_{decomm}$ $- P_{distribution} - P_{import}$	(6.46)	$R_{supplied} = R_{produced} \cup R_{received} \cup R_{imported}$ $- R_{sent} - R_{exported}$	(6.45)

Table 6.5: Generic infrastructure system model behaviors and properties

Behavior	Functional Form and Dependent Properties	
Exchange Inputs	$\mathcal{R}_{exchange}(\mathbf{E}_j^n, \mathbf{E}_i^n, R_{input})$	(6.47)
	$\mathcal{R}_{transform}(\mathbf{E}_i^n, \{(\tau_{currency}, P_{input})\}, \{\})$	(6.48)
Exchange Outputs	$\mathcal{R}_{exchange}(\mathbf{E}_i^n, \mathbf{E}_j^n, R_{output})$	(6.49)
	$\mathcal{R}_{transform}(\mathbf{E}_i^n, \{\}, \{(\tau_{currency}, V_{output})\})$	(6.50)
Exchange Revenue	$\mathcal{R}_{exchange}(\mathbf{E}_i^n, \mathbf{E}_{social}^n, \{(\tau_{currency}, Q_{netRevenue})\})$	(6.51)
Waste Resources	$\mathcal{R}_{transform}(\mathbf{E}_i^n, R_{wasted}, \{\})$	(6.52)

have an implementation-specific functional form specified by a resource pricing model. The exchange inputs behavior defined in Eq. 6.47–6.48 exchanges resources R_{input} from system j to i and consumes expenses P_{input} as a function of input resources. The exchange outputs behavior defined in Eq. 6.49–6.50 exchanges resources from system i to j and produces revenue V_{output} as a function of output resources. Both cases use two behaviors: one to exchange resources and one to transform currency for expenses or revenue. An alternative implementation could combine the two to exchange both resources and currency in a single behavior. The exchange revenue behavior defined in Eq. 6.51 exchanges currency $Q_{netRevenue}$ to the social system. The waste resources behavior defined in Eq. 6.52 consumes unused resources r_{wasted}.

SIPS-G Resource Pricing Model

The SIPS-G resource pricing model calculates revenues and expenses associated with resource flows with a linear pricing model which uses constant unit resource prices. Local prices for resource type τ are set with parameter π^τ_{local}, import prices with π^τ_{import}, and export prices with π^τ_{export}.

Using the linear pricing model, Eq. 6.53 computes input resource expenses and Eq. 6.54 computes distribution expenses. Equation 6.55 computes import expenses using the import price.

Similarly, Eq. 6.56 computes output revenue and Eq. 6.57 computes distribution revenue using the local price. Equation 6.58 computes export revenue using the export price.

$$P_{input} = \sum_{(\tau, Q^{\tau}_{input}) \in R_{input}} \pi^{\tau}_{local} \cdot Q^{\tau}_{input} \tag{6.53}$$

$$P_{distribution} = \sum_{(\tau, Q^{\tau}_{received}) \in R_{received}} \pi^{\tau}_{local} \cdot Q^{\tau}_{received} \tag{6.54}$$

$$P_{import} = \sum_{(\tau, Q^{\tau}_{imported}) \in R_{imported}} \pi^{\tau}_{import} \cdot Q^{\tau}_{imported} \tag{6.55}$$

$$V_{output} = \sum_{(\tau, Q^{\tau}_{output}) \in R_{output}} \pi^{\tau}_{local} \cdot Q^{\tau}_{output} \tag{6.56}$$

$$V_{distribution} = \sum_{(\tau, Q^{\tau}_{sent}) \in R_{sent}} \pi^{\tau}_{local} \cdot Q^{\tau}_{sent} \tag{6.57}$$

$$V_{export} = \sum_{(\tau, Q^{\tau}_{exported}) \in R_{exported}} \pi^{\tau}_{export} \cdot Q^{\tau}_{exported} \tag{6.58}$$

The pricing model as implemented is limited by its linear (constant unit price) assumption. It does not accurately reflect economic elasticity of supply versus demand. Furthermore, there is no effect of currency valuation changes (e.g. inflation, deflation). These limitations may be addressed in future work with a more complex pricing model considering equilibrium between supply and demand within each infrastructure system and changes in currency valuation over time.

6.3.3 Agriculture System Model

The agriculture system includes system elements (type A_S), production elements (type A_P), and distribution elements (type A_D) with behaviors and properties described in Table 6.6.

The agriculture system element (type A_S) defines behaviors for updating labor resources and interacting with a global food market. The stock of arable land is initialized with a fixed maximum area q^{land}_{stock} usable for any agricultural purpose. The update labor behaviors in Eq. 6.59–6.60 modify the stock of labor usable for agricultural production. The export/import behaviors in Eq. 6.61–6.62 allow transactions with the global food market.

Agriculture production elements (type A_P) include behaviors for producing food. They directly inherit commission, operate, and decommission behaviors from the generic infrastructure element in Eq. 6.63–6.68. The produce food behavior in Eq. 6.69 consumes water and variable expenses to produce food.

Agriculture distribution elements (type A_D) include behaviors for distributing food. They directly inherit commission, operate, and decommission behaviors from the generic infrastructure element in Eq. 6.70–6.75. The distribute food behavior in Eq. 6.76 consumes variable expenses to transport food.

This formulation does not include food storage elements which may be a topic of future work to represent strategic food reserves.

Table 6.6: Agriculture system elements, behaviors, and properties

Type	Behavior	Functional Form and Dependent Properties	
A_S	Update Labor	$\mathcal{R}_{transform}(\mathbf{E}^n_{agricul.}, \{\}, \{(\tau_{labor}, q^{labor}_{produced})\})$	(6.59)
		$\mathcal{R}_{store}(\mathbf{E}^n_{agricul.}, \{(\tau_{labor}, q^{labor}_{stored})\}, \{\})$	(6.60)
	Export Food	$\mathcal{R}_{transform}(\mathbf{E}^n_{agricul.}, \{(\tau_{food}, q^{food}_{exported})\}, \{\})$	(6.61)
	Import Food	$\mathcal{R}_{transform}(\mathbf{E}^n_{agricul.}, \{\}, \{(\tau_{food}, q^{food}_{imported})\})$	(6.62)
A_P	Commission	$\mathcal{E}_{transform}(e, s_c)$	(6.63)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{capital})\}, \{\})$	(6.64)
	Operate	$\mathcal{E}_{transform}(e, s_o)$	(6.65)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{fixed})\}, \{\})$	(6.66)
	Decommission	$\mathcal{E}_{transform}(e, s_d)$	(6.67)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{decomm})\}, \{\})$	(6.68)
	Produce Food	$\mathcal{R}_{transform}(e, \{(\tau_{water}, q^{water}_{input}), (\tau_{currency}, p_{variable})\}, \{(\tau_{food}, q^{food}_{produced})\})$	(6.69)
A_D	Commission	$\mathcal{E}_{transform}(e, s_c)$	(6.70)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{capital})\}, \{\})$	(6.71)
	Operate	$\mathcal{E}_{transform}(e, s_o)$	(6.72)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{fixed})\}, \{\})$	(6.73)
	Decommission	$\mathcal{E}_{transform}(e, s_d)$	(6.74)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{decomm})\}, \{\})$	(6.75)
	Distribute Food	$\mathcal{R}_{transport}(e, \{(\tau_{food}, q^{food}_{sent}), (\tau_{currency}, p_{variable})\}, \{(\tau_{food}, q^{food}_{received})\})$	(6.76)

SIPS-G Agriculture System Element Operations Model

The SIPS-G agriculture system element (type A_S) operations model defines labor production as a fraction of the population growth in Eq. 6.77. The parameter $f_{people}^{labor} \subseteq [0, 1]$ represents the agricultural labor participation rate in the agriculture system.

$$q_{produced}^{labor} = f_{people}^{labor} \cdot \left(Q_{next}^{population}(\mathbf{E}_{social}^{n}) - Q_{last}^{population}(\mathbf{E}_{social}^{n}) \right) \tag{6.77}$$

The agriculture system element assumes the maximum stock of arable land area q_{stock}^{land} remains constant over time and is usable for any agricultural production. Future work may model several types of land to represent varying climates or irrigation requirements.

SIPS-G Agriculture Production Element Operations Model

The SIPS-G agriculture production element (type A_P) operations model defines the functional form of commissioning and food production. A production element includes parameters for land and labor input in Eq. 6.78 where the parameter f_{land}^{labor} defines the labor required to work a unit of land.

$$q_{input}^{labor} = f_{land}^{labor} \cdot q_{input}^{land} \tag{6.78}$$

Food production consumes water, generates variable expenses, and produces food as a function of the input land utilized. Equation 6.79 defines the water input as a linear function of land area with coefficient f_{land}^{water} specifying the water requirements per unit area. Equation 6.80 defines variable expenses as a linear function of land area with coefficient $f_{land}^{currency}$ specifying the cost per unit area. Finally, Eq. 6.81 defines food production as a linear function of land area with coefficient f_{land}^{food} specifying the yield.

$$q_{input}^{water} = f_{land}^{water} \cdot q_{input}^{land} \tag{6.79}$$

$$p_{variable} = f_{land}^{currency} \cdot q_{input}^{land} \tag{6.80}$$

$$q_{produced}^{food} = f_{land}^{food} \cdot q_{input}^{land} \tag{6.81}$$

The agriculture production element is limited by the linear operations assumption as a function of land area used. Future work may develop functional forms for labor, water, variable costs, and production as a function of climate, geography, or level of mechanization.

SIPS-G Agriculture Distribution Element Operations Model

The SIPS-G agriculture distribution element operations model defines the functional form of food distribution. Equation 6.82 defines variable expenses as a linear function of sent food with coefficient $f_{food}^{currency}$ as the cost per unit sent. Equation 6.83 defines food received as a fraction of food sent with efficiency $\eta \subseteq [0, 1]$.

$$p_{variable} = f_{food}^{currency} \cdot q_{sent}^{food} \tag{6.82}$$

$$q_{received}^{food} = \eta \cdot q_{sent}^{food} \tag{6.83}$$

The agriculture distribution element is limited by the linear operations assumption as a function of food sent. Future work may develop functional forms for variable costs and food received—for example, as a function of distance between nodes.

SIPS-G Agriculture System Flow Optimization

Food production and distribution within the SIPS-G agricultural system implementation can be optimized to minimize cost using the following linear programming (LP) problem formulation. The design vector in Eq. 6.84 includes food production and distribution variables for each infrastructure element as well as import and export variables for each system element. The cost function in Eq. 6.85 sums operational expenses for infrastructure elements, import expenses, and export revenue ($P_{operations} + P_{import} - V_{export}$). The constraints in Eq. 6.86–6.87 restrict element production and distribution below maximum values. The constraints in Eq. 6.88–6.89 restrict aggregated element production to available land and labor at each node. Finally, the constraint in Eq. 6.90 ensures the net food supplied ($Q^{food}_{supplied}$) meets demands at each node.

$$\text{find: } \begin{array}{l} q^{land}_{input}(e), q^{food}_{sent}(e)\ \forall\ e \in \mathbf{E}_{agricul.}; \\ q^{food}_{imported}(\mathbf{E}^n_{agricul.}), q^{food}_{exported}(\mathbf{E}^n_{agricul.})\ \forall\ n \in \mathbf{N} \end{array} \tag{6.84}$$

$$\text{to minimize: } \begin{aligned} &\sum_{e \in \mathbf{E}_{agricul.}} \left(f^{currency}_{land}(e) + f^{water}_{land}(e) \cdot \pi^{water}_{local}\right) \cdot q^{land}_{input}(e) \\ &+ \sum_{e \in \mathbf{E}_{agricul.}} f^{currency}_{food}(e) \cdot q^{food}_{sent}(e) \\ &+ \sum_{n \in \mathbf{N}} \pi^{food}_{import} \cdot q^{food}_{imported}(\mathbf{E}^n_{agricul.}) - \pi^{food}_{export} \cdot q^{food}_{exported}(\mathbf{E}^n_{agricul.}) \end{aligned} \tag{6.85}$$

$$\text{subject to: } q^{land}_{input}(e) \leq q^{land}_{input,max}(e)\ \forall\ e \in \mathbf{E}_{agricul.} \tag{6.86}$$

$$q^{food}_{sent}(e) \leq q^{food}_{sent,max}(e)\ \forall\ e \in \mathbf{E}_{agricul.} \tag{6.87}$$

$$\sum_{e \in \mathbf{E}^n_{agricul.}} q^{land}_{input}(e) \leq q^{land}_{stock}(\mathbf{E}^n_{agricul.})\ \forall\ n \in \mathbf{N} \tag{6.88}$$

$$\sum_{e \in \mathbf{E}^n_{agricul.}} f^{labor}_{land}(e) \cdot q^{land}_{input}(e) \leq q^{labor}_{stock}(\mathbf{E}^n_{agricul.})\ \forall\ n \in \mathbf{N} \tag{6.89}$$

$$\begin{aligned} &\sum_{e \in \mathbf{E}^n_{agricul.}} \left(f^{food}_{land}(e) \cdot q^{land}_{input}(e) - q^{food}_{sent}(e)\right) + \sum_{e \in \mathbf{E}^{n\star}_{agricul.}} \left(\eta(e) \cdot q^{food}_{sent}(e)\right) \\ &+ q^{food}_{imported}(\mathbf{E}^n_{agricul.}) - q^{food}_{exported}(\mathbf{E}^n_{agricul.}) \geq Q^{food}_{output}(\mathbf{E}^n_{agricul.})\ \forall\ n \in \mathbf{N} \end{aligned} \tag{6.90}$$

Assuming fixed prices where $\pi^{food}_{export} < \pi^{food}_{local} < \pi^{food}_{import}$, food import and export quantities are given by Eq. 6.91–6.92, however optimization is still required to determine element-level production and distribution values.

$$q^{food}_{imported}(\mathbf{E}^n_{agricul.}) = \max\left(0, Q^{food}_{output}(\mathbf{E}^n_{agricul.}) - Q^{food}_{supplied}(\mathbf{E}^n_{agricul.})\right)\ \forall\ n \in \mathbf{N} \tag{6.91}$$

Table 6.7: Water system elements, behaviors, and properties

Type	Behavior	Functional Form and Dependent Properties	
W_S	Aquifer Withdraw	$\mathcal{R}_{store}(\mathbf{E}^n_{water}, \{\}, \{(\tau_{aquifer}, q^{aquifer}_{retrieved}\})$	(6.93)
	Aquifer Recharge	$\mathcal{R}_{transform}(\mathbf{E}^n_{water}, \{\}, \{(\tau_{aquifer}, q^{aquifer}_{recharged})\})$	(6.94)
		$\mathcal{R}_{store}(\mathbf{E}^n_{water}, \{(\tau_{aquifer}, q^{aquifer}_{recharged})\}, \{\})$	(6.95)
	Import Water	$\mathcal{R}_{transform}(\mathbf{E}^n_{water}, \{\}, \{(\tau_{water}, q^{water}_{imported})\})$	(6.96)
	Produce Water	$\mathcal{R}_{transform}(\mathbf{E}^n_{water}, \{(\tau_{elect}, q^{elect}_{input}), (\tau_{aquifer}, q^{aquifer}_{input})\},$ $\{(\tau_{water}, q^{water}_{produced})\})$	(6.97)
W_P	Commission	$\mathcal{E}_{transform}(e, s_c)$	(6.98)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{capital})\}, \{\})$	(6.99)
	Operate	$\mathcal{E}_{transform}(e, s_o)$	(6.100)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{fixed})\}, \{\})$	(6.101)
	Decommission	$\mathcal{E}_{transform}(e, s_d)$	(6.102)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{decomm})\}, \{\})$	(6.103)
	Produce Water	$\mathcal{R}_{transform}(e, \{(\tau_{elect}, q^{elect}_{input}), (\tau_{aquifer}, q^{aquifer}_{input}),$ $(\tau_{currency}, p_{variable})\}, \{(\tau_{water}, q^{water}_{produced})\})$	(6.104)
W_D	Commission	$\mathcal{E}_{transform}(e, s_c)$	(6.105)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{capital})\}, \{\})$	(6.106)
	Operate	$\mathcal{E}_{transform}(e, s_o)$	(6.107)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{fixed})\}, \{\})$	(6.108)
	Decommission	$\mathcal{E}_{transform}(e, s_d)$	(6.109)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{decomm})\}, \{\})$	(6.110)
	Distribute Water	$\mathcal{R}_{transport}(e, \{(\tau_{water}, q^{water}_{sent}), (\tau_{elect}, q^{elect}_{input}),$ $(\tau_{currency}, p_{variable})\}, \{(\tau_{water}, q^{water}_{received})\})$	(6.111)

$$q^{food}_{exported}(\mathbf{E}^n_{agricul.}) = \max\left(0, Q^{food}_{supplied}(\mathbf{E}^n_{agricul.}) - Q^{food}_{output}(\mathbf{E}^n_{agricul.})\right) \;\forall\, n \in \mathbf{N} \tag{6.92}$$

6.3.4 Water System Model

The water system includes system elements (type W_S), production elements (type W_P), and distribution elements (type W_D) with behaviors and properties described in Table 6.7.

The water system element (type W_S) defines behaviors for withdrawing and recharging aquifer resources, importing water, and producing water from unmanaged infrastructure. The aquifer withdrawal behavior in Eq. 6.93 retrieves aquifer resources. The aquifer recharge behavior in Eq. 6.94–6.95 restores a portion of aquifer resources. The import water behavior in Eq. 6.96 acquires water from the global market. Finally, the produce water behavior in Eq. 6.97 produces water from private infrastructure by consuming electricity and aquifer resources. Private production differs from other water production by avoiding direct expenses, however it is only used to meet shortfalls in supply.

Water production elements (type W_P) include behaviors for producing water. They directly inherit commission, operate, and decommission behaviors from the generic infrastructure element in Eq. 6.98–6.103. The produce water behavior in Eq. 6.104 consumes variable expenses and electricity and aquifer resources to produce water.

Water distribution elements (type W_D) include behaviors for distributing water. They directly inherit commission, operate, and decommission behaviors from the generic infrastructure element in Eq. 6.105–6.110. The distribute water behavior in Eq. 6.111 consumes variable expenses and electricity resources to transport water.

This formulation does not include surface-level water storage elements such as reservoirs which may be a topic of future work to represent strategic water resources.

SIPS-G Water System Element Operations Model

The SIPS-G water system element operations model defines the functional form for aquifer recharging and private water production. Aquifer recharge takes place at a constant recharge rate $r_{recharge}$ as defined in Eq. 6.112 to avoid exceeding aquifer capacity $q_{max}^{aquifer}$. Equations 6.113–6.114 define electricity and aquifer input resources as a linear function of water production with coefficients f_{water}^{elect} and $f_{water}^{aquifer}$ as specific electricity and aquifer resource consumption per unit produced (nominally $f_{water}^{aquifer} = 1.0$ for private production sourced from aquifers).

$$q_{recharged}^{aquifer} = \begin{cases} r_{recharge} & \text{if } q_{stock}^{aquifer} < q_{max}^{aquifer} - r_{recharge} \\ q_{max}^{aquifer} - r_{recharge} & \text{if } q_{max}^{aquifer} - r_{recharge} \leq q_{stock}^{aquifer} < q_{max}^{aquifer} \\ 0 & \text{otherwise} \end{cases} \tag{6.112}$$

$$q_{input}^{elect} = f_{water}^{elect} \cdot q_{produced}^{water} \tag{6.113}$$

$$q_{input}^{aquifer} = f_{water}^{aquifer} \cdot q_{produced}^{water} \tag{6.114}$$

The water system element assumes all aquifer resources are accessible through a single stock, recharge at a constant rate, and are usable for any water production. It is also limited in its linear assumption of private production electricity consumption as a function of water produced. Future work may develop electricity consumption as a function of aquifer depth (itself a function of aquifer volume).

SIPS-G Water Production Element Operations Model

The SIPS-G water production element operations model defines the functional form for water production. Equations 6.115–6.117 define electricity and aquifer input resources and variable expenses as a linear function of water production with coefficients f_{water}^{elect} and $f_{water}^{aquifer}$ as specific electricity and aquifer resource consumption per unit produced and $f_{water}^{currency}$ as the cost per unit produced.

$$q_{input}^{elect} = f_{water}^{elect} \cdot q_{produced}^{water} \tag{6.115}$$

$$q_{input}^{aquifer} = f_{water}^{aquifer} \cdot q_{produced}^{water} \tag{6.116}$$

$$p_{variable} = f_{water}^{currency} \cdot q_{produced}^{water} \tag{6.117}$$

The water production element is limited by the linear operations assumption as a function of water produced. Future work may develop functional forms for electricity and variable costs as a function of aquifer depth or technological advancement.

SIPS-G Water Distribution Element Operations Model

The SIPS-G water distribution element operations model defines the functional form for water transport. Equations 6.118–6.119 define electricity input resources and variable expenses as a linear function of water distribution with coefficient f_{water}^{elect} as specific electricity resource consumption per unit sent and $f_{water}^{currency}$ as the cost per unit sent. Equation 6.120 defines water received as a fraction of water sent with efficiency $\eta \subseteq [0, 1]$.

$$q_{input}^{elect} = f_{water}^{elect} \cdot q_{sent}^{water} \tag{6.118}$$

$$p_{variable} = f_{water}^{currency} \cdot q_{sent}^{water} \tag{6.119}$$

$$q_{received}^{water} = \eta \cdot q_{sent}^{water} \tag{6.120}$$

The water distribution element is limited by the linear operations assumption as a function of water sent. Future work may develop functional forms for electricity and variable costs as a function of distance or elevation change between nodes and pipe diameter.

SIPS-G Water System Flow Optimization

Water production and distribution within the SIPS-G water system implementation can be optimized to minimize cost using the following linear programming (LP) problem formulation. The design vector in Eq. 6.121 includes water production and distribution variables for each infrastructure element as well as private production and import variables for each system element. The cost function in Eq. 6.122 includes operational expenses for infrastructure elements and import expenses ($P_{operations} + P_{import}$). The M factor bounded in Eq. 6.123 sets water production from elements to be preferred over private production and private production to be preferred over import. The constraints in Eq. 6.124–6.125 restrict element production and distribution below maximum values and those in Eq. 6.126 restrict aggregated element production to available aquifer quantities at each node. Finally, the constraint in Eq. 6.127 ensures the net water supplied ($Q_{supplied}^{water}$) meets demands at each node.

$$\text{find: } \begin{array}{l} q^{water}_{produced}(e), q^{water}_{sent}(e) \ \forall\, e \in \mathbf{E}_{water}; \\ q^{water}_{produced}(\mathbf{E}^n_{water}), q^{water}_{imported}(\mathbf{E}^n_{water}) \ \forall\, n \in \mathbf{N} \end{array} \tag{6.121}$$

$$\begin{aligned} \text{to minimize: } & \sum_{e \in \mathbf{E}_{water}} \left(f^{currency}_{water}(e) + f^{elect}_{water}(e) \cdot \pi^{elect}_{local} \right) \cdot q^{water}_{produced}(e) \\ & + \sum_{e \in \mathbf{E}_{water}} \left(f^{currency}_{water}(e) + f^{elect}_{water}(e) \cdot \pi^{elect}_{local} \right) \cdot q^{water}_{sent}(e) \\ & + \sum_{n \in \mathbf{N}} M \cdot q^{water}_{produced}(\mathbf{E}^n_{water}) + \sum_{n \in \mathbf{N}} \pi^{water}_{import} \cdot q^{water}_{imported}(\mathbf{E}^n_{water}) \end{aligned} \tag{6.122}$$

$$\text{where } \max_{e \in \mathbf{E}_{water}} \left(f^{currency}_{water}(e) + f^{elect}_{water}(e) \cdot \pi^{elect}_{local} \right) < M < \pi^{water}_{import} \tag{6.123}$$

$$\text{subject to: } q^{water}_{produced}(e) \le q^{water}_{produced,max}(e) \ \forall\, e \in \mathbf{E}_{water} \tag{6.124}$$

$$q^{water}_{sent}(e) \le q^{water}_{sent,max}(e) \ \forall\, e \in \mathbf{E}_{water} \tag{6.125}$$

$$\begin{aligned} & \sum_{e \in \mathbf{E}^n_{water}} \left(f^{aquifer}_{water}(e) \cdot q^{water}_{produced}(e) \right) + f^{aquifer}_{water}(\mathbf{E}^n_{water}) \cdot q^{water}_{produced}(\mathbf{E}^n_{water}) \\ & \quad \le Q^{aquifer}_{stock}(\mathbf{E}^n_{water}) \ \forall\, n \in \mathbf{N} \end{aligned} \tag{6.126}$$

$$\begin{aligned} & \sum_{e \in \mathbf{E}^n_{water}} \left(q^{water}_{produced}(e) - q^{water}_{sent}(e) \right) + \sum_{e \in \mathbf{E}^{n\star}_{water}} \left(\eta(e) \cdot q^{water}_{sent}(e) \right) \\ & \quad + q^{water}_{produced}(\mathbf{E}^n_{water}) + q^{water}_{imported}(\mathbf{E}^n_{water}) \ge Q^{water}_{output}(\mathbf{E}^n_{water}) \ \forall\, n \in \mathbf{N} \end{aligned} \tag{6.127}$$

Assuming fixed prices where $\pi^{water}_{local} < \pi^{water}_{import}$ and preference for element-level production, private water production and import quantities are given by Eq. 6.128–6.129, however optimization is still required to determine element-level production and distribution values.

$$\begin{aligned} q^{water}_{produced}(\mathbf{E}^n_{water}) = \min \Big(& Q^{water}_{output}(\mathbf{E}^n_{water}) - Q^{water}_{supplied}(\mathbf{E}^n_{water}), \\ & Q^{aquifer}_{stock}(\mathbf{E}^n_{water}) / f^{aquifer}_{water}(\mathbf{E}^n_{water}) \Big) \ \forall\, n \in \mathbf{N} \end{aligned} \tag{6.128}$$

$$q^{water}_{imported}(\mathbf{E}^n_{water}) = Q^{water}_{output}(\mathbf{E}^n_{water}) - Q^{water}_{supplied}(\mathbf{E}^n_{water}) - q^{water}_{produced}(\mathbf{E}^n_{water}) \ \forall\, n \in \mathbf{N} \tag{6.129}$$

6.3.5 Petroleum System Model

The petroleum system includes system elements (type P_S), production elements (type P_P), and distribution elements (type P_D) with behaviors and properties described in Table 6.8.

The petroleum system element (type P_S) defines behaviors for extracting from reservoirs, and interacting with the global oil market. The extract oil behavior in Eq. 6.130 withdraws reservoir resources. The export/import behaviors in Eq. 6.131–6.132 allow transactions with the global oil market.

Petroleum production elements (type P_P) include behaviors for producing oil. They directly inherit commission, operate, and decommission behaviors from the generic infrastructure element in Eq. 6.133–6.138. The produce oil behavior in Eq. 6.139 consumes variable expenses and reservoir resources to produce oil.

Table 6.8: Petroleum system elements, behaviors, and properties

Type	Behavior	Functional Form and Dependent Properties	
P_S	Extract Oil	$\mathcal{R}_{store}(\mathbf{E}^n_{petrol.}, \{\}, \{(\tau_{reservoir}, q^{reservoir}_{retrieved})\})$	(6.130)
	Export Oil	$\mathcal{R}_{transform}(\mathbf{E}^n_{petrol.}, \{(\tau_{oil}, q^{oil}_{exported})\}, \{\})$	(6.131)
	Import Oil	$\mathcal{R}_{transform}(\mathbf{E}^n_{petrol.}, \{\}, \{(\tau_{oil}, q^{oil}_{imported})\})$	(6.132)
P_P	Commission	$\mathcal{E}_{transform}(e, s_c)$	(6.133)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{capital})\}, \{\})$	(6.134)
	Operate	$\mathcal{E}_{transform}(e, s_o)$	(6.135)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{fixed})\}, \{\})$	(6.136)
	Decommission	$\mathcal{E}_{transform}(e, s_d)$	(6.137)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{decomm})\}, \{\})$	(6.138)
	Produce Oil	$\mathcal{R}_{transform}(e, \{(\tau_{reservoir}, q^{reservoir}_{input}), (\tau_{currency}, p_{variable})\},$ $\{(\tau_{oil}, q^{oil}_{produced})\})$	(6.139)
P_D	Commission	$\mathcal{E}_{transform}(e, s_c)$	(6.140)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{capital})\}, \{\})$	(6.141)
	Operate	$\mathcal{E}_{transform}(e, s_o)$	(6.142)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{fixed})\}, \{\})$	(6.143)
	Decommission	$\mathcal{E}_{transform}(e, s_d)$	(6.144)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{decomm})\}, \{\})$	(6.145)
	Distribute Oil	$\mathcal{R}_{transport}(e, \{(\tau_{oil}, q^{oil}_{sent}), (\tau_{elect}, q^{elect}_{input}), (\tau_{currency}, p_{variable})\},$ $\{(\tau_{oil}, q^{oil}_{received})\})$	(6.146)

Petroleum distribution elements (type P_D) include behaviors for transporting oil. They directly inherit commission, operate, and decommission behaviors from the generic infrastructure element in Eq. 6.140–6.145. The distribute oil behavior in Eq. 6.146 consumes electricity and variable expenses to transport oil.

This formulation does not include surface-level oil storage elements which may be a topic of future work to represent strategic oil resources.

SIPS-G Petroleum System Element Operations Model

The SIPS-G petroleum system element does not modify any resource stocks or allow private production. It assumes all reservoir resources are accessible through a single stock and are usable for any oil production. Future work may develop reservoir models with uncertainty in proven reserves and multiple stocks of varying quality or ease of access.

SIPS-G Petroleum Production Element Operations Model

The SIPS-G petroleum production element operations model defines the functional form for oil production. Equations 6.147–6.148 define reservoir input resources and variable expenses as a linear function of oil production with coefficients $f_{oil}^{reservoir}$ as specific reservoir resource consumption (nominally 1.0 for reservoir extraction) and $f_{oil}^{currency}$ as the cost per unit produced.

$$q_{input}^{reservoir} = f_{oil}^{reservoir} \cdot q_{produced}^{oil} \tag{6.147}$$

$$p_{variable} = f_{oil}^{currency} \cdot q_{produced}^{oil} \tag{6.148}$$

The petroleum production element is limited by the linear operations assumption as a function of oil produced. Future work may develop functional forms for variable costs as a function of reservoir depth, quality, or technological advancement.

SIPS-G Petroleum Distribution Element Operations Model

The SIPS-G petroleum distribution element operations model defines the functional form for oil transport. Equations 6.149–6.150 define electricity input resources and variable expenses as a linear function of oil distribution with coefficient f_{oil}^{elect} as specific electricity resource consumption per unit sent and $f_{oil}^{currency}$ as the cost per unit sent. Equation 6.151 defines oil received as a fraction of oil sent with efficiency $\eta \subseteq [0, 1]$.

$$q_{input}^{elect} = f_{oil}^{elect} \cdot q_{sent}^{oil} \tag{6.149}$$

$$p_{variable} = f_{oil}^{currency} \cdot q_{sent}^{oil} \tag{6.150}$$

$$q_{received}^{oil} = \eta \cdot q_{sent}^{oil} \tag{6.151}$$

The petroleum distribution element is limited by the linear operations assumption as a function of oil sent. Future work may develop functional forms for electricity and variable costs as a function of distance or elevation change between nodes and pipe diameter.

SIPS-G Petroleum System Flow Optimization

Oil production and distribution within the SIPS-G petroleum system implementation can be optimized to minimize cost using the following linear programming (LP) problem formulation. The design vector in Eq. 6.152 includes oil production and distribution variables for each infrastructure element as well as import and export variables for each system element. The cost function in Eq. 6.153 includes operational expenses for infrastructure elements, import expenses, and export revenue ($P_{operations} + P_{import} - V_{export}$). The constraints in Eq. 6.154–6.155 restrict element production and distribution below maximum values. The constraints in Eq. 6.156 restricts aggregated element production to available reservoir quantities. Finally, the constraint in Eq. 6.157 ensures the net oil supplied ($Q^{oil}_{supplied}$) meets demands.

$$\text{find: } \begin{array}{l} q^{oil}_{produced}(e), q^{oil}_{sent}(e) \ \forall\ e \in \mathbf{E}_{petrol.}; \\ q^{oil}_{exported}(\mathbf{E}^n_{petrol.}), q^{oil}_{imported}(\mathbf{E}^n_{petrol.}) \ \forall\ n \in \mathbf{N} \end{array} \tag{6.152}$$

$$\text{to minimize: } \begin{aligned} &\sum_{e \in \mathbf{E}_{petrol.}} f^{currency}_{oil}(e) \cdot q^{oil}_{produced}(e) \\ &+ \sum_{e \in \mathbf{E}_{petrol.}} \left(f^{currency}_{oil}(e) + f^{elect}_{oil}(e) \cdot \pi^{elect}_{local} \right) \cdot q^{oil}_{sent}(e) \\ &+ \sum_{n \in \mathbf{N}} \pi^{oil}_{import} \cdot q^{oil}_{imported}(\mathbf{E}^n_{petrol.}) - \pi^{oil}_{export} \cdot q^{oil}_{exported}(\mathbf{E}^n_{petrol.}) \end{aligned} \tag{6.153}$$

$$\text{subject to: } q^{oil}_{produced}(e) \le q^{oil}_{produced,max}(e) \ \forall\ e \in \mathbf{E}_{petrol.} \tag{6.154}$$

$$q^{oil}_{sent}(e) \le q^{oil}_{sent,max}(e) \ \forall\ e \in \mathbf{E}_{petrol.} \tag{6.155}$$

$$\sum_{e \in \mathbf{E}^n_{petrol.}} \left(f^{reservoir}_{oil}(e) \cdot q^{oil}_{produced}(e) \right) \le q^{reservoir}_{stock}(\mathbf{E}^n_{petrol.}) \ \forall\ n \in \mathbf{N} \tag{6.156}$$

$$\begin{aligned} &\sum_{e \in \mathbf{E}^n_{petrol.}} \left(q^{oil}_{produced}(e) - q^{oil}_{sent}(e) \right) + \sum_{e \in \mathbf{E}^{n\star}_{petrol.}} \left(\eta(e) \cdot q^{oil}_{sent}(e) \right) \\ &+ q^{oil}_{imported}(\mathbf{E}^n_{petrol.}) - q^{oil}_{exported}(\mathbf{E}^n_{petrol.}) \ge Q^{oil}_{output}(\mathbf{E}^n_{petrol.}) \ \forall\ n \in \mathbf{N} \end{aligned} \tag{6.157}$$

Assuming fixed prices where $\pi^{oil}_{export} < \pi^{oil}_{local} < \pi^{oil}_{import}$, oil import and export quantities are given by Eq. 6.158–6.159, however optimization is still required to determine element-level production and distribution values.

$$q^{oil}_{imported}(\mathbf{E}^n_{petrol.}) = \max\left(0, Q^{oil}_{output}(\mathbf{E}^n_{petrol.}) - Q^{oil}_{supplied}(\mathbf{E}^n_{petrol.})\right) \ \forall\ n \in \mathbf{N} \tag{6.158}$$

$$q^{oil}_{exported}(\mathbf{E}^n_{petrol.}) = \max\left(0, Q^{oil}_{supplied}(\mathbf{E}^n_{petrol.}) - Q^{oil}_{output}(\mathbf{E}^n_{petrol.})\right) \ \forall\ n \in \mathbf{N} \tag{6.159}$$

6.3.6 Electricity System Model

The electricity system includes system elements (type E_S), production elements (type E_P), and distribution elements (type E_D) with behaviors and properties described in Table 6.9.

The electricity system element (type E_S) defines behaviors producing electricity using private

Table 6.9: Electrical system elements, behaviors, and properties

Type	Behavior	Functional Form and Dependent Properties	
E_S	Produce Electricity	$\mathcal{R}_{transform}(\mathbf{E}^n_{elect.}, \{(\tau_{oil}, q^{oil}_{input})\}, \{(\tau_{elect}, q^{elect}_{produced})\})$	(6.160)
E_P	Commission	$\mathcal{E}_{transform}(e, s_c)$	(6.161)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{capital})\}, \{\})$	(6.162)
	Operate	$\mathcal{E}_{transform}(e, s_o)$	(6.163)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{fixed})\}, \{\})$	(6.164)
	Decommission	$\mathcal{E}_{transform}(e, s_d)$	(6.165)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{decomm})\}, \{\})$	(6.166)
	Produce Electricity	$\mathcal{R}_{transform}(e, \{(\tau_{water}, q^{water}_{input}), (\tau_{oil}, q^{oil}_{input}), (\tau_{currency}, p_{variable})\},$ $\{(\tau_{elect}, q^{elect}_{produced})\})$	(6.167)
E_D	Commission	$\mathcal{E}_{transform}(e, s_c)$	(6.168)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{capital})\}, \{\})$	(6.169)
	Operate	$\mathcal{E}_{transform}(e, s_o)$	(6.170)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{fixed})\}, \{\})$	(6.171)
	Decommission	$\mathcal{E}_{transform}(e, s_d)$	(6.172)
		$\mathcal{R}_{transform}(e, \{(\tau_{currency}, p_{decomm})\}, \{\})$	(6.173)
	Distribute Electricity	$\mathcal{R}_{transport}(e, \{(\tau_{elect}, q^{elect}_{sent}), (\tau_{currency}, p_{variable})\},$ $\{(\tau_{elect}, q^{elect}_{received})\})$	(6.174)

infrastructure. The produce electricity behavior in Eq. 6.160 consumes oil to generate electricity. Private production differs from other electricity production by avoiding direct expenses, however it is only used to meet shortfalls in public supply.

Electricity production elements (type E_P) include behaviors for producing electricity. They directly inherit commission, operate, and decommission behaviors from the generic infrastructure element in Eq. 6.161–6.166. The produce electricity behavior in Eq. 6.167 consumes water and oil input resources and variable expenses to produce electricity.

Electricity distribution elements (type E_D) include behaviors for transporting electricity. They directly inherit commission, operate, and decommission behaviors from the generic infrastructure element in Eq. 6.168–6.173. The distribute electricity behavior in Eq. 6.174 consumes variable expenses to transport electricity.

SIPS-G Electricity System Element Operations Model

The SIPS-G electricity system element operations model defines the functional form for private electricity production. Equation 6.175 defines oil input resources as a linear function of electricity production with coefficient f^{oil}_{elect} as the specific oil consumption per unit produced.

$$q^{oil}_{input} = f^{oil}_{elect} \cdot q^{elect}_{produced} \tag{6.175}$$

The electricity system element is limited in its linear assumption of private production oil consumption as a function of electricity produced. Future work may develop private production models as a function of technological advancement.

SIPS-G Electricity Production Element Operations Model

The SIPS-G electricity production element operations model defines the functional form for electricity production. Equations 6.176–6.178 define oil and water input resources and variable expenses as a linear function of electricity production with coefficients f_{elect}^{water} and f_{elect}^{oil} as specific water and oil resource consumption and $f_{elect}^{currency}$ as the cost per unit produced.

$$q_{input}^{water} = f_{elect}^{water} \cdot q_{produced}^{elect} \tag{6.176}$$

$$q_{input}^{oil} = f_{elect}^{oil} \cdot q_{produced}^{elect} \tag{6.177}$$

$$p_{variable} = f_{elect}^{currency} \cdot q_{produced}^{elect} \tag{6.178}$$

The electricity production element is limited by the linear operations assumption as a function of electricity produced. Future work may develop functional forms for variable costs and oil and water resource consumption to incorporate nonlinearities and as a function of technological advancement.

SIPS-G Electricity Distribution Element Operations Model

The SIPS-G electricity distribution element operations model defines the functional form for electricity transport. Equation 6.179 defines variable expenses as a linear function of electricity distribution with coefficient $f_{elect}^{currency}$ as the cost per unit sent. Equation 6.180 defines electricity received as a fraction of electricity sent with efficiency $\eta \subseteq [0, 1]$.

$$p_{variable} = f_{elect}^{currency} \cdot q_{sent}^{elect} \tag{6.179}$$

$$q_{received}^{elect} = \eta \cdot q_{sent}^{elect} \tag{6.180}$$

The electricity distribution element is limited by the linear operations assumption as a function of electricity sent. Future work may develop functional forms for resistive losses as a function of distance between nodes and voltage level.

SIPS-G Electricity System Flow Optimization

Electricity production and distribution within the SIPS-G electricity system implementation can be optimized to minimize cost using the following linear programming (LP) problem formulation. The design vector in Eq. 6.181 includes electricity production and distribution variables for each infrastructure element as well as a private production variable for each system element. The cost function in Eq. 6.182 includes operational expenses ($P_{operations}$). The M factor bounded in Eq. 6.183 makes electricity production from elements always preferred over private production. The constraints in Eq. 6.184–6.185 restrict element production and distribution below maximum values. Finally, the constraint in Eq. 6.186 ensures the net electricity supplied ($Q_{supplied}^{elect.}$) meets demands.

$$\text{find: } \begin{array}{l} q^{elect.}_{produced}(e), q^{elect.}_{sent}(e)\ \forall\ e \in \mathbf{E}_{petrol.}; \\ q^{elect.}_{produced}(\mathbf{E}^n_{petrol.})\ \forall\ n \in \mathbf{N} \end{array} \tag{6.181}$$

$$\text{to minimize: } \begin{array}{l} \sum_{e \in \mathbf{E}_{petrol.}} \left(f^{currency}_{elect.}(e) + f^{oil}_{elect.}(e) \cdot \pi^{oil}_{local} + f^{water}_{elect.}(e) \cdot \pi^{water}_{local} \right) \cdot q^{elect.}_{produced}(e) \\ \quad + \sum_{e \in \mathbf{E}_{elect.}} f^{currency}_{elect.}(e) \cdot q^{elect.}_{sent}(e) + \sum_{n \in \mathbf{N}} M \cdot q^{elect.}_{produced}(\mathbf{E}^n_{petrol.}) \end{array} \tag{6.182}$$

$$\text{where } M > \max_{e \in \mathbf{E}_{elect.}} \left(f^{currency}_{elect}(e) + f^{oil}_{elect.}(e) \cdot \pi^{oil}_{local} + f^{water}_{elect.}(e) \cdot \pi^{water}_{local} \right) \tag{6.183}$$

$$\text{subject to: } q^{elect.}_{produced}(e) \leq q^{elect.}_{produced,max}(e)\ \forall\ e \in \mathbf{E}_{elect.} \tag{6.184}$$

$$q^{elect.}_{sent}(e) \leq q^{elect.}_{sent,max}(e)\ \forall\ e \in \mathbf{E}_{elect.} \tag{6.185}$$

$$\begin{array}{l} \sum_{e \in \mathbf{E}^n_{elect.}} \left(q^{elect.}_{produced}(e) - q^{elect.}_{sent}(e) \right) + \sum_{e \in \mathbf{E}^{n\star}_{elect.}} \left(\eta(e) \cdot q^{elect.}_{sent}(e) \right) \\ \quad + q^{elect.}_{produced}(\mathbf{E}^n_{elect.}) \geq Q^{elect.}_{output}(\mathbf{E}^n_{elect.})\ \forall\ n \in \mathbf{N} \end{array} \tag{6.186}$$

Assuming element-level production is preferred, private electricity production is given by Eq. 6.187, however optimization is still required to determine element-level production and distribution values.

$$q^{elect.}_{produced}(\mathbf{E}^n_{elect.}) = Q^{elect.}_{output}(\mathbf{E}^n_{elect.}) - Q^{elect.}_{supplied}(\mathbf{E}^n_{elect.})\ \forall\ n \in \mathbf{N} \tag{6.187}$$

6.3.7 Assumptions and Limitations

The simulation model formulation presented in this section is balance of the triadic game design elements of reality, meaning, and play. While it contains aspects of the real-world infrastructure, there are a number of assumptions breaking from reality to improve meaning or play. In particular, as this application emphasizes the architectural structure and long-term behavior of infrastructure systems, there is minimal representation of internal model details (e.g. of an infrastructure element), interaction between models (e.g. of resource exchange between systems), and short-term operational behaviors (e.g. at time-scales smaller than one year).

First, in terms of the overall framing, all infrastructure systems and elements are managed by players as representatives of the associated government ministries. Although a reasonable assumption for the water and energy sectors, this is not particularly accurate of the agriculture sector where much of the production and distribution is performed by the private sector. Still, the government role in agriculture has historically been one of policy-making to encourage or discourage private sector actions which can be seen as managing infrastructure once removed.

This formulation also assumes there is a sole source of each resource type. This assumption reduces the amount of information required for exchange between systems. For example, if both water and electricity systems could produce electricity, some combination of the two would satisfy demands. However, the systems must coordinate to prevent shortfalls or wasted resources. Omitting this detail is not a limitation of the ISoS framework, but rather is intentional simplification for the prototype simulation game. Multiple sources of a resource type would require additional operational logic at each time step and would increase the number of iteration cycles required within each time advancement due to information dependencies. Furthermore, the model assumes there is never a shortfall in resources supplied to meet required outputs. Each infrastructure system contains a mechanism for unbounded resource supply: the agriculture system can import food, the water system can import water, the petroleum system can import oil, and the electrical system can use private generators for electricity.

Although the underlying mechanics cover certain economic activities such as sales and trade, there are no realistic economic mechanisms in the present formulation. All resource prices, both domestic and international, are held constant without elasticities to supply or demand. This simplification is intentional to reduce cognitive load and ease implementation, however future extensions may introduce local price elasticities and geographical variances and import/export price variance to improve realism. Variable resource prices would contribute additional interdependencies between systems to reach economic equilibrium conditions.

Other nested models within the social system such as resource demand models and population growth models are also highly stylized. Several of the demand models are only a function of time and are unaffected by the player actions. Similarly, the logistic population growth model in this formulation is exogenous and is in no way affected by surplus or shortage of key resources important for quality of life. More complex models could improve realism and also introduce important feedbacks relating growth and prosperity. However, one must also consider the relative magnitude of infrastructure as a small portion of the entire national economy.

Finally, the infrastructure element models defined are also simplified to express linear behaviors as a function or production or distribution level. The numerous life-cycle model attributes including capital cost, fixed operations cost, and decommission cost help to create a nonlinear overall life-cycle cost profile while maintaining linearized components. All other attributes including resource consumption intensities of production or distribution (e.g. water, electricity, and variable operations cost) and other resource use intensities (e.g. water aquifer, petroleum reservoir, arable land, labor) are linear in the production or distribution value. This is not an inherent limitation in the modeling framework, but rather a conscious simplification to both reduce cognitive load and to enable linear programming for automated micro-managing of infrastructure operations. While future applications of this modeling framework could add more complex models, the effect of linear behavior models is somewhat mitigated by aggregating multiple models across several systems, allowing compositions of linear element behaviors to produce non-linear system results.

6.4 Baseline Scenario Model Instantiations

This section introduces a set of model instantiations which specify values for the parameterized properties of infrastructure systems and elements. The instantiations are selected for triadic design of play, reality, and meaning in the SIPS-G application. Some parameters are based on historical estimates of infrastructure operations in Saudi Arabia while others are purposefully selected to improve game play. In general, parameters dealing with physical resource transformations tend to reflect reality. Where appropriate, citations to referenced data sources help to identify validated parameters. Some concepts represented in the SIPS-G application to improve meaning do not directly coincide with publicly-available data, requiring interpretation and approximation. Parameters dealing with finances tend to be more sophistic to support balanced play. In particular, all currency-based values are reported a fictitious currency of simoleons represented with the § symbol.

Table 6.10: Population model parameters

Variable	Description	$n_{industrial}$	n_{urban}	n_{rural}	Units
t_0	Datum time	1980	1980	1980	year
r	Growth rate of population	7	6	5	%
P_0	Datum population	3.0	6.0	0.75	million people
P_{max}	Maximum population	17.5	20.0	4.0	million people

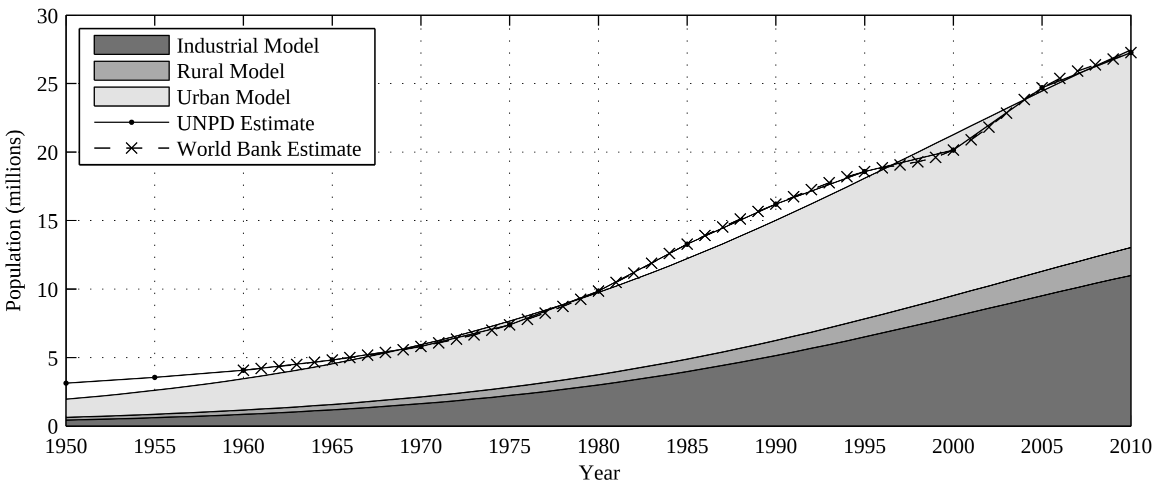

Figure 6-11: Population model validation. A stacked area chart compares regional model values to population estimates between 1950 and 2010. Data sources: UNPD 2013; World Bank 2012.

6.4.1 Social System Model Instantiation

The social system model instantiation provides parameter values for the population model and food, water, and electricity demand models.

Population Model Parameters

Table 6.10 describes parameters for the SIPS-G logistic growth population model implementation at each node. Parameter values approximate highest growth rates in the industrial node, highest initial population in the urban node, and lowest initial population and growth in the rural node. Figure 6-11 compares aggregate results from the three nodal population models with historical estimates with good fit. The disaggregation of population among the three nodes, however, is balanced for game play.

Food Demand Model Parameters

The food demand model assumes a single food resource type aggregating commonly-used classifications of food, animal feed, and seed. Figure 6-12 illustrates estimates of direct food consumption compared to total food energy supply (food, feed, and seed). The gap after 1975 is driven by expansion of animal sources of food which nearly doubles the total food supply required. Table 6.11 describes parameter values for the food demand model using the SIPS-G per-capita logistic growth

Table 6.11: Food demand model parameters

Variable	Description	$n_{industrial}$	n_{urban}	n_{rural}	Units
t_0^{food}	Datum time	1975	1975	1975	year
r_{food}	Growth rate of per-capita demand	20	20	20	%
d_{min}^{food}	Minimum per-capita demand	1800	1800	1800	kcal/day
d_0^{food}	Datum per-capita demand	2300	2300	2300	kcal/day
d_{max}^{food}	Maximum per-capita demand	5800	5800	5800	kcal/day

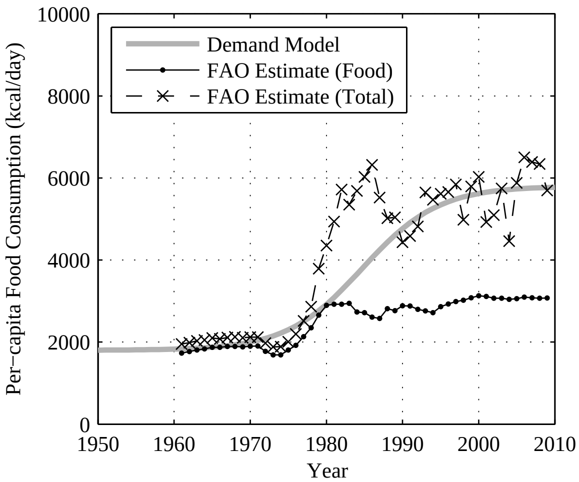

Figure 6-12: Food demand model outputs compared to historical estimates of per-capita food consumption and total food supply (including feed and seed) between 1950 and 2010. Data source: FAO 2013b (total food consumption, total food supply).

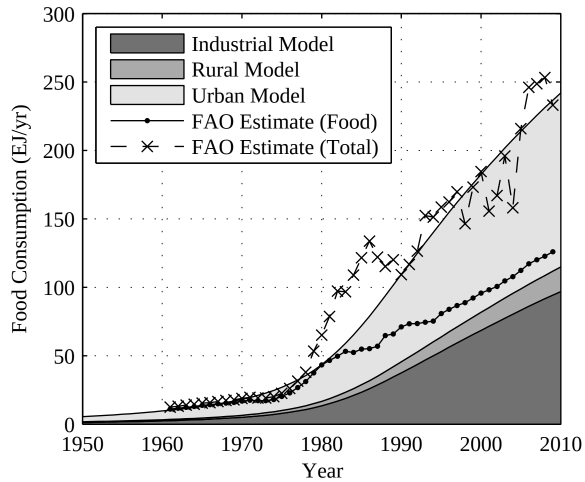

Figure 6-13: A stacked area chart compares aggregated demand model outputs with historical estimates of food consumption and total food supply (including feed and seed) between 1950 and 2010. Data source: FAO 2013b (total food consumption, total food supply).

model implementation with equal values at each node. It uses a lower bound of 1800 kilo-calories per day (kcal/day) and a maximum of 5800 kcal/day with a growth rate of 20%. A datum demand of 2300 kcal/day is fixed in 1975.

Figures 6-12–6-13 compare per-capita and aggregate food consumption estimates with resulting demand model values between 1950-2010. While the model does not capture the rapid increases observed between 1975 and 1980, it follows the general trend of per-capita demand increasing by nearly 4000 kcal/day between 1950 and 2010.

Water Demand Model Parameters

Table 6.12 describes parameter values for the water demand model using the SIPS-G per-capita logistic growth model with equal values at each node. It uses a lower bound of 25 liters per day (L/day) and a maximum of 325 L/day with a 8% annual growth rate. A datum demand of 175 L/day is fixed in 1965. Figures 6-14–6-15 compare per-capita and aggregate water consumption estimates and demand model values between 1950-2010. Despite limited data points, the demand

Table 6.12: Water demand model parameters

Variable	Description	$n_{industrial}$	n_{urban}	n_{rural}	Units
t_0^{water}	Datum time	1965	1965	1965	year
r_{water}	Growth rate of per-capita demand	8	8	8	%
d_{min}^{water}	Minimum per-capita demand	25	25	25	L/day
d_0^{water}	Datum per-capita demand	175	175	175	L/day
d_{max}^{water}	Maximum per-capita demand	325	325	325	L/day

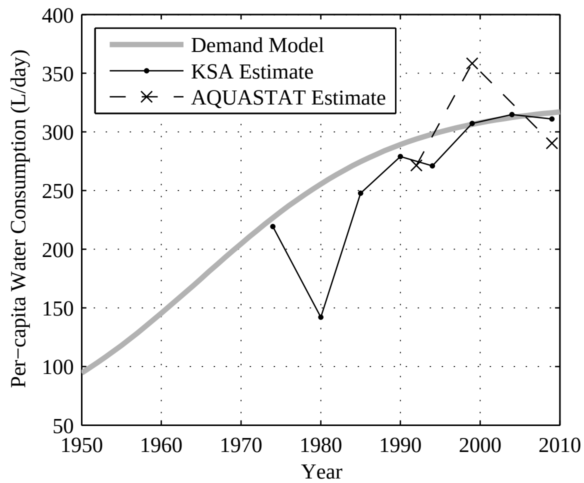

Figure 6-14: Water demand model outputs compared to historical estimates of per-capita water consumption between 1950 and 2010. Data sources: KSA 1990; KSA 1995; KSA 2000; KSA 2005; KSA 2010 (municipal plus industrial withdrawals) FAO 2013a (municipal plus industrial withdrawals).

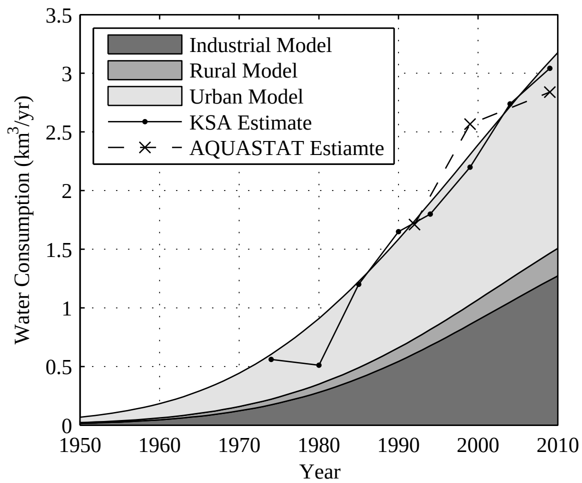

Figure 6-15: A stacked area chart compares aggregated demand model outputs with historical estimates of water consumption between 1950 and 2010. Data sources: KSA 1990; KSA 1995; KSA 2000; KSA 2005; KSA 2010 (municipal plus industrial withdrawals) FAO 2013a (municipal plus industrial withdrawals).

model illustrates a tripling of per-capita water demand between 1950 and 2010.

Oil Demand Model Parameters

Table 6.13 describes parameter values for the oil demand model using the SIPS-G per-capital logistic growth model with equal values at each node. It uses a lower bound of 0 tonnes of oil equivalent per year (toe/yr) and a maximum of 7 toe/yr with a 7% annual growth rate. A datum demand of 1 toe/yr is fixed in 1970. Figures 6-16–6-17 compare per-capita and aggregate oil consumption estimates and demand model values between 1950-2010.

Electricity Demand Model Parameters

Table 6.14 describes parameter values for the electricity demand model using the SIPS-G per-capita logistic growth model with equal values at each node. It uses a lower bound of 0 kilowatt hours

Table 6.13: Oil demand model parameters

Variable	Description	$n_{industrial}$	n_{urban}	n_{rural}	Units
t_0^{oil}	Datum time	1970	1970	1970	year
r_{oil}	Growth rate of per-capita demand	7	7	7	%
d_{min}^{oil}	Minimum per-capita demand	0	0	0	toe/yr
d_0^{elect}	Datum per-capita demand	1	1	1	toe/yr
d_{max}^{oil}	Maximum per-capita demand	9	9	9	toe/yr

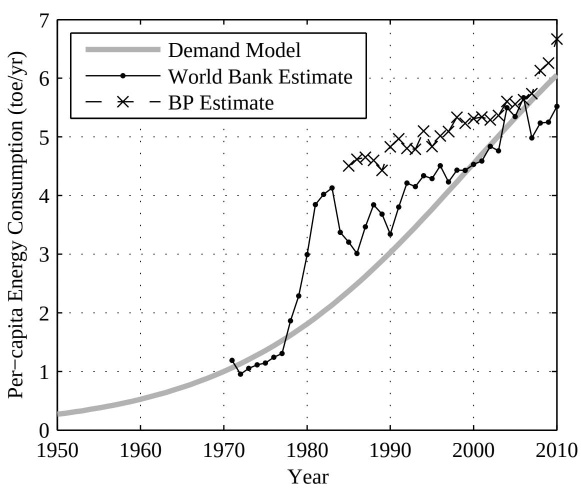

Figure 6-16: Oil demand model outputs compared to historical estimates of per-capita energy consumption between 1950 and 2010. Data sources: World Bank 2012 (energy less electric power consumption), BP 2013 (primary energy consumption less electricity generation).

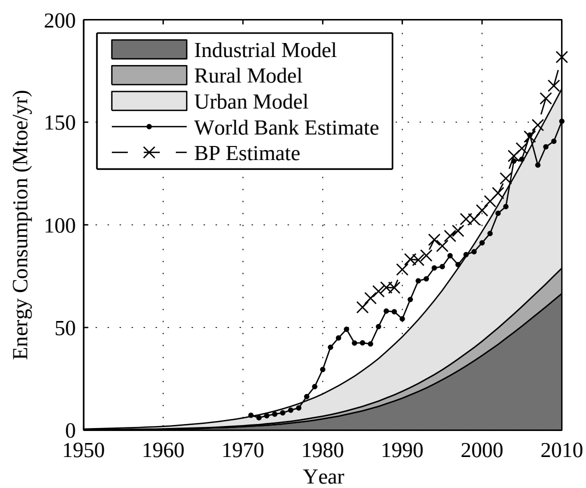

Figure 6-17: A stacked area chart compares aggregated demand model outputs with historical estimates of energy consumption between 1950 and 2010. Data sources: World Bank 2012 (energy less electric power consumption), BP 2013 (primary energy consumption less electricity generation).

Table 6.14: Electricity demand model parameters

Variable	Description	$n_{industrial}$	n_{urban}	n_{rural}	Units
t_0^{elect}	Datum time	1950	1950	1950	year
r_{elect}	Growth rate of per-capita demand	9	9	9	%
d_{min}^{elect}	Minimum per-capita demand	0	0	0	kWh/day
d_0^{elect}	Datum per-capita demand	0.25	0.25	0.25	kWh/day
d_{max}^{elect}	Maximum per-capita demand	40	40	40	kWh/day

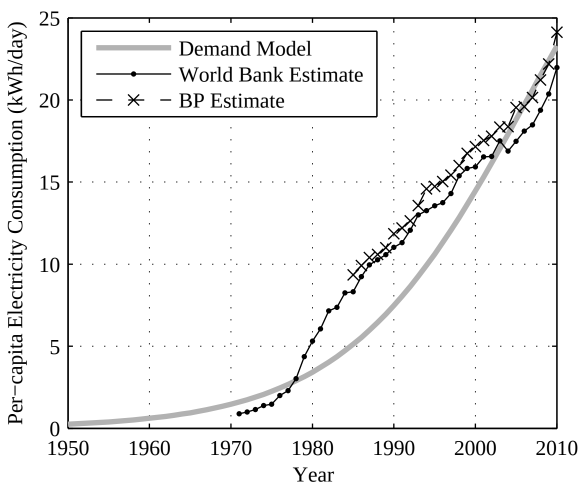

Figure 6-18: Electricity demand model outputs compared to historical estimates of per-capita electricity consumption between 1950 and 2010. Data sources: World Bank 2012 (electricity consumption), BP 2013 (electricity generation).

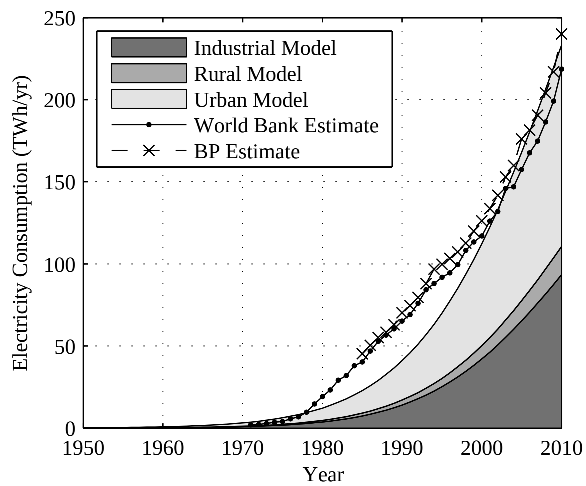

Figure 6-19: A stacked area chart compares aggregated demand model outputs with historical estimates of electricity consumption between 1950 and 2010. Data sources: World Bank 2012 (electricity consumption), BP 2013 (electricity generation).

per day (kWh/day) and a maximum of 40 kWh/day with a 9% annual growth rate. It is fixed to a datum demand of 0.25 kWh/day in 1950. Figures 6-18–6-19 compare per-capita and aggregate electricity consumption estimates and demand model values between 1950-2010. The model slightly under-estimates demands before 1978 and after 2002 and over-estimates demands between the two dates.

6.4.2 Agriculture System Model Instantiation

The agriculture system model instantiations are based on the agricultural sector of Saudi Arabia between 1950 and 2010 simplified to a single crop representing all food, feed, and seed sources of food energy. Cereals compose about half of the food energy production in Saudi Arabia as shown in Figure 6-20, suggesting a single representative crop can approximate a large portion of agricultural production. The following system model parameters, element options, and baseline scenario follow the historical growth of wheat-based agriculture during the 1980s and decline in the 2000s.

Table 6.15: Agriculture system model parameters

Variable	Description	$n_{industrial}$	n_{urban}	n_{rural}	Units
π^{food}_{local}	Local food price	60	60	60	§/GJ
π^{food}_{import}	Import food price	70	70	70	§/GJ
π^{food}_{export}	Export food price	50	50	50	§/GJ
f^{labor}_{people}	Maximum labor participation rate	0.04	0.04	0.40	–
q^{land}_{stock}	Usable agriculture land area	8	10	15	thousand km^2

Agriculture System Model Parameters

The agriculture system model parameters in Table 6.15 set local, import, and export food prices and the stocks of land and labor usable by infrastructure elements. Each node has identical prices such that the export price is below the local price and the import price is above the local price to reflect potential tariffs; however the particular values are balanced for game play.

Labor in the agriculture system is inferred as persons involved with the end-to-end production, processing, and distribution of domestic agriculture. Labor participation rates are balanced with the associated labor required for crop production and should not be considered a valid representation of agricultural workforce. Whereas the industrial and urban nodes have relatively small fractions of agriculture-related labor at 4%, the rural region has a large fraction at 40%.

Agricultural land models are based on estimates of arable land and permanent crop area illustrated in Figure 6-21. As the rural, urban, and industrial nodes are aggregated regions rather than distinct geographical entities, the particular distribution of land is balanced for game play. A national upper bound of 33,000 square kilometers (km^2) for agricultural purposes is mostly held by the rural region (15,000 km^2 or 45%), followed by the urban region (10,000 km^2 or 30%), and finally the industrial region (8,000 km^2 or 25%); however the disaggregation of land among the three nodes is not validated. For comparison, the national agricultural land use in Saudi Arabia between 1990 and 2010 was about 10,000 km^2.

Agriculture Element Model Parameters

Agriculture production elements in Tables 6.16–6.17 define small and large instantiations for wheat production capable of producing up to 2.5 and 5 exajoules per year (EJ/yr). Capital expenses are balanced to allow a positive return on investment in one to two years at maximum capacity. Fixed operations expenses are set to 5% of capital expenses and there are no decommissioning expenses. The specific expense per unit output is balanced to about 40% of total operations expenses with large-sized elements benefiting slightly from economies of scale. Parameters for labor requirements are balanced with labor participation rates, population models, and land constraints. Parameters for water consumption are based on estimates of agricultural aquifer withdrawals and harvested crop area in Figure 6-22. The model parameter of 1.5 million cubic meters per square kilometer (MCM/km^2) is based on the harvested area of all vegetal products to avoid over-estimation if using only cereals which has higher water consumption. Finally, parameters for food energy yield are based on historical data for cereal production in Saudi Arabia in Figure 6-23. Due to agricultural improvements over time, the model parameter of 5 terajoules per square kilometer (TJ/km^2) over-estimates production before 1985.

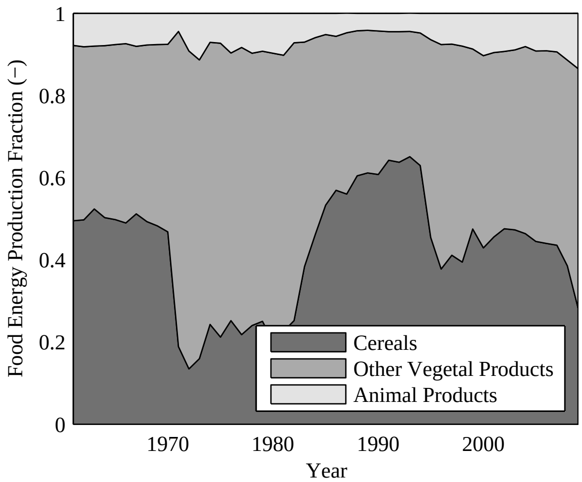

Figure 6-20: Food energy supply by type. Cereals make up about half of food energy supply in Saudi Arabia. Data source: FAO 2013b (harvested area, crop production, and food supply).

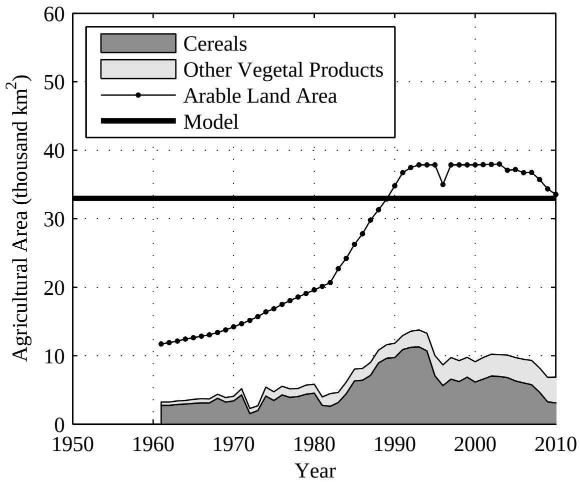

Figure 6-21: Agricultural land model validation. Arable land in Saudi Arabia totals under 40 thousand km^2. Data source: FAO 2013b (arable land and permanent crops, harvested area).

Table 6.16: Agricultural production (A_P) element parameters

Element Type	$p_{capital}$ (§M)	$p_{fixed}/\Delta t$ (§M/yr)	p_{decomm} (§M)	$f_{land}^{currency}/\Delta t$ (§/km^2/yr)
Small Wheat Field	100	5	0	50,000
Large Wheat Field	180	9	0	45,000

Table 6.17: Agricultural production (A_P) element parameters (continued)

Element Type	q_{input}^{max} (km^2)	f_{land}^{labor} (person/km^2)	$f_{land}^{water}/\Delta t$ (MCM/km^2/yr)	$f_{land}^{food}/\Delta t$ (TJ/km^2/yr)
Small Wheat Field	500	60	1.5	5
Large Wheat Field	1000	60	1.5	5

Table 6.18: Agricultural distribution (A_D) element parameters

Element Type	$p_{capital}$ (§M)	$p_{fixed}/\Delta t$ (§M/yr)	p_{decomm} (§M)	$f_{food}^{currency}$ (§/GJ)	$q_{sent}^{max}/\Delta t$ (EJ/yr)	η (–)
Low-volume Transport	50	2.5	0	2	2	0.92
High-volume Transport	300	15	0	2	15	0.94

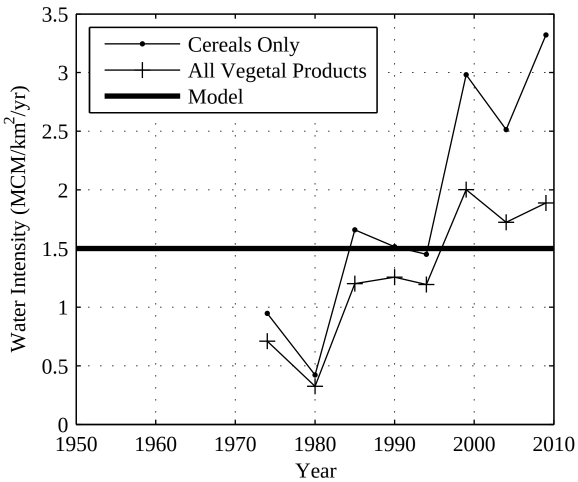

Figure 6-22: Food water use validation. Estimates for water use are based on reported agricultural withdrawals and harvest area. Data sources: FAO 2013b (harvested area), KSA 1990; KSA 1995; KSA 2000; KSA 2005; KSA 2010 (agricultural withdrawals)

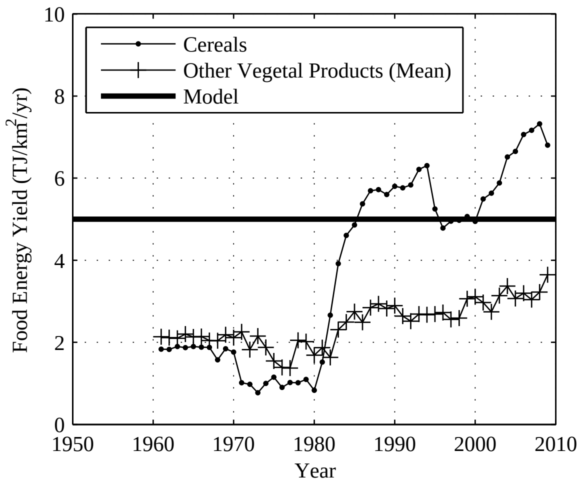

Figure 6-23: Food energy yield validation. Estimates for cereals and other vegetal product yields are based on reported harvest area and food supply data. Data source: FAO 2013b (harvested area, crop production, and food supply).

Agriculture distribution elements in Table 6.18 identify low- and high-volume transport methods with maximum capacities of 2 and 15 EJ/yr. Larger capacity trades higher capital costs for lower operations costs and higher efficiencies (94% compared to 92%). Capital expenses are balanced to contribute about 30-35% of an unadjusted 30-year lifecycle cost. Fixed operations expenses are set to 5% of capital expenses and there are no decommission expenses. The specific expense per unit sent is balanced to 60-65% of the operational expense at maximum capacity. Other parameters are balanced for game play, including capacity limits to accommodate food energy transport of 2 and 15 exajoules per year (EJ/yr) and.

Baseline Agriculture Scenario

The baseline scenario in Table 6.19 defines element model instantiations at locations with associated commissioning and decommissioning times ($t_0, t_\varnothing$) and durations ($d_0, d_\varnothing$). Missing times (–) reflect dates outside of the 1950-2010 time period. All durations are set to zero (i.e. operations can immediately start) to reflect the relatively short timescale required to establish agricultural infrastructure and most elements are not decommissioned over the expected timescale horizon.

Figure 6-24 shows agricultural labor use in the baseline scenario. Figure 6-25 illustrates agricul-

Table 6.19: Agriculture system baseline scenario

t_0	d_0	$t_\varnothing$	$d_\varnothing$	Element(s)	Location
–	0	–	0	Small Wheat Farm 1	$(n_{industrial}, n_{industrial})$
–	0	–	0	Small Wheat Farm 2	(n_{urban}, n_{urban})
–	0	–	0	Small Wheat Farm 3	(n_{rural}, n_{rural})
–	0	–	0	Low-volume Transport 1	$(n_{rural}, n_{industrial})$
–	0	–	0	Low-volume Transport 2	(n_{rural}, n_{urban})
1962	0	–	0	Small Wheat Farm 4	(n_{rural}, n_{rural})
1982	0	–	0	Small Wheat Farm 5	$(n_{industrial}, n_{industrial})$
1982	0	–	0	Small Wheat Farm 6	(n_{urban}, n_{urban})
1984	0	–	0	Large Wheat Farm 1	(n_{rural}, n_{rural})
1984	0	–	0	Large Wheat Farm 2	(n_{rural}, n_{rural})
1984	0	–	0	Large Wheat Farm 3	(n_{rural}, n_{rural})
1984	0	–	0	High-volume Transport 1	$(n_{rural}, n_{industrial})$
1984	0	–	0	High-volume Transport 2	(n_{rural}, n_{urban})
1986	0	2008	0	Large Wheat Farm 4	$(n_{industrial}, n_{industrial})$
1986	0	–	0	Large Wheat Farm 5	(n_{urban}, n_{urban})
1986	0	1996	0	Large Wheat Farm 6	(n_{urban}, n_{urban})
1988	0	–	0	Large Wheat Farm 7	(n_{rural}, n_{rural})
1988	0	–	0	Large Wheat Farm 8	(n_{rural}, n_{rural})
1988	0	1996	0	Large Wheat Farm 9	(n_{urban}, n_{urban})
1990	0	1996	0	Large Wheat Farm 10	$(n_{industrial}, n_{industrial})$
1990	0	1994	0	Large Wheat Farm 11	$(n_{industrial}, n_{industrial})$
1990	0	1994	0	Large Wheat Farm 12	$(n_{industrial}, n_{industrial})$
1990	0	1994	0	Large Wheat Farm 13	$(n_{industrial}, n_{industrial})$
1992	0	1994	0	Large Wheat Farm 14	(n_{urban}, n_{urban})
1992	0	1994	0	Large Wheat Farm 15	(n_{urban}, n_{urban})
2002	0	–	0	Large Wheat Farm 16	(n_{rural}, n_{rural})
2004	0	2008	0	Large Wheat Farm 17	$(n_{industrial}, n_{industrial})$
2004	0	2008	0	Large Wheat Farm 18	(n_{urban}, n_{urban})

tural land use in the baseline scenario compared to historical estimates. As the model only considers a single crop and no animal products to supply food energy, more land is used in later years as compared to historical data. Also, as the food energy yield parameter over-estimates production before 1985, less land is used as compared to historical data in early years.

Figure 6-26 compares results from the baseline agriculture scenario with historical estimates of agricultural water withdrawals. The composite models show slight over-estimation before 1996 and underestimation in years following. Figure 6-27 compares results from the baseline scenario with historical estimates of food energy production, showing generally good agreement.

6.4.3 Water System Model Instantiation

The water system model instantiations are based on the water sector of Saudi Arabia between 1950 and 2010. As a majority of the water production is based on direct withdrawals from aquifers, pump-based private production provides the majority of the water resources. Starting in the 1980s, capital-intensive desalination plants operate in the urban (east coast) and industrial (west coast) nodes to transform seawater into consumable water. The east and west coast water networks are largely separate in Saudi Arabia, as illustrated in Figure 6-28, and the baseline scenario does not include any inter-regional distribution lines.

Water System Model Parameters

The water system model parameters in Table 6.20 set the local and import prices for water, aquifer volumes and recharge rates, and efficiencies for private production of water. Aquifer models are based on estimates of aquifer volumes and historical withdrawals and recharge estimates. Sources agree on an approximate peak in withdrawals between 20-25 cubic kilometers per year (km^3/yr) around 2000 and renewable water sources totaling between 2-8 km^3/yr as illustrated in Figure 6-29 (KSA 2010; World Bank 2005; FAO 2013a). Total aquifer volumes are estimated between 428 and 500 km^3/yr in 1997 and 1985, respectively (KSA 2005; World Bank 2005). The 1997 estimate of 428 km^3/yr from KSA (2005) is used in Figure 6-30 with withdrawal estimates from KSA (2010) to extrapolate initial steady-state aquifer volumes. A maximum and initial national aquifer volume of 600 km^3 is divided across the three nodes such that the rural node access the most (250 km^3 or 42%), followed by the industrial node (200 km^3 or 33%) and the urban node (150 km^3 or 25%). Annual recharge totaling 3.5 km^3/yr are divided such that the urban node receives the highest fraction of its steady-state volume (2.2 km^3/yr or 0.88%), followed by rural node (1.2 km^3/yr or 0.8%), and finally the industrial node (0.1 km^3/yr or 0.05%).

Parameters for private production assume a perfect efficiency of transforming aquifer resources to water. An electrical intensity only considering gravitational potential energy changes and pump system efficiency is estimated using the equation

$$f_{water}^{elect} = \frac{\rho g h}{\eta_{pump}} \tag{6.188}$$

where ρ is the density of water (approximated at 1000 kg/m^3), g is the gravitational acceleration constant (9.8 m/s^2), h is the aquifer depth, and η_{pump} is the pump efficiency. This equation does not consider friction or viscosity losses and assumes the aquifer depth remains constant over time and is at zero gage pressure. Using hypothetical values of η_{pump}=0.3 and h=100 m, the resulting

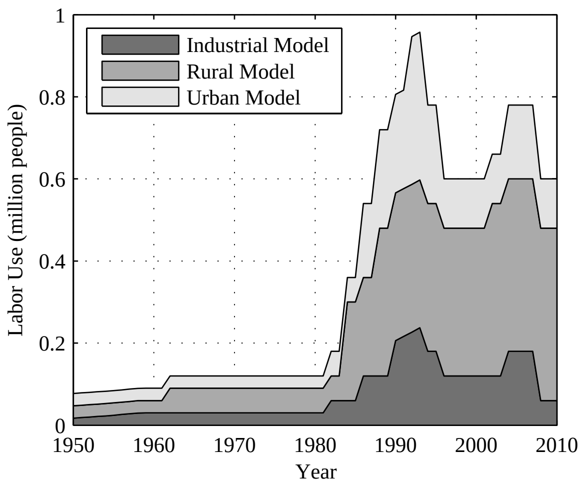

Figure 6-24: Stacked area chart of baseline agriculture scenario labor use.

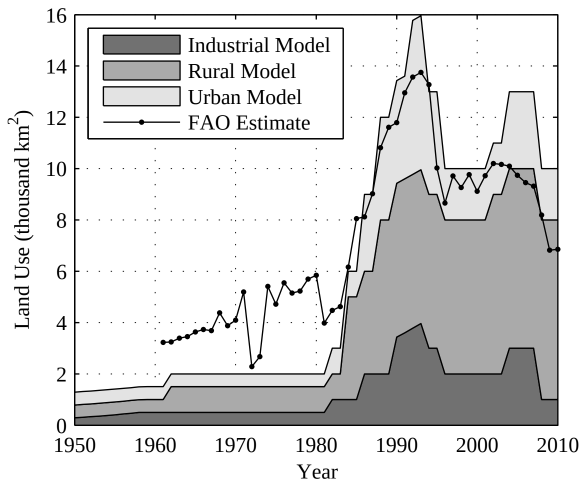

Figure 6-25: Stacked area chart of baseline agriculture scenario land use compared with FAO estimate of harvested area for vegetal products. Data source: FAO 2013b.

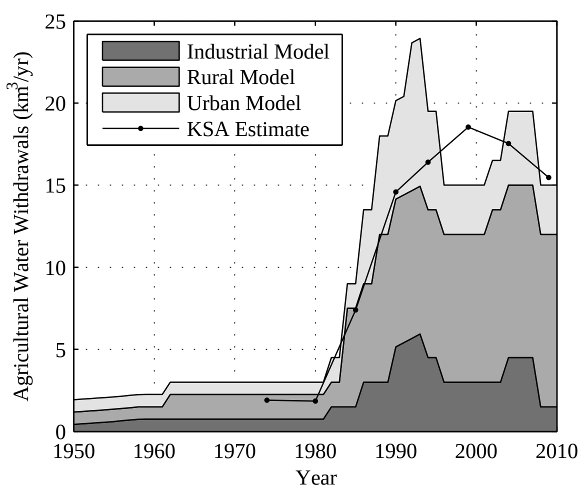

Figure 6-26: Stacked area chart of baseline agriculture scenario water use compared with KSA estimate of agricultural withdrawals. Data sources: KSA 1990; KSA 1995; KSA 2000; KSA 2005; KSA 2010.

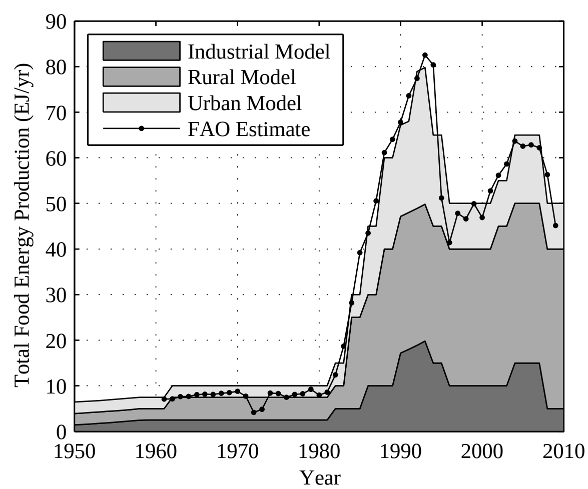

Figure 6-27: Stacked area chart of baseline scenario food production compared with FAO estimate of domestic food supply. Data source: FAO 2013b (total domestic food supply).

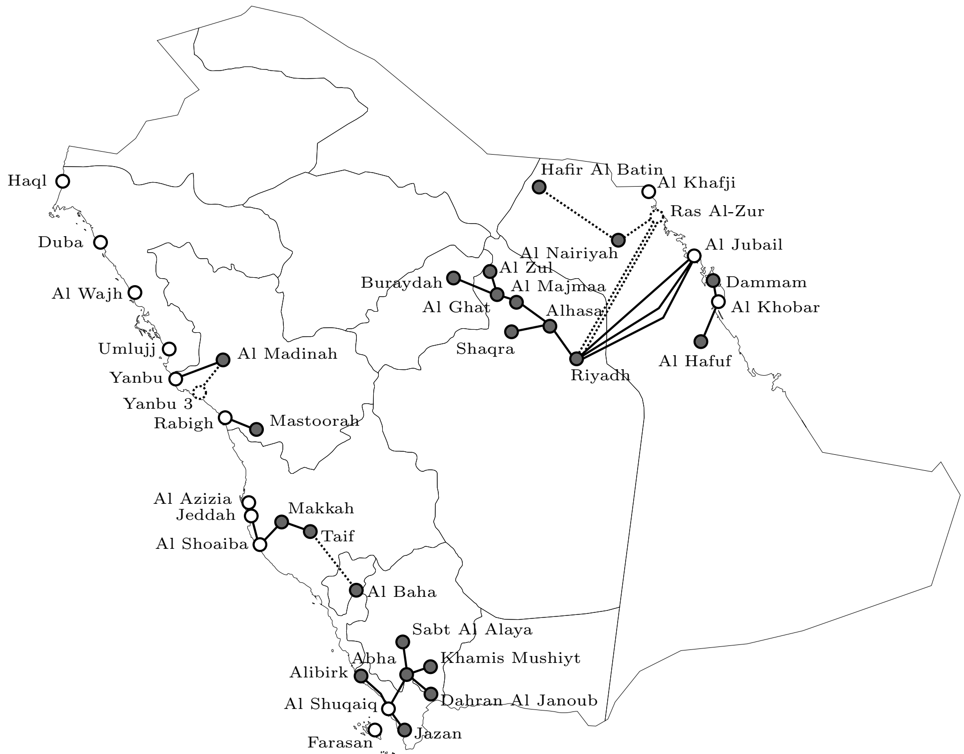

Figure 6-28: Desalination projects in Saudi Arabia. White circles mark desalination plants, gray circles mark supplied cities, solid lines mark pipelines, and dashed lines mark planned plants and pipelines. East and west coast desalination operations are largely separate. Data source: SWCC 2012.

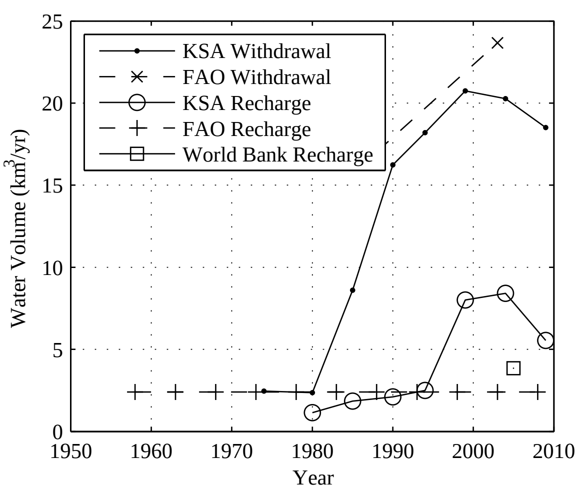

Figure 6-29: Aquifer withdrawal and recharge estimates using KSA, FAO AQUASTAT, and World Bank sources. Data sources: KSA 2005; World Bank 2005; FAO 2013a.

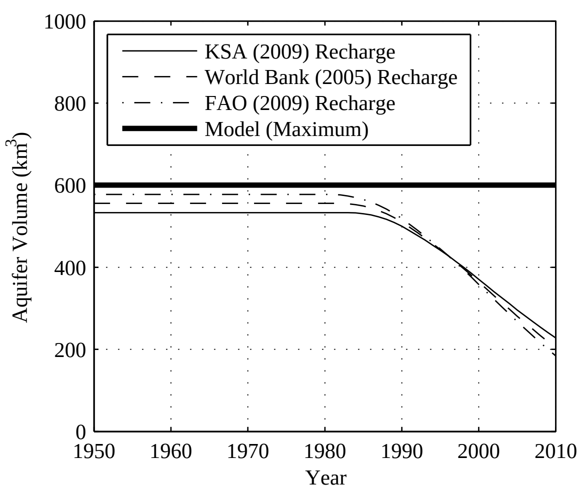

Figure 6-30: Aquifer model volume validation. Back-estimated aquifer volumes using 1997 datum from KSA (2005), KSA withdrawal estimates, and various recharge estimates. Data sources: KSA 2005; World Bank 2005; FAO 2013a.

Table 6.20: Water system model parameters

Variable	Description	$n_{industrial}$	n_{urban}	n_{rural}	Units
π^{water}_{local}	Local water price	0.05	0.05	0.05	§/m^3
π^{water}_{import}	Import water price	10	10	10	§/m^3
$q^{aquifer}_{stock,max}$	Maximum aquifer volume	200	150	250	km^3
$q^{aquifer}_{stock,0}$	Initial aquifer volume	200	150	250	km^3
$r_{recharge}$	Aquifer recharge rate	0.1	2.2	1.2	km^3/yr
$b_{coastal}$	Coastal sea access	1	1	0	–
$f^{aquifer}_{water}$	Private production aquifer efficiency	1.0	1.0	1.0	m^3/m^3
f^{elect}_{water}	Private production electrical intensity	0.9	0.9	0.9	kWh/m^3

Table 6.21: Water production (W_P) element parameters

Element Type	$p_{capital}$ (§M)	$p_{fixed}/\Delta t$ (§M/yr)	p_{decomm} (§M)	$f^{currency}_{water}$ (§/m^3)	$q^{max}_{produced}/\Delta t$ (MCM/yr)	f^{elect}_{water} (kWh/m^3)	$f^{aquifer}_{water}$ (m^3/m^3)
Small RO Plant	200	1.0	5.0	0.014	50	5.5	0.0
Large RO Plant	500	2.5	12.5	0.012	150	4.5	0.0
Huge RO Plant	2000	10.0	50.0	0.012	600	4.5	0.0

electrical intensity is 0.9 kilowatt hours per cubic meter (kWh/m^3) to one significant digit. Local prices are set at nearly 14 times the electricity resource cost of private production and import prices are set at 200 times the local water price to reflect the large cost of long-distance transportation, however the specific prices are balanced for game play.

Water Element Model Parameters

The water production elements in Table 6.21 identify small-, large-, and huge-capacity reverse osmosis (RO) desalination plants with capacity limits of 50 million cubic meters per year (MCM/year), 150 MCM/year, and 600 MCM/year respectively. Each RO plant require coastal access to produce water without using aquifer reserves. Capital expenses are balanced to about twice the maximum operational expenses over a 30 year lifecycle. Fixed operations expenses are set to 0.5% of capital expenses and decommission expenses are set to 5% of capital expenses. The specific expense per unit output is balanced to about 20% of total operations expense at maximum capacity. Small-capacity RO plants have a relatively high specific energy at 5.5 kWh/m^3 which is improved to 4.5 kWh/m^3 for large- and huge-capacity plants to reflect economies of scale and technology improvements. These values compare with a theoretical minimum of less than 1.0 kWh/m^3 and typical values of 3.7–5.3 kWh/m^3 for seawater RO plants (Avlonitis et al. 2003).

The water distribution elements in Table 6.22 identify low- and high-volume transport methods with maximum capacities 50 and 150 MCM/yr. Capital expenses are balanced to contribute about 30% of expenses over a 30 year lifecycle. Fixed operations expenses are set to 2% of capital expenses and there are no decommission expenses. The specific expense per unit sent is balanced to a third of total operations expense at maximum capacity. Specific energy for distribution is set to 2 kWh/m^3

Table 6.22: Water distribution (W_D) element parameters

Element Type	$p_{capital}$ (§M)	$p_{fixed}/\Delta t$ (§M/yr)	p_{decomm} (§M)	$f_{water}^{currency}$ (§/m^3)	$q_{sent}^{max}/\Delta t$ (MCM/yr)	f_{water}^{elect} (kWh/m^3)	η (–)
Low-volume Pipeline	20	0.4	0	0.008	50	2	0.88
High-volume Pipeline	50	1.0	0	0.008	150	2	0.9

Table 6.23: Water system baseline scenario

t_0	d_0	$t_\varnothing$	$d_\varnothing$	Element(s)	Location
1978	2	–	1	Small RO Plant 1	$(n_{industrial}, n_{industrial})$
1980	2	–	1	Small RO Plant 2	(n_{urban}, n_{urban})
1982	2	–	1	Large RO Plant 1	$(n_{industrial}, n_{industrial})$
1982	2	–	1	Small RO Plant 3	(n_{urban}, n_{urban})
1988	2	–	1	Large RO Plant 2	(n_{urban}, n_{urban})
1988	2	–	1	Large RO Plant 3	$(n_{industrial}, n_{industrial})$
1992	2	–	1	Large RO Plant 4	(n_{urban}, n_{urban})
1994	2	–	1	Small RO Plant 4	$(n_{industrial}, n_{industrial})$
2002	2	–	1	Large RO Plant 5	$(n_{industrial}, n_{industrial})$
2002	2	–	1	Large RO Plant 6	(n_{urban}, n_{urban})

— higher than local private production and lower than RO production. More detailed distribution models may parameterize expenses, specific energy, and efficiencies by the distance and elevation change between nodes.

Baseline Water Scenario

The baseline scenario in Table 6.23 defines element instantiations with associated commissioning and decommissioning times $(t_0, t_\varnothing)$ and durations $(d_0, d_\varnothing)$. Missing times (–) reflect dates outside of the 1950-2010 time period. A total of two small RO plants and three large RO plants are commissioned in both industrial and urban nodes between 1978 and 2000. All plants are still operational as of 2010 with a maximum operational duration of 50 years. The baseline scenario does not include any regional water distribution to reflect the separation of east and west coast water networks in Saudi Arabia.

Figure 6-31 shows a stacked area chart of electricity use by water elements during the baseline scenario, not including electricity required for private production. At maximum capacity, water infrastructure in the industrial and urban nodes each require about 2.5 terawatt hours per year (TWh/yr). Validation data is not immediately available for desalination consumption of electricity in Saudi Arabia due to the common use of combined-cycle desalination power plants which only report net electricity output.

Figure 6-32 compares the water production in the baseline scenario to historical estimates from KSA and SWCC documents. The composite models show good agreement with historical trends which is disaggregated approximately equally between east and west coast desalination.

Table 6.24: Petroleum system model parameters

Variable	Description	$n_{industrial}$	n_{urban}	n_{rural}	Units
π^{oil}_{local}	Local oil price	8	8	8	§/toe
π^{oil}_{import}	Import oil price	35	35	35	§/toe
π^{oil}_{export}	Export oil price	30	30	30	§/toe
$q^{reservoir}_{stock,max}$	Maximum oil reservoir volume	65	0	0	Btoe
$q^{reservoir}_{stock,0}$	Initial oil reservoir volume	65	0	0	Btoe

Figures 6-29 and 6-30 compare the water withdrawals and aquifer volumes from the baselines scenario with historical estimates. The withdrawal estimates generally agree with data largely due to validation of the agricultural system.

6.4.4 Petroleum System Model Instantiation

The petroleum system model instantiations are based on the combined oil and natural gas sector of Saudi Arabia between 1950 and 2010. During this time period, petroleum production rises and export to establish Saudi Arabia as a world leader. The petroleum system model captures the oil and gas resources held in the industrial region and element models capture extraction and distribution among the three regions.

Petroleum System Model Parameters

The petroleum system model parameters in Table 6.24 set the local, import, and export prices for oil and reservoir volumes. The import and export prices are set about four times the local price to represent higher global market prices with import higher than export to reflect potential tariffs, however the particular values are balanced for game play.

The oil reservoir model is based on estimates of proven oil and gas reserves with back-estimated historical production. Energy production estimates ranges from around 100 million tonnes of oil equivalent per year (Mtoe/yr) in 1975 to about 600 Mtoe/yr in 2005 as shown in Figure 6-35. Proven reserves are estimated at 43.9 billion tonnes of oil equivalent (Btoe) including 36.5 Btoe in oil and 8.2 billion ft^3 in gas at 0.9 toe/ft^3 in 2012 (BP 2013), leading to a datum reservoir volume estimate of 65 Btoe in 1950 as illustrated in Figure 6-36.

Petroleum Element Model Parameters

The petroleum production elements in Table 6.25 include small and large oil wells with maximum production capacities of 25 and 100 Mtoe/yr respectively. Capital expenses are balanced to provide a positive return on investment in about half a year at maximum production and export prices. Fixed operations expenses are set to 5% of capital expenses and decommission expenses are also set to 5% of capital expenses. The specific expense per unit output is balanced to set unit production costs near the local price, with larger elements benefit from economies of scale to decrease specific expenses per unit output. Future extensions may specify specific energy and cost of oil production as a function of reservoir volume.

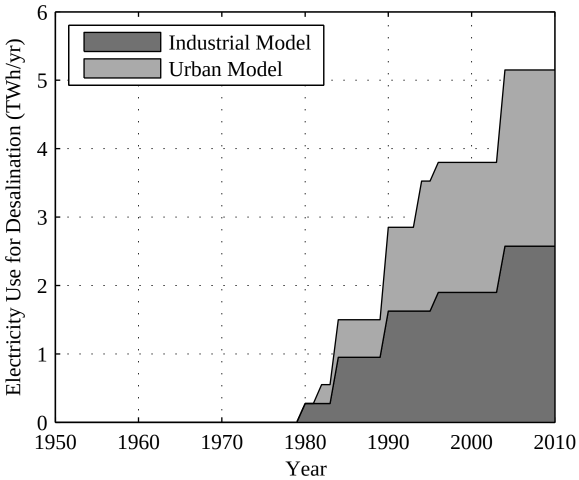

Figure 6-31: Stacked area chart of baseline water scenario electricity use. Does not include electricity for private production of water via pumping from aquifers.

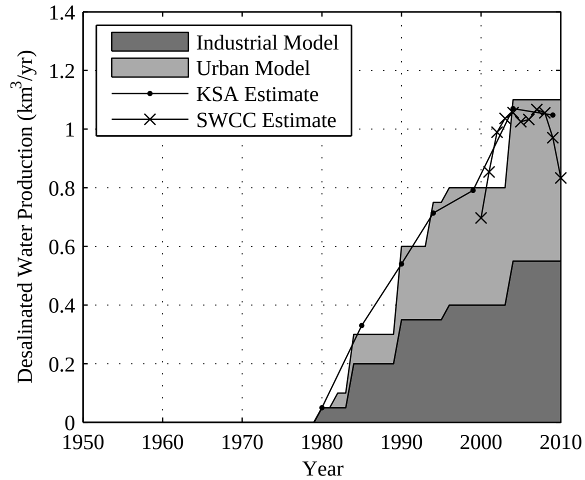

Figure 6-32: Stacked area chart of baseline water scenario production compared with KSA and SWCC estimates. Data sources: KSA 1990; KSA 1995; KSA 2000; KSA 2005; KSA 2010; SWCC 2004; SWCC 2008; SWCC 2012.

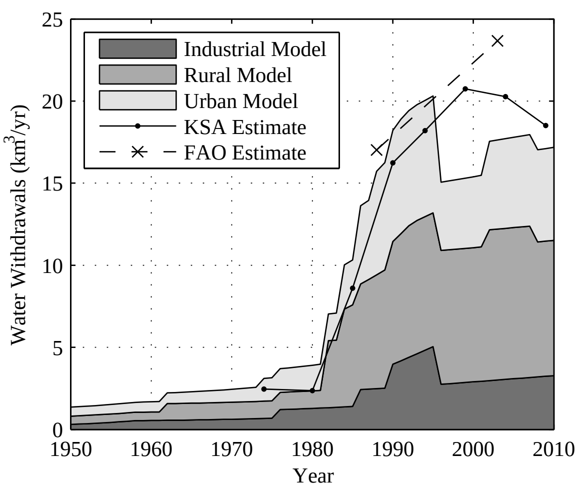

Figure 6-33: Stacked area chart of baseline water scenario withdrawals compared with KSA and FAO AQUASTAT estimates. Data sources: KSA 1990; KSA 1995; KSA 2000; KSA 2005; KSA 2010; FAO 2013a.

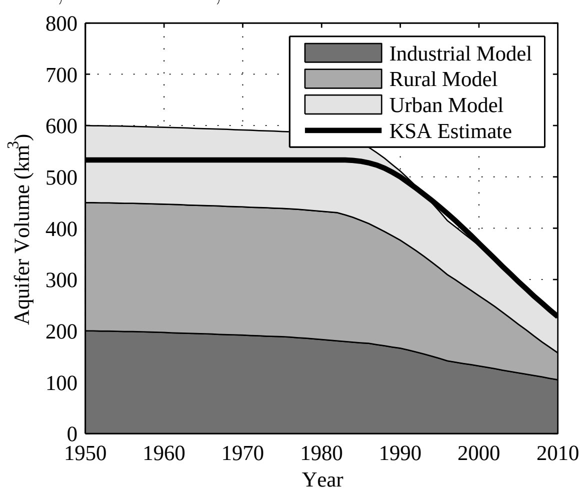

Figure 6-34: Stacked area chart of baseline water scenario aquifer use compared with KSA estimates of recharge and datum aquifer volume. Data source: KSA 2005.

Table 6.25: Petroleum production (P_P) element parameters

Element	$p_{capital}$ (§M)	$p_{fixed}/\Delta t$ (§M/yr)	p_{decomm} (§M)	$f_{oil}^{currency}$ (§/toe)	$q_{produced}^{max}/\Delta t$ (Mtoe/yr)	f_{oil}^{elect} (kWh/toe)	$f_{oil}^{reservoir}$ (toe/toe)
Small Oil Well	500	25.0	25.0	6.00	25	0	1.0
Large Oil Well	1750	87.5	87.5	5.75	100	0	1.0

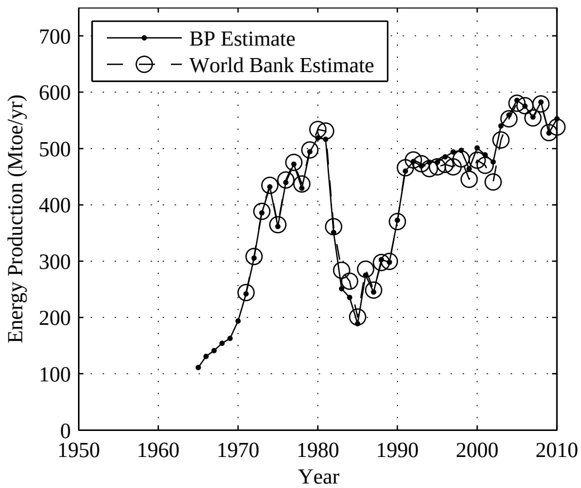

Figure 6-35: Oil reservoir withdrawal estimates. Data sources: BP 2013 (oil and gas production), World Bank 2012 (energy production).

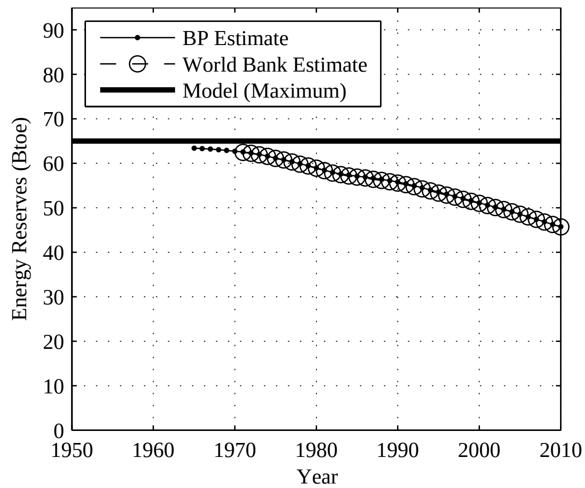

Figure 6-36: Oil reservoir model volume validation. Back-estimated using 2012 proven reserve estimates from BP (2013) and various withdrawal estimates. Data sources: BP 2013 (oil and gas production and proven reserves), World Bank 2012 (energy production).

Table 6.26: Petroleum distribution (P_D) element parameters

Element	$p_{capital}$ (§M)	$p_{fixed}/\Delta t$ (§M/yr)	p_{decomm} (§M)	$f_{oil}^{currency}$ (§/toe)	$q_{sent}^{max}/\Delta t$ (Mtoe/yr)	f_{oil}^{elect} (kWh/toe)	η (–)
Low-volume Pipeline	100	2	0	0.1	10	2	0.98
High-volume Pipeline	450	9	0	0.1	50	2	0.99

The petroleum distribution elements in Table 6.26 identify low- and high-volume transport methods with maximum capacities 10 and 50 Mtoe/yr. Capital expenses are balanced to account for about half of unit transport costs over a 30 year lifecycle. Fixed operations expenses are set to 2% of capital expenses are there are no decommission expenses. Specific expenses per unit sent are balanced to account for about a third of total operations expenses. More detailed distribution models may parameterize expenses, electrical intensity, and efficiencies by the distance and elevation change between nodes.

Baseline Petroleum Scenario

The baseline scenario described in Table 6.27 defines element instantiations with associated commissioning and decommissioning times ($t_0, t_\varnothing$) and durations ($d_0, d_\varnothing$). Missing times (–) reflect dates outside of the 1950-2010 time period.

Figure 6-37 compares baseline oil production with historical estimates. The features of capacity increases until 1980, followed by production cuts and recovery are clearly apparent. Oil production reaches a capacity of 600 Mtoe in the mid 2000s. Figure 6-38 compares baseline oil reservoir volume with historical estimates, exhibiting good fit.

Table 6.27: Petroleum system baseline scenario

t_0	d_0	$t_\varnothing$	$d_\varnothing$	Element(s)	Location
–	1	–	0	Small Oil Well 1	$(n_{industrial}, n_{industrial})$
–	1	1978	0	Low-volume Pipeline 1	$(n_{industrial}, n_{urban})$
–	1	1990	0	Low-volume Pipeline 2	$(n_{industrial}, n_{rural})$
1950	1	–	0	Small Oil Well 2	$(n_{industrial}, n_{industrial})$
1955	1	–	0	Small Oil Well 3	$(n_{industrial}, n_{industrial})$
1962	1	–	0	Small Oil Well 4	$(n_{industrial}, n_{industrial})$
1964	1	–	0	Small Oil Well 5	$(n_{industrial}, n_{industrial})$
1966	1	–	0	Small Oil Well 6	$(n_{industrial}, n_{industrial})$
1968	1	–	0	Small Oil Well 7	$(n_{industrial}, n_{industrial})$
1970	1	–	0	Small Oil Well 8	$(n_{industrial}, n_{industrial})$
1970	2	1981	1	Large Oil Well 1	$(n_{industrial}, n_{industrial})$
1972	2	1982	1	Large Oil Well 2	$(n_{industrial}, n_{industrial})$
1976	2	1983	1	Large Oil Well 3	$(n_{industrial}, n_{industrial})$
1976	2	–	0	High-volume Pipeline 1	$(n_{industrial}, n_{urban})$
1976	1	–	0	Low-volume Pipeline 3	$(n_{industrial}, n_{rural})$
1984	2	–	1	Large Oil Well 4	$(n_{industrial}, n_{industrial})$
1988	2	–	1	Large Oil Well 5	$(n_{industrial}, n_{industrial})$
1988	1	–	0	Low-volume Pipeline 4	$(n_{industrial}, n_{rural})$
1990	2	–	1	Large Oil Well 6	$(n_{industrial}, n_{industrial})$
1992	2	–	0	High-volume Pipeline 2	$(n_{industrial}, n_{urban})$
2002	2	–	1	Large Oil Well 7	$(n_{industrial}, n_{industrial})$
2004	2	–	0	High-volume Pipeline 3	$(n_{industrial}, n_{urban})$
2008	1	–	0	Low-volume Pipeline 5	$(n_{industrial}, n_{rural})$

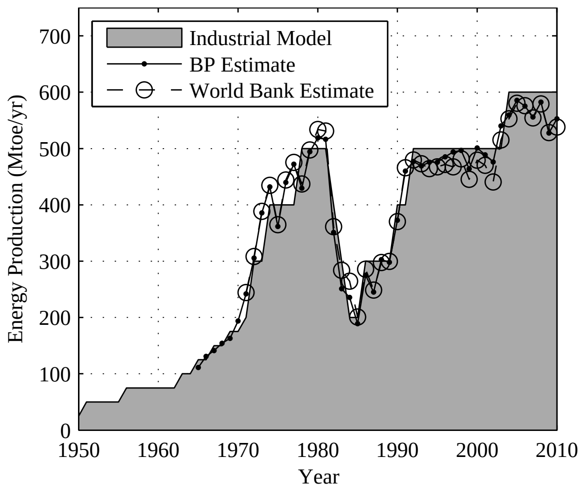

Figure 6-37: Baseline petroleum scenario oil production with historical estimates. Data sources: BP 2013 (oil and gas production), World Bank 2012 (energy production).

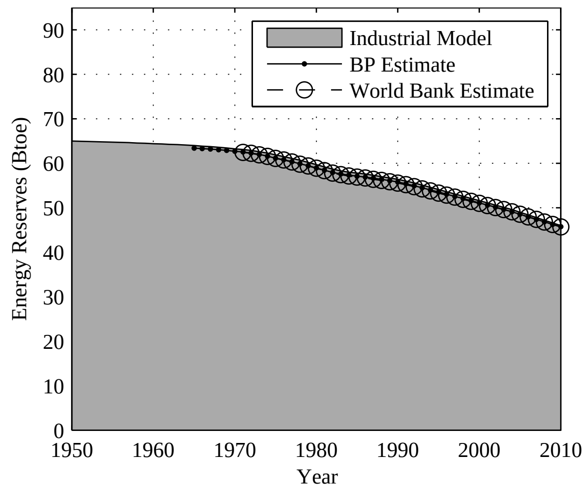

Figure 6-38: Baseline petroleum scenario oil reservoir use compared with historical estimates. Data sources: BP 2013 (oil and gas production and proven reserves), World Bank 2012 (energy production).

Table 6.28: Electricity system model parameters

Variable	Description	$n_{industrial}$	n_{urban}	n_{rural}	Units
π_{local}^{oil}	Local electricity price	4	4	4	§/MWh
f_{elect}^{oil}	Private electrical production efficiency	0.5	0.5	0.5	toe/MWh

6.4.5 Electricity System Model Instantiation

The electricity system model instantiations are based on the electricity sector of Saudi Arabia between 1950 and 2010. During this time period, fossil fuels are the exclusive source of electricity, although renewable technologies exist elsewhere. The electricity system model captures the public electrical generation and distribution among the three regions.

Electricity System Model Parameters

The electricity system model parameters in Table 6.28 set the local price of electricity and the specific oil consumption for private production. Considered to be an unimproved power generation method, private electricity production is set to convert oil energy at 17% efficiency.[3] The local price is set to equal the direct cost of oil for private electricity production.

Electricity Element Model Parameters

The electrical production elements in Tables 6.29– 6.30 create small and large instantiations for two generation methods with maximum capacities 2 and 10 terawatt hours per year (TWh/yr). Thermal power plants have capital expenses balanced to provide a positive return on investment

[3] Efficiency calculated with conversion 1 toe = 11.64 MWh as 1/(0.5 toe/MWh)/(11.64 MWh/toe) = 0.17.

Table 6.29: Electrical production (E_P) element parameters

Element	$p_{capital}$ (§M)	$p_{fixed}/\Delta t$ (§M/yr)	p_{decomm} (§M)	$f_{elect}^{currency}$ (§/MWh)
Small Thermal Power Plant	25	0.25	1.25	0
Large Thermal Power Plant	150	1.50	7.50	0
Small Solar PV Power Plant	200	3.00	2.00	0
Large Solar PV Power Plant	900	13.50	9.00	0

Table 6.30: Electrical production (E_P) element parameters (continued)

Element	$q_{produced}^{max}/\Delta t$ (TWh/yr)	f_{elect}^{oil} (toe/MWh)	f_{elect}^{water} (m^3/MWh)
Small Thermal Power Plant	2	0.30	0
Large Thermal Power Plant	10	0.25	0
Small Solar PV Power Plant	2	0	0
Large Solar PV Power Plant	10	0	0

in 8–9 years with fixed operational expenses at 1% of capital expenses and decommission expenses at 5% of capital expenses. Small and large thermal power plants convert oil energy at 29% and 34% efficiencies respectively. Solar photo-voltaic (PV) plants have capital expenses balanced to produce lifecycle costs similar to thermal plants with fixed operational expenses set at 1.5% of capital expenses and decommission expenses at 1% of capital expenses.

The electrical distribution elements in Table 6.31 identify low- and high-capacity transport methods with maximum capacities 10 and 50 TWh/yr. Capital expenses are balanced to contribute about half of the total expenses over a 30 year lifecycle. Fixed operational expenses are set to 4% of the capital expenses and there are no decommission expenses. More detailed distribution models may parameterize expenses and efficiencies by the distance and elevation change between nodes and other properties such as line voltage.

Baseline Electricity Scenario

The baseline scenario described in Table 6.32 defines element instantiations with associated commissioning and decommissioning times ($t_0, t_\varnothing$) and durations ($d_0, d_\varnothing$).

Figure 6-39 illustrates oil consumed to produce electricity via both private and infrastructure production methods. Figure 6-40 compares electricity production from baseline infrastructure with

Table 6.31: Electrical distribution (E_D) element parameters

Element	$p_{capital}$ (§M)	$p_{fixed}/\Delta t$ (§M/yr)	p_{decomm} (§M)	$f_{elect}^{currency}$ (§/MWh)	$q_{sent}^{max}/\Delta t$ (TWh/yr)	η (–)
Low-capacity Power Line	50	2	0	0	10	0.94
High-capacity Power Line	225	9	0	0	50	0.96

Table 6.32: Electricity system baseline scenario

t_0	d_0	$t_\varnothing$	$d_\varnothing$	Element(s)	Location
1950	1	–	0	Small Thermal Power Plant 1	(n_{urban}, n_{urban})
1960	1	–	0	Small Thermal Power Plant 2	$(n_{industrial}, n_{industrial})$
1966	1	–	0	Small Thermal Power Plant 3	(n_{rural}, n_{rural})
1970	1	–	0	Low-capacity Power Line 1	$(n_{industrial}, n_{rural})$
1972	1	–	0	Low-capacity Power Line 2	$(n_{industrial}, n_{rural})$
1972	2	–	0	Large Thermal Power Plant 1	(n_{urban}, n_{urban})
1974	2	–	0	Large Thermal Power Plant 2	$(n_{industrial}, n_{industrial})$
1982	2	–	0	Large Thermal Power Plant 3	(n_{urban}, n_{urban})
1984	2	–	0	Large Thermal Power Plant 4	$(n_{industrial}, n_{industrial})$
1986	1	–	0	Small Thermal Power Plant 4	(n_{rural}, n_{rural})
1988	2	–	0	Large Thermal Power Plant 5	(n_{urban}, n_{urban})
1990	2	–	0	Large Thermal Power Plant 6	(n_{urban}, n_{urban})
1992	2	–	0	Large Thermal Power Plant 7	$(n_{industrial}, n_{industrial})$
1994	2	–	0	Large Thermal Power Plant 8	(n_{urban}, n_{urban})
1996	1	–	0	Small Thermal Power Plant 5	(n_{rural}, n_{rural})
1996	2	–	0	Large Thermal Power Plant 9	(n_{urban}, n_{urban})
1998	2	–	0	Large Thermal Power Plant 10	$(n_{industrial}, n_{industrial})$
2000	2	–	0	Large Thermal Power Plant 11	$(n_{industrial}, n_{industrial})$
2000	1	–	0	Small Thermal Power Plant 6	(n_{rural}, n_{rural})
2002	2	–	0	Large Thermal Power Plant 12	$(n_{industrial}, n_{industrial})$
2002	2	–	0	Large Thermal Power Plant 13	(n_{urban}, n_{urban})
2004	2	–	0	Large Thermal Power Plant 14	$(n_{industrial}, n_{industrial})$
2004	2	–	0	Large Thermal Power Plant 15	(n_{urban}, n_{urban})
2006	2	–	0	Large Thermal Power Plant 16	$(n_{industrial}, n_{industrial})$
2006	2	–	0	Large Thermal Power Plant 17	(n_{urban}, n_{urban})
2008	2	–	0	Large Thermal Power Plant 18	(n_{urban}, n_{urban})

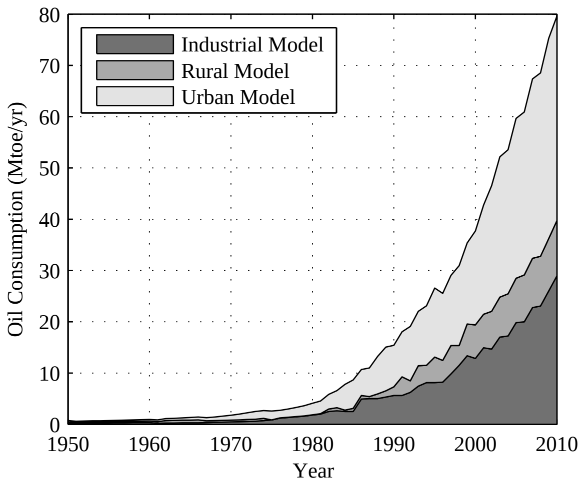

Figure 6-39: Stacked area chart of baseline electricity scenario oil consumption. Includes both private production and infrastructure element production.

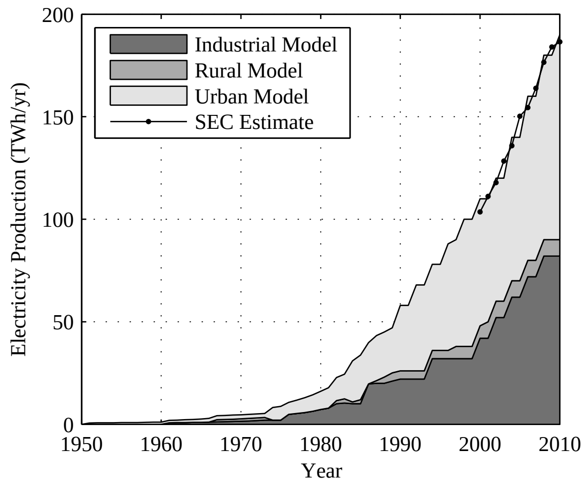

Figure 6-40: Stacked area chart of baseline electricity scenario production compared with SEC estimates. Does not include private production. Data sources: SEC 2003; SEC 2005; SEC 2007; SEC 2009; SEC 2011.

historical estimates since 2000. The magnitude of production and capacity increases between 2000-2010 show good fit.

References for Chapter 6

Avlonitis, S., K. Kouroumbas, and N. Vlachakis (2003). "Energy consumption and membrane replacement cost for seawater RO desalination plants". In: *Desalination* 157, pp. 151–158.

BP (2013). *BP Statistical Review of World Energy*. URL: `http://www.bp.com/statisticalreview/` (visited on 07/17/2013).

Brewer, G. D. (1973). *Politicians, Bureaucrats, and the Consultant*. New York: Basic Books, Inc.

Food and Agriculture Organization of the United Nations (2013a). *AQUASTAT Main Country Database*. URL: `http://www.fao.org/nr/water/aquastat/dbase/index.stm` (visited on 10/25/2013).

— (2013b). *FAOSTAT*. URL: `http://faostat3.fao.org/` (visited on 07/15/2013).

Harteveld, C. (2011). *Triadic Game Design: Balancing Reality, Meaning, and Play*. London: Springer.

Hasnain, S. (1998). "Review on Sustainable Thermal Energy Storage Technologies Part II: Cool Thermal Storage". In: *Energy Conservation in Management* 39.11, pp. 1139–1153. DOI: `10.1016/S0196-8904(98)00024-7`.

Kingdom of Saudi Arabia (1970). *The 1st Development Plan*. Riyadh: Ministry of Planning. URL: `http://www.mep.gov.sa/` (visited on 07/14/2013).

— (1975). *The 2nd Development Plan*. Riyadh: Ministry of Planning. URL: `http://www.mep.gov.sa/` (visited on 07/14/2013).

— (1990). *The 5th Development Plan*. Riyadh: Ministry of Planning. URL: `http://www.mep.gov.sa/` (visited on 07/14/2013).

Kingdom of Saudi Arabia (1995). *The 6th Development Plan*. Riyadh: Ministry of Planning. URL: http://www.mep.gov.sa/ (visited on 07/14/2013).

— (2000). *The 7th Development Plan*. Riyadh: Ministry of Planning. URL: http://www.mep.gov.sa/ (visited on 07/14/2013).

— (2005). *The 8th Development Plan*. Riyadh: Ministry of Economy and Planning. URL: http://www.mep.gov.sa/ (visited on 07/14/2013).

— (2010). *The Ninth Development Plan*. Riyadh: Ministry of Economy and Planning. URL: http://www.mep.gov.sa/ (visited on 07/14/2013).

— (2013). *Forty Eighth Annual Report: The Latest Economic Developments*. Riyadh: Saudi Arabian Monetary Agency. URL: http://www.sama.gov.sa/sites/samaen/ReportsStatistics/ (visited on 07/19/2013).

Kristensen, P. (2004). "The DPSIR framework". In: *Comprehensive/Detailed Assessment of the Vulnerability of Water Resources to Environmental Change in Africa using River Basin Approach*.

Saline Water Conversion Corporation (2004). *Annual Report 2004*. Riyadh. URL: http://www.swcc.gov.sa/ (visited on 11/05/2013).

— (2008). *Annual Report No. 35*. Riyadh. URL: http://www.swcc.gov.sa/ (visited on 11/05/2013).

— (2012). *Annual Report No. 39*. Riyadh. URL: http://www.swcc.gov.sa/ (visited on 11/05/2013).

Saudi Electric Company (2003). *2003 Annual Report*. Riyadh. URL: http://www.sec.com.sa/ (visited on 11/12/2013).

— (2005). *2005 Annual Report*. Riyadh. URL: http://www.sec.com.sa/ (visited on 11/12/2013).

— (2007). *2007 Annual Report*. Riyadh. URL: http://www.sec.com.sa/ (visited on 11/12/2013).

— (2009). *2009 Annual Report*. Riyadh. URL: http://www.sec.com.sa/ (visited on 11/12/2013).

— (2011). *2011 Annual Report*. Riyadh. URL: http://www.sec.com.sa/ (visited on 11/12/2013).

Siddiqi, A. and L. D. Anadon (2011). "The water–energy nexus in Middle East and North Africa". In: *Energy Policy* 39.8, pp. 4529–4540. DOI: 10.1016/j.enpol.2011.04.023.

United Nations (2013). *The Millennium Development Goals Report 2013*. United Nations Statistics Division Department of Economic and Social Affairs. URL: http://mdgs.un.org/unsd/mdg/ (visited on 07/19/2013).

United Nations, Department of Economic and Social Affairs, Population Division (2013). *World Population Prospects: The 2012 Revision*. URL: http://esa.un.org/unpd/wpp/ (visited on 07/13/2013).

World Bank (2005). *A Water Sector Assessment Report on the Countries of the Cooperation Council of the Arab States of the Gulf*. 32539-MNA. World Bank Water, Environment, Social, Rural Development Department, Middle East, and North Africa Region.

— (2012). *World Development Indicators*. URL: http://data.worldbank.org/country/saudi-arabia/ (visited on 07/13/2013).

Chapter 7

Supporting Collaborative Design

> "Metrics that describe the operating states of interdependent infrastructures and scale of interdependency-related disruptions are sorely lacking. These metrics should include a range of economic, social, and national security considerations."
>
> Steven M. Rinaldi, James P. Peerenboom, and Terrence K. Kelly in "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies" (2001)

An integrated modeling tool such as the SIPS-G application considers the system-wide impacts of design decisions to evaluate trade-offs between alternatives. In infrastructure systems, this captures the effects of cross-sector resource interdependencies on the overall objective of sustainability. Similar to the observation by Rinaldi et al. (2001) above in the context of operational infrastructure resiliency, a strategic sustainability objective should incorporate a wide range of factors including economics, social equity, and environmental impact as a basis for decision-making.

While this chapter does not tackle the challenging question of how to formulate a measure of sustainability for infrastructure systems, it does address the implications of using similar measures in collaborative design. Section 7.1 reviews literature in related areas of decision-making processes among multiple actors with potentially-competing objectives. Section 7.2 introduces a role-play design scenario using the SIPS-G application in the fictional nation of Idas Abara to study infrastructure systems planning. A formulation of individual objectives purposefully creates conflict between roles as well as identifies a consistent joint objective. Section 7.3 outlines a design experiment to study the impact of software tool variations in a time-constrained design activity. Section 7.4 presents experimental results and analysis. Finally, Section 7.5 discusses results and future work.

7.1 Collaboration in Design

Lu et al. describe coordination, cooperation, and collaboration as the three basic levels of collective human endeavors. The three terms are distinguished by:

> while coordination avoids gaps or overlaps in individuals' assigned tasks, and cooperation strives for mutual benefits by sharing or partitioning tasks, collaboration aims at achieving a common goal and collective results that individuals would be incapable of accomplishing alone. In other words, collaboration requires a team of individuals to work on tasks that not only have shared resources (as in coordination) and shared outcomes (as in cooperation), but, most importantly, a shared goal. (Lu et al. 2007)

Table 7.1: Methods to solve engineering design problems based on Lu et al. (2007)

	Design Objectives	
Decision Style	Natural Phenomena	Human Preferences
Isolated and Individual	Optimization	Classical Decision Analysis
Interactive but Separate	Multi-objective Optimization	Non-cooperative Game Theory
Joint and Collective	Exploration of Objectives	Co-construction and Negotiation

Infrastructure systems design, particularly at a national level, is a collaborative activity. It involves multiple actors each with partial control over constituent systems, shared resources for investment in new infrastructure, shared outcomes based on resource interdependencies between actors, and a shared goal of sustainability. Relying on human preferences rather than evaluation of natural phenomena is a major challenge to achieving shared goals. To elaborate on this point, Table 7.1 illustrates six types of engineering design problems classified by decision style and design objectives using the collaborative engineering problem roadmap from Lu et al. (2007).

Traditional engineering activities seek design objectives related to natural phenomena, e.g. stability of buildings, load capacity of bridges, and electrical properties of circuits. Systems of equations modeling natural phenomena can be used in mathematical optimization to solve isolated decisions for individual actors. Extensions of multi-objective or multi-criteria optimization methods may be applied to solve interactive decisions across separate actors. For joint decisions among collective actors, collaborative methods are required to explore objectives and arrive at a single decision. Concurrent engineering, for example, focuses less on multi-objective optimization and more on exploration and information-sharing across actors to identify a joint design solution.

Many contemporary engineering activities take a broader perspective to achieve design objectives related to human preferences. In this framing, a design solution only remains valid while preferences are fixed. Classical decision analysis methods such as multi-attribute utility theory can be applied if decisions are isolated to an individual actor. Non-cooperative game theory can be applied to interactive decisions among separate actors to identify individual solutions based on expected interactions from others. Joint decisions among collective actors, however, rely on co-construction of knowledge and negotiation for solutions.

Infrastructure systems design lies between interactive and joint decisions and between separate and collective actors. Framed as a system-of-systems, each actor has some, but not complete, independent control over their constituent system. Decisions are *at least* interactive based on resource interdependencies and *at most* joint under centralized control. Furthermore, while the design objective of sustainability relies partly on natural phenomena, it is heavily influenced by human preferences. The following sections present insights from three fields—collaborative engineering, negotiation, and integrated assessment—for supporting collaborative design activities.

7.1.1 Insights from Collaborative Engineering

Collaborative engineering is a field of study which "facilitates the communal establishment of technical agreements among a team of interdisciplinary stakeholders, who work jointly toward a common goal with limited resources or conflicting resources" (Lu et al. 2007). It spans existing disciplines including organizational science (how individuals and groups behave in an organization), social cognition (how individuals perceive, influence, and relate to others), social choice (how individual preferences can contribute to a consistent group preference), and decision science (how to make rational and optimal decisions) to apply collaborative sciences to engineering practice.

Lu et al. (2007) propose a research hypothesis of engineering collaboration via negotiation (ECN). It outlines a four-step process by which collaborative design is achieved:

1. *Manage interactions*: social interactions change individuals' perspectives of the design task.
2. *Construct understanding*: changing perspectives, if properly managed, leads to a common understanding of the design task.
3. *Discourse preference*: a common understanding of the design task anchors discussion and allows for comparison of individual preferences to form a group preference.
4. *Attain agreement*: a consistent group preference allows individual decisions to contribute to a collaborative group decision.

The ECN hypothesis aligns with findings from an observational study of practitioners at the ESA Concurrent Design Facility (CDF) (Kolfschoten et al. 2012). It identifies 22 guidelines for collaboration support in 7 categories in Table 7.2, where category 7 manages interactions, 1–4 construct understanding, 5 discourses preferences, and 6 attains an agreement.

7.1.2 Insights from Negotiation

Negotiation is a structured process to make joint decisions among two or more parties. Klein et al. (2003) discuss the use of models in mediated single-text negotiation. Under this type of negotiation, a mediator proposes an initial contract which is critiqued and iterated upon by participants to generate improved contracts. In most realistic design contexts, a model of a contract includes too many states to exhaustively evaluate, for example a sample problem with 100 boolean-valued issues has about 10^{30} possible states. Without the ability to sample widely in the contract space, sequential negotiation may get stuck in local extrema.

Klein et al. compare two approaches for decisions: a hill-climbing agent only accepts better contracts in each round while an annealing agent can accept worse contracts with a certain probability. A dilemma arises when considering combinations of agents in a multi-actor decision. Two annealing agents result in better outcomes for both parties as compared to two hill-climbers; however a mixed team results in better outcomes for the hill-climbing agent compared to its annealing counterpart. To avoid issues in detecting the behavior of agents (i.e. identifying hill-climbing), the authors propose an annealing mediator with greater ability to follow acceptability of contracts as indicated by agents. The mediator serves as collaboration support to avoid dominant individual strategies in conflict with one which maximizes social welfare.

7.1.3 Insights from Integrated Assessment

Integrated assessment (IA) is "a structured process of dealing with complex issues, using knowledge from various scientific disciplines and/or stakeholders, such that integrated insights are made available to decision makers" (Rotmans 1998). IA employs both analytical and participatory methods. Analytical methods draw from natural sciences to use model, scenario, and risk analyses to represent and structure scientific knowledge. Participatory methods draw from social sciences to use

Table 7.2: Guidelines for collaboration support in Kolfschoten et al. (2012)

Guideline Category	Detailed Guidelines
1. Share and generate knowledge	Incorporate users, stakeholders, and multi-disciplinary teams Foster reflection and experience sharing among stakeholders Use brainstorming tools with multiple perspectives Use a shared file space with version control and access rights
2. Distill the important information	Keep feasible alternatives but describe concisely, remove redundancy, and make choices via evaluation
3. Clarify for shared understanding	Use rich media and shared workspaces Separate content from meta-discussion
4. Organize to reduce complexity	Use visualizations to create and explain relationships Use flexible and modular visualization to adapt to new ideas Use framework and structure to organize information
5. Evaluate and compare alternatives	Evaluate in rounds and analyze in between Document key considerations Separate preferences from quality assessment Use multi-criteria decision matrices for group assessment
6. Build consensus and commitment	Create clear rules for decision-making Invite critique on design alternatives Discuss proposals to move from exploration to choice
7. Coordinate team efforts	Facilitate turn-taking and joint editing Create a clear view of goals, deliverables, agenda, and roles Share personal information for trust-building Explore personal motivation for participation Create a team bond to sustain relations and commitment

expert panels, Delphi methods, gaming, policy exercises, and focus groups to involve non-scientists as stakeholders. Rotmans (1998) describes two approaches to combine methods: a supply-driven IA activity uses analytical methods in anticipation of social relevance which are enriched with participatory methods, while a demand-driven IA activity uses participatory methods to address relevant problems and to determine supporting analytical methods.

In part to bridge the two approaches, de Kraker et al. (2011) describes an activity called participatory integrated assessment (PIA) which emphasizes "social learning of stakeholders, that is, a process of reframing and convergence of their perspectives on the problem and possible solutions." The authors show computer models support social learning by linking choices with consequences in a feedback loop and providing a platform and structure for negotiation. However, the authors find in most cases the computer model failed to play a significant role due to limited stakeholder acceptance, insufficient time to complete the feedback loop, poor user-friendliness, high model complexity, and model inflexibility to incorporate issues of interest to stakeholders. Alternative approaches of mediated modeling and companion modeling may improve model salience, legitimacy, and credibility by stronger stakeholder involvement, but are time-consuming and resource-intensive to implement.

7.1.4 Research Objective

Collaborative engineering literature identifies processes and guidelines to improve outcomes of collaborative design activities. Interoperable simulation gaming provides a mechanism for information exchange to change designers' perspectives, constructs a common understanding of causes of problems, and identifies possible options for response. Additionally, objective metrics as model outputs may contribute to preference discourse leading to agreements.

Negotiation literature identifies challenges to decision-making in complex design spaces due to self-interested designers. Hill-climbing approaches provide a dominant strategy for negotiation, but may get stuck in local extrema, resulting in inferior decisions. A third party mediator may be needed to encourage participants to make temporary concessions and explore new portions of the design space.

IA literature recognizes models as a tool for social learning provided sufficient time for feedback loops and involvement of stakeholders. Interoperable simulation gaming is an activity similar to participatory IA as players collectively build a model of future infrastructure systems from individual elements.

Combining these insights, this study poses an exploratory research question previously stated in Chapter 2 to investigate collaborative design activities using the prototype SIPS-G application:

3. What design-in-the-small elements of an interoperable simulation game can lead to improved design activity outcomes?

Based on background literature in related areas, the following hypotheses are formulated:

H1. Displaying an objective metric quantifying a consistent group preference leads to improved design activity outcomes.

H2. Integrated, distributed simulation tools enable more frequent information exchanges and lead to improved design activity outcomes.

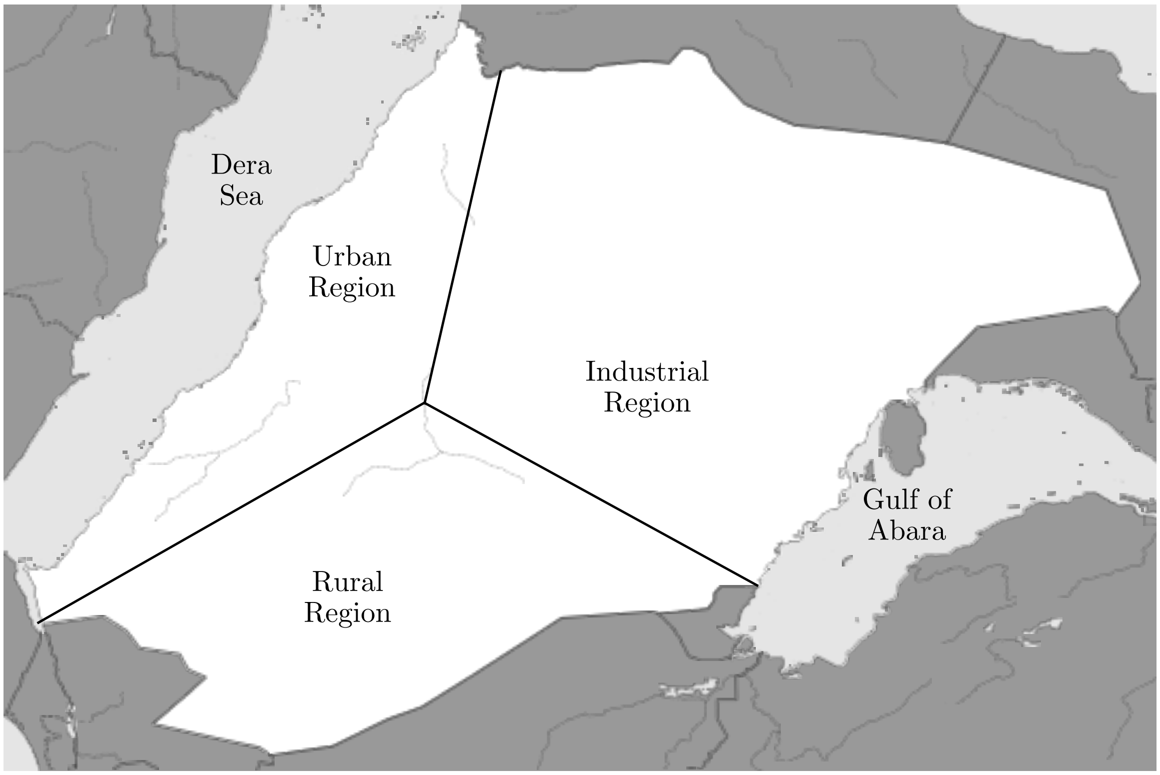

Figure 7-1: Geographic layout of the fictional country Idas Abara with Industrial, Rural, and Urban regions. Map credit: Wikimedia user NordNordWest.

This study proposes a controlled human subjects experiment to evaluate these hypotheses using multiple tool variants as experimental conditions. This study does not directly compare the SIPS-G application with another design tool to measure effectiveness of a simulation gaming approach. While potentially interesting, there are no comparable tools which would allow a reasonable comparison. Instead, it only seeks to evaluate the effect of factors which are enabled by the approach.

7.2 SIPS-G Design Scenario

The SIPS-G design scenario is based on the context of Saudi Arabia developed in Chapter 6, but purposefully fictionalized to allow for greater freedom from existing mental models which may not be represented in the simplified system models. This section introduces the design scenario and formulates objectives for each role player.

7.2.1 Scenario Overview

The scenario centers on the fictional country of Idas Abara as a desert nation seeking to develop a sustainable infrastructure plan. Geographic depictions use a transformed (vertically-reflected and horizontally-stretched) map of Saudi Arabia shown in Figure 7-1. The three regions—industrial, rural, and urban—correspond to the same regions identified in Chapter 6.

Table 7.3: Infrastructure player roles and responsibilities

Player Role	National Control	Similar Ministries	Local Control
Ministry of Agriculture (MoA)	$\mathbf{E}_{agricul.}$	MoA	$\{\mathbf{E}^{urban}_{agricul.}, \mathbf{E}^{rural}_{agricul.}, \mathbf{E}^{industrial}_{agricul.}\}$
Ministry of Water (MoW)	$\mathbf{E}_{water}$	MoWE[1]	$\{\mathbf{E}^{urban}_{water}, \mathbf{E}^{rural}_{water}, \mathbf{E}^{industrial}_{water}\}$
Ministry of Energy (MoE)	$\mathbf{E}_{petrol.}, \mathbf{E}_{elect.}$	MoPMR, MoWE[2]	$\{\mathbf{E}^{urban}_{petrol.}, \mathbf{E}^{rural}_{petrol.}, \mathbf{E}^{industrial}_{petrol.}, \mathbf{E}^{urban}_{elect.}, \mathbf{E}^{rural}_{elect.}, \mathbf{E}^{industrial}_{elect.}\}$

[1] Water functions only
[2] Electricity functions only

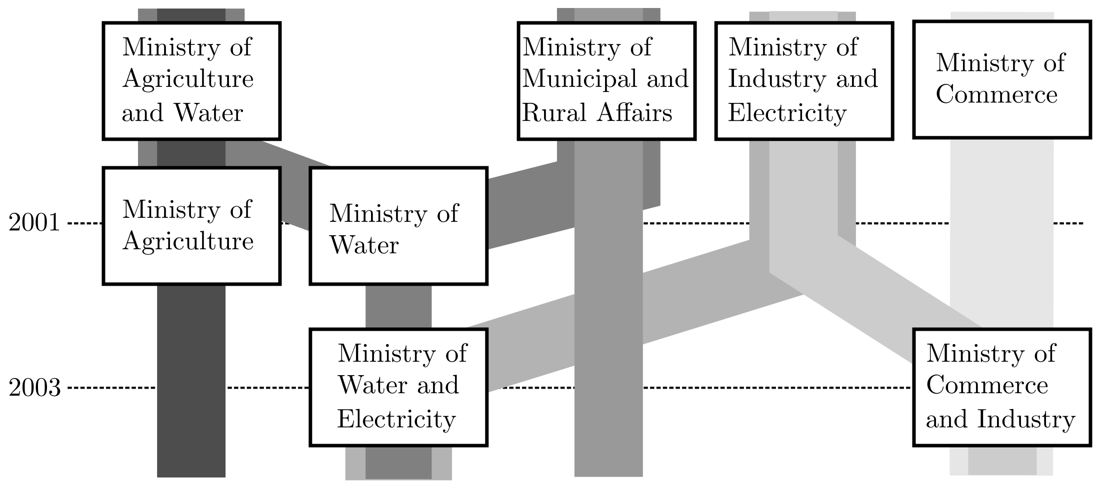

Figure 7-2: The Ministry of Water and Electricity is historically rooted in the Ministry of Agriculture and Water, Ministry of Municipal and Rural Affairs, and the Ministry of Industry and Commerce.

Participants represent an advisory council of three infrastructure ministries. Each role is assigned control over at one or two national infrastructure sectors in Table 7.3. Player roles approximate functions of government ministries, with activities coordinated by equivalents to the Ministry of Economy of Planning (MoEP) and the Ministry of Municipal and Rural Affairs (MoMRA). The agricultural role, similar to the Ministry of Agriculture (MoA), manages arable land allocation for domestic food production. The water role, similar to the water portion of the Ministry of Water and Electricity (MoWE), manages infrastructure to produce and distribute water. Finally, the energy role, similar to the Ministry of Petroleum and Mineral Resources (MoPMR) and the electricity portion of MoWE, manages infrastructure to produce and distribute oil and electricity.

While both the water and energy players share some functions of the present-day MoWE, this is a relatively new organizational unit in Saudi Arabia. Figure 7-2 illustrates the recent history of several Saudi ministries. The former Ministry of Agriculture and Water and the Ministry of Municipal and Rural Affairs merged water functions to create the Ministry of Water in 2001 (MoWE 2013). The former Ministry of Industry and Electricity transferred electricity functions to the Ministry of Water to create the Ministry of Water and Electricity in 2003 (MCI 2013). Therefore, the decision to group the electricity sector with the energy role more closely follows the historical perspective on promoting industry and commerce.

Table 7.4: Summary of available infrastructure element templates

Role	Element	Capital Expense
Agriculture	Large Wheat Farm	§180M/yr for 1 year
	High-volume Food Transport	§300M/yr for 1 year
Water	Large RO Desalination Plant	§250M/yr for 3 years
	Huge RO Desalination Plant	§1000M/yr for 3 years
Energy	Large Thermal Power Plant	§75M/yr for 3 years
	Large Solar PV Power Plant	§450M/yr for 3 years
	Large Oil Well	§975M/yr for 3 years
	High-volume Oil Pipeline	§225M/yr for 3 years

The planning scenario takes place in the year 1980, providing historical context from the prior 30 years (1950–1980) using the baseline scenario developed in Chapter 6. The design session develops a plan for infrastructure investment for the following 30 years (1980–2010) describing which elements to create, where they should be placed, and when they are to operate. Each player seeks to maximize an individual objective describing their role's preferences and a common national objective, both detailed in the following sections. As a constraint to meet these objectives, the total annual national capital expenditures are limited by

$$\sum_i P_{capital}(\mathbf{E}_i, t) < B_{limit} \; \forall \; t \tag{7.1}$$

where $B_{limit} =$ §4 billion per year. There is no limit on the operational expenses within any sector.

There are a number of targeted simplifications to allow non-experts to participate in design sessions on the order of 60 minutes. First, each player is limited to two infrastructure templates within each sector summarized in Table 7.4. The MoA can commission large wheat farms and high-volume food transport, the MoW can commission large or huge desalination plants, and the MoE can commission large oil wells, high-volume oil pipelines, and large thermal or solar photo-voltaic power plants. Second, most element-level parameters are not displayed in the simulation tool and the detailed formulation of score components is not disclosed.[1] Instead, participants may ask the facilitator for any desired supporting technical information during the design session.

7.2.2 Design Objectives

Explicit objective metrics are formulated for each player based on their ministry's role and for the overall national objective. Objective metrics use cumulative stocks or average flows to prevent manipulation of boundary conditions in a game setting. For example, resource security terms use an average value over the future scenario simulation (1980–2010) to discount high resource security in a single final year. Similarly, capital investment and net revenue terms are a function of cumulative stock values at the end of the simulation. Appendix C provides a detailed description and mathematical formulation of each objective metric and an analysis of interactions between

[1]Pilot testing found most players became overwhelmed if model details were provided. When omitted, players were able to focus on the bigger challenge of planning in a short-duration session.

components.

The agriculture role's objective metric ($\mathbf{J}_{MoA}$) includes three components of equal weight:

1. *Food Security* ($\mathbf{S}_{food}$): fraction of food supply from domestic sources.

2. *Capital Investment* ($\mathbf{I}_{MoA}$): cumulative capital investment in the agriculture sector.

3. *Net Revenue* ($\mathbf{R}_{MoA}$): cumulative net revenue from the agriculture sector.

The water role's objective metric ($\mathbf{J}_{MoW}$) includes three components of equal weight:

1. *Aquifer Security* ($\mathbf{S}_{aquifer}$): expected aquifer lifetime at current withdrawal rates.

2. *Capital Investment* ($\mathbf{I}_{MoW}$): cumulative capital investment in the water sector.

3. *Net Revenue* ($\mathbf{R}_{MoW}$): cumulative net revenue from the water sector.

The energy role's objective metric ($\mathbf{J}_{MoE}$) includes three components of equal weight:

1. *Reservoir Security* ($\mathbf{S}_{reservoir}$): expected oil reservoir lifetime at current extraction rates.

2. *Capital Investment* ($\mathbf{I}_{MoE}$): cumulative capital investment in oil and electricity sectors.

3. *Net Revenue* ($\mathbf{R}_{MoE}$): cumulative net revenue from the oil and electricity sectors.

Finally, the national objective metric ($\mathbf{J}_{IA}$) includes four components of equal weight:

1. *Food Security* ($\mathbf{S}_{food}$): previously described in the agriculture objective.

2. *Aquifer Security* ($\mathbf{S}_{aquifer}$): previously described in the water objective.

3. *Reservoir Security* ($\mathbf{S}_{reservoir}$): previously described in the energy objective.

4. *Net Revenue* ($\mathbf{R}_{IA}$): cumulative net revenue of all infrastructure sectors.

Table 7.5 summarizes anticipated couplings between infrastructure decisions and objectives where more symbols represent greater impact. Investment by each player positively contributes to their own objectives with the strongest effect for agriculture and weakest for water. The agriculture-water tension contributes a strong negative effect between agriculture investment and water objectives. The water-energy budget tension contributes a moderate negative effect between water investment and energy objectives and between energy investment and water objectives. Agriculture contributes less budget tension with a slight negative effect with water and energy. Overall, individual investments have a mixed effect on the national objective depending on levels of investment and interactions across roles.

Table 7.5: Anticipated couplings between investments and objectives

Infrastructure Decision	$\mathbf{J}_{MoA}$	$\mathbf{J}_{MoW}$	$\mathbf{J}_{MoE}$	$\mathbf{J}_{IA}$
Wheat Fields	$+++$	$---$	$-$	$+/-$
Desalination Plants	$-$	$+$	$--$	$+/-$
Power Plants, Oil Wells	$-$	$--$	$++$	$+/-$

7.2.3 Assumptions and Limitations

Limitations in the SIPS-G design scenario arise from the assumption that participants have no background experience in infrastructure planning within their assigned role. First, this limits the quantity and detail of model information which can be presented and interpreted. As discussed in the overview, each sector only has two available infrastructure templates to consider, and few data are directly displayed. Second, the lack of familiarity and experience requires an external formulation of objectives rather than relying on internalized preferences for a role.

The individual and national objectives formulated above are a highly simplified set of metrics selected for this design scenario. They assign equal weight to each component to simplify their interpretation by participants. A future extension may allow user control over weightings of components; however this may limit the direct comparison of results across design sessions. More realistic applications would likely draw from a larger set of metrics to address important components of sustainability including economics, social equity, and environmental impact. While the SIPS-G objectives capture a portion of these factors from an infrastructure-oriented perspective, others to be considered include carbon emissions, balance of investment across regions, equity of resource access, agricultural contributions to local economies, and stability of budget and revenue.

7.3 Experimental Methodology

This section outlines an experimental methodology to evaluate the effect of SIPS-G tool variations on outcomes in collaborative design between player roles. The following sections discuss the experimental design, procedure, and limitations.

7.3.1 Experimental Design

This study proposes a controlled human subjects experiment using the SIPS-G scenario as a collaborative design task. The experiment is structured as a between-subjects study with a design session as the unit of analysis. Testing varies three experimental variables—mode of data exchange, layout of design stations, and form of national objectives—to create three experimental conditions. Each variable is discussed in detail below.

The data exchange mode determines how technical information is communicated between player roles. Figure 7-3 illustrates two modes used in this study. The synchronous mode leverages interoperable simulation using the HLA as described in Chapter 5. In this format, each subject has local control over simulation inputs within their respective role; however all three subjects must synchronously run a distributed simulation to update outputs. During a simulation execution the component models exchange data at each time step using the iterative approach to resolve dependencies across system boundaries to minimize discrepancies.

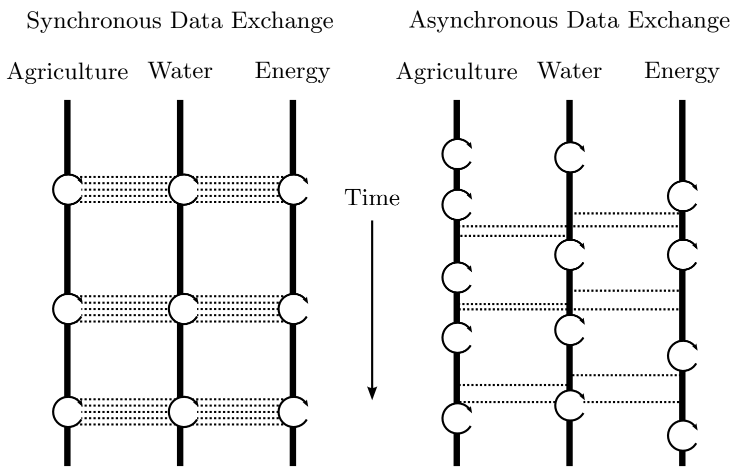

Figure 7-3: The synchronous mode (left) uses dynamically exchanges data during integrated simulation executions. The asynchronous mode (right) uses files to exchange static data between simulation executions. Dashed lines illustrate data exchanges and circular arrows represent simulation executions.

The asynchronous mode of data exchange uses an augmented design tool relying on static files as input/output from tools. Rather than performing dynamic data exchanges during a simulation execution, the resource flows for each time step are saved in a data file which is transferred between design stations using a shared network folder. In this format, subjects retain local control over simulation inputs, execution, and outputs; however there may be moderate discrepancies between players due to data dependencies. For example, to observe the effect of agricultural food production on national net revenue requires three simulations with intermediate file exchanges: one for the effect of agricultural production on water demand, a second for the effect of water demand on water supply, and a third for the effect of water supply on net revenue.

In addition to differences in operational data exchange, the physical layout of design stations varies for the two modes. Figure 7-4 compares the centralized layout (top) for the synchronous mode with the distributed layout (bottom) for the asynchronous mode. While both conditions have a central table for initial briefing, the distributed layout has more isolated design stations to approximate barriers to collaboration. There are no other limitations on the design process for the distributed layout aside from the location of design stations.

Finally, national objectives are conveyed in two forms. The descriptive form qualitatively explains the four components (food security, aquifer security, reservoir security, and national net revenue) during overview materials. The quantitative form includes the description during introductory materials and incorporates a national objective metric display within the tool. Figure 7-5 compares available displays in the tool under these two conditions.

Due to the relatively high cost of scheduling volunteers in the sampling frame, this experimental design calls for sessions in three conditions to measure the effect of alternative tools rather than a factorial design to measure the effect of each variable. Table 7.6 lists three conditions combining the factors of data exchange mode, design station layout, and national objective form.

Figure 7-4: Comparison of centralized (top) and distributed (bottom) design stations.

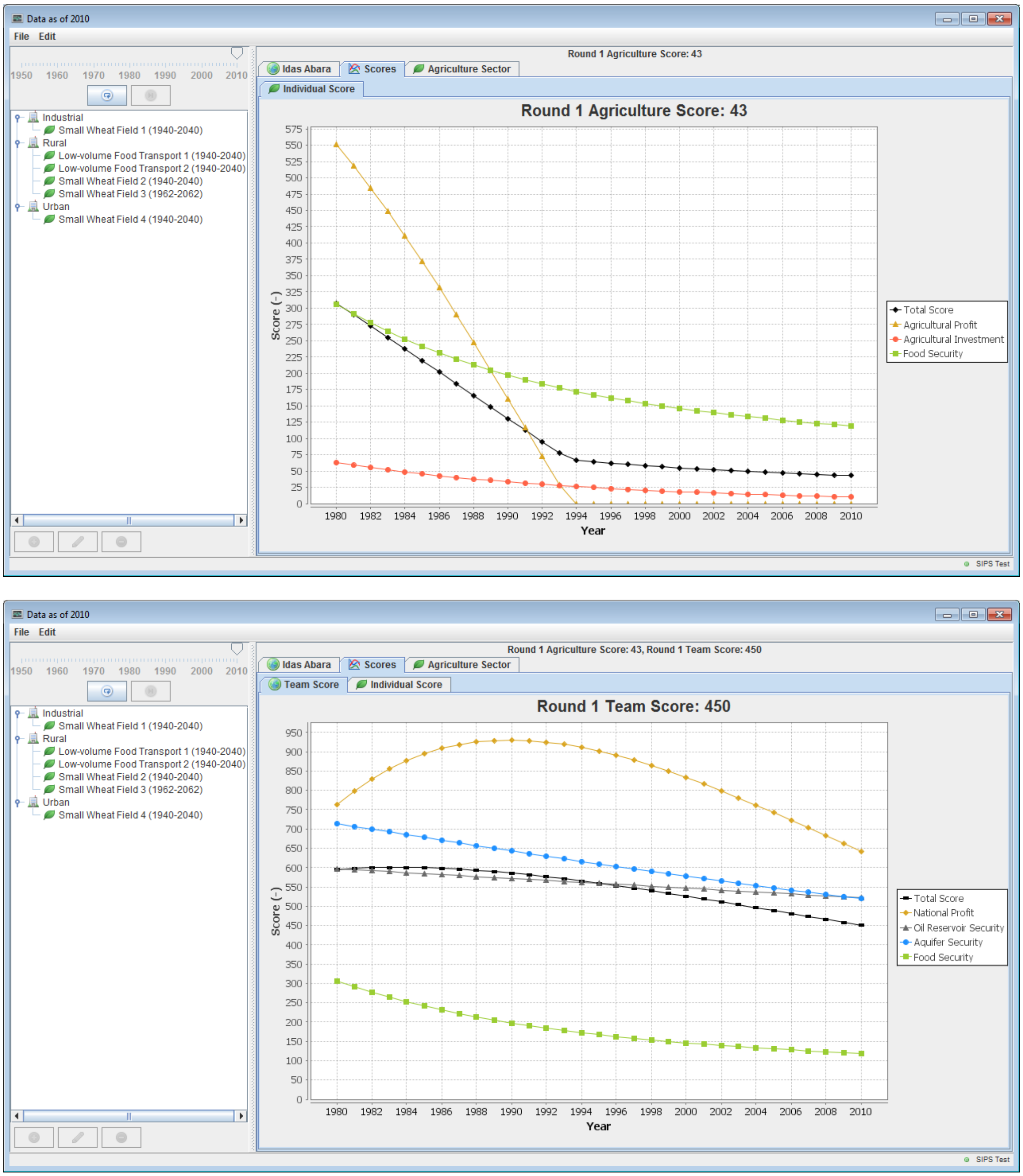

Figure 7-5: Comparison of tools for descriptive (top) and quantitative (bottom) national objectives. The quantitative form displays a “team score” and time series chart alongside the individual score.

Table 7.6: Comparison of SIPS-G experimental conditions

Condition	Exchange Mode	Design Station Layout	National Objective
Variant 1A	Synchronous	Centralized	Quantitative
Variant 1B	Synchronous	Centralized	Descriptive
Variant 2	Asynchronous	Distributed	Descriptive

Table 7.7: SIPS-G subject demographics in 15 sessions

		Variant				
Category	Value	1A	1B	2	Tot.	(%)
Gender	Male	11	8	10	29	64.4
	Female	4	7	5	16	35.6
Age	18–24	3	2	1	6	13.3
	25–29	9	13	10	32	71.1
	30–34	3	0	4	7	15.6
	35–39	0	0	0	0	0.0
	40–49	0	0	0	0	0.0
	50+	0	0	0	0	0.0
Years of college education in a technical field	0	0	0	0	0	0.0
	1–2	0	0	0	0	0.0
	3–4	2	1	0	3	6.7
	5–6	1	3	4	8	17.8
	7–8	7	4	5	16	35.6
	9+	5	7	6	18	40.0

		Variant				
Category	Value	1A	1B	2	Tot.	(%)
Years of professional work experience in a technical field	0	3	3	6	12	26.7
	1–2	9	9	5	23	51.1
	3–4	1	3	2	6	13.3
	5–6	1	0	2	3	6.7
	7–8	1	0	0	1	2.2
	9+	0	0	0	0	0.0
Frequency of past interactions with other subjects	Never	19	16	18	53	58.9
	Once	3	1	0	4	4.4
	Rarely	0	5	3	8	8.9
	Monthly	1	1	0	2	2.2
	Weekly	5	5	8	18	20.0
	Daily	2	2	1	5	5.6

7.3.2 Experimental Procedure

15 groups of 3 subjects participated in this study under an IRB-approved protocol. Volunteers were recruited from a convenience sample of peers in graduate programs at MIT and were not paid for their efforts. Subjects were predominately male (64.4%) and 25–29 years of age (71.1%) with more college education than work experience in technical fields. Most subjects had never interacted with each other in the past (58.9% of pairs), although a significant fraction (20.0%) interact on a weekly basis. Table 7.7 summarizes the complete subject demographics. By inspection, there are no significant demographic differences among the three experimental conditions.

Design sessions are scheduled when three volunteers are available to form ad-hoc groups. While there is no random assignment of subjects to sessions, there is also no purposeful selection. Conditions are assigned in partially-randomized order with sessions 1–8 randomly assigned Variant 1A or 1B and sessions 9–16 randomly assigned Variant 2 or (remaining) 1B. All experiments are conducted in university classrooms using wireless network connections. At the start of the session subjects are assigned a color (red, green, or blue) and seated on one side of a rectangular design station with the fourth seat reserved for the facilitator. All sessions were facilitated by the same researcher. During the design session, subjects remain at the central design station under Variant

1 (A or B) or move to separate design stations under Variant 2 to assume the roles of agriculture (green), water (blue), or energy (red).

Each session is conducted using a standard procedure. Participants may exit the study at any point, however no such events occurred. A 15-minute scripted presentation introduces the SIPS-G design context including the three regions (industrial, rural, urban), infrastructure within each sector, resource interdependencies, operational rules built into the simulation model, other high-level assumptions relating to price and cost, budget and time constraints, and a description of national objectives. Subjects also receive a confidential material sheet describing individual objectives and an overview of key issues in their respective sector. Participants may either share the confidential information or keep it to themselves. Next, a 15-minute tutorial introduces the subjects to the software tool including simulation inputs (existing elements and available templates), execution control buttons (initialize and run), and a walk-through of all output screens.

After completing the overview and tutorial and addressing any related questions, subjects immediately enter the 60-minute timed design session period. Software logs store infrastructure decisions before each simulation run and an audio recorder captures verbal conversation. Subjects are allowed to move about the room and share their display during the design session but may not move the design stations themselves. Subjects can also ask the facilitator for any additional information not displayed in the software tool, clarifications on model assumptions, or other questions excluding advice on design decisions. The facilitator updates the remaining time at several points during the session. Following the design task the facilitator leads a de-briefing session to explain the study objectives and explore experiences and observations from the design session.

7.3.3 Limitations and Threats to Validity

This study has several limitations which pose threats to the validity of results. First, it does not employ a full-factorial design of the three independent variables (data exchange mode, design station layout, and national objective form). Rather, it only evaluates the three tool variants (1A, 1B, and 2), limiting the ability to distinguish between the effect of coupled variables. Additionally, group processes are largely uncontrolled during design sessions. Subjects are not constrained to follow a particular process for design, nor are there limits on discussion or sharing of information. Furthermore, there are no imposed preferences for individual versus national objectives. This lack of control introduces additional variation beyond tool variants which may limit conclusions.

This design does not fully leverage randomization of conditions for practical reasons. Groups are formed as subjects' schedules allow rather than random assignment of subjects. Potential biases are partially mitigated by the non-purposeful assignment of conditions to sessions which are randomly assigned except for Variant 2 which is limited to the second half of sessions. The ordering effect may bias results due to facilitator maturation effects and is partially mitigated by adhering to a common scripted introduction and tutorial across all sessions.

Researcher participation as the facilitator in all design sessions introduces additional potential biases, especially as subjects are sampled from peer groups of the researcher. Furthermore, the researcher is the developer of the software tool. Scripted introduction and tutorial materials and passive facilitator participation only interacting in response to direct questions address a portion of these concerns, however the possibility of additional biases must be acknowledged.

Several factors limit the generalizability of results beyond the design sessions considered. Previously discussed limitations in the SIPS-G model and design scenario limit direct extensions of results to more realistic situations. Similarly, the sample of subjects is not representative of infrastructure planners, although their backgrounds in technical areas may be similar. There are also numerous potential reactive effects of experimental arrangements. First, subjects participate in ad-hoc teams and are not required to have background experience in the design domain of infrastructure systems. Second, design sessions are conducted in general-purpose classrooms using unfamiliar software tools. A large portion of the design time may be required to simply understand the task. Finally, subjects participating in a finite, clearly fictional session may not fully consider the implications of decisions having great socio-economic impact in the real world.

7.4 Results and Analysis

Table 7.8 summarizes results after the final round for 15 experimental sessions sorted by experimental condition and re-labeled session number. All results are post-processed from an aggregated model incorporating designs from all players to address potential discrepancies introduced in the asynchronous data exchange mode. Expanded results are available in Appendix D.

Six sessions violate budget constraints in one or more years, however this alone does not affect numerical results of objective metrics. Most budget violations are small and isolated to a few years which suggests they could be alleviated by adjusting planning schedules while achieving nearly identical objective metrics. In one particular outlying case, session 9, inspection shows sufficient budget capacity in adjacent years. Thus, the over-budget condition is considered indicative of time pressure limiting the actions of the designers.

Caution must be used when approaching analysis of these results, particularly for objective metrics which, strictly speaking, are ordinal scale similar to most applications of utility theory. Nonparametric statistical tests including median measure of central tenancy, Spearman correlation, the Mann-Whitney-Wilcoxson rank sum test, and the Kruskal-Wallis test are used in these analyses.

7.4.1 Validation of Expected Correlations

Table 7.9 inspects Spearman correlations between infrastructure inputs measured by final production capacity and objectives to validate the anticipated effects in Table 7.5. These results are only simple correlations and do not consider non-linear behaviors and other confounding effects such as coincident infrastructure decisions. The limited statistical significance for most correlation values reflect these known limitations.

As expected, infrastructure investment has a positive effect on the individual objective for all three roles with the weakest effect for water. Also as expected, some negative correlations exist between several infrastructure-objective pairs corresponding to agriculture–water and water–energy couplings. Overall, the national objective is negatively correlated with agriculture and water infrastructure and positively correlated with energy infrastructure.

7.4.2 Analysis of Outcome Variables

Figure 7-6 shows a box plot of final agriculture, water, energy, and national objective metrics under the three tool variants. The Kruskal-Wallis test assesses a null hypothesis that data in more than two groups are samples from the same distribution. The null hypothesis cannot be rejected for

Table 7.8: Summary of SIPS-G outcome results by session

Session	Variant	Budget	Wheat EJ/yr	$\mathbf{J}_{MoA}$	Desal. MCM/yr	$\mathbf{J}_{MoW}$	Oil Mtoe/yr	Power TWh/yr	$\mathbf{J}_{MoE}$	$\mathbf{J}_{IA}$
S–1	2	Under	85.0	711.8	10250	344.0	1200	234	778.6	514.2
S–2	2	Under	130.0	950.4	11900	340.4	500	94	600.2	349.7
S–3	2	Under	75.0	469.5	14300	378.4	500	234	710.0	449.3
S–4	2	Under	102.5	489.6	3200	367.4	800	234	655.0	467.3
S–5	2	Under	210.0	936.0	3650	345.2	700	234	742.7	349.9
S–6	1B	Under	90.0	662.8	5300	344.2	1700	204	779.5	509.9
S–7	1B	Over[7]	90.0	613.4	2600	312.1	600	164	688.9	484.4
S–8	1B	Over[8]	95.0	654.1	11600	347.8	500	204	722.4	466.9
S–9	1B	Over[9]	75.0	364.3	11600	400.0	450	294	584.4	438.5
S–10	1B	Under	130.0	794.7	7850	353.0	375	150	663.7	505.8
S–11	1A	Over[11]	150.0	624.6	10850	352.2	700	114	602.8	344.8
S–12	1A	Over[12]	140.0	657.7	8400	355.2	1100	244	786.9	497.9
S–13	1A	Under	80.0	401.1	7400	384.0	500	74	578.4	445.1
S–14	1A	Under	100.0	736.3	6200	342.7	800	184	724.4	517.1
S–15	1A	Over[15]	100.0	570.7	10250	359.5	1000	264	792.3	486.9
		Min	75.0	364.3	2600	312.1	375	74	578.4	344.8
		Median	100.0	654.1	8400	352.2	700	204	710.0	467.3
		Max	210.0	950.4	14300	400.0	1700	294	792.3	517.1

[7] §885 million (22%) in 1990
[8] §160 million (4%) in 1982, §530 million (13%) in 1986
[9] §450 million (11%) in 1990, §2.5 billion (63%) in 1991, §300 million (8%) in 1992, §1.0 billion (25%) in 1995–1996, §230 million (6%) in 2000, §300 million (8%) in 2001–2002
[11] §360 million (9%) in 1980, §610 million (15%) in 1990, §250 million (6%) in 1991
[12] §360 million (9%) in 1982, §775 million (19%) in 2000
[15] §275 million (7%) in 1999, §485 million (12%) in 2000, §250 million (6%) in 2004, §530 million (13%) in 2005

Table 7.9: Spearman correlation between infrastructure decisions and objective metrics

Infrastructure Capacity	$\mathbf{J}_{MoA}$	$\mathbf{J}_{MoW}$	$\mathbf{J}_{MoE}$	$\mathbf{J}_{IA}$
Wheat Farms (EJ/yr)	0.615*	-0.289	0.172	-0.174
Desalination Plants (MCM/yr)	-0.116	0.263	-0.172	-0.367
Oil Wells (Mtoe/yr)	0.182	-0.298	0.735**	0.514*
Power Plants (TWh/yr)	-0.289	0.388	0.516*	0.193

* $p < 0.05$
** $p < 0.01$

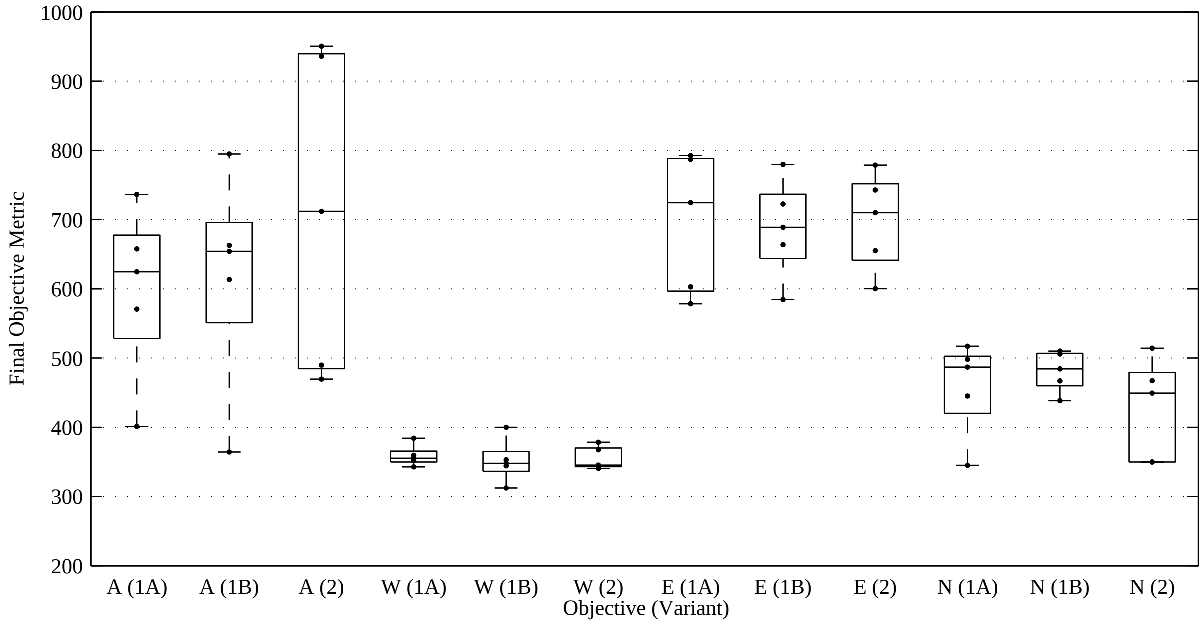

Figure 7-6: Box plot of SIPS-G results for agriculture (A), water (W), energy (E) and national (N) objectives by variant. Boxes bound the first and third quartiles and whiskers bound extremes within 1.5 times the interquartile range.

comparing national objective metrics between Variants 1A, 1B, and 2 ($\chi^2(2) = 0.78, p = 0.68$). Similarly, a multiple comparison test yields no significant differences among any variant pairs.

7.4.3 Analysis of Process Variables

Process variables are those observed and measured during the experimental session rather than outcome variables which are only measured at the end. Figures 7-7–7-8, for example, trace the evolution of objective metrics during each of the 15 design sessions on a tradespace of objectives. The actual tradespace is four-dimensional ($\mathbf{J}_{MoA} \times \mathbf{J}_{MoW} \times \mathbf{J}_{MoE} \times \mathbf{J}_{IA}$) but is represented in each plot with two dimensions where the set of Pareto efficient designs relative to the two objectives lies along the upper-right.

Figure 7-7 shows a positive correlation between agriculture and national objectives up until an inflection point around $\mathbf{J}_{MoA} = 600$ where achieving higher individual agriculture objectives reduces the national objective. This effect can be partially countered through increased investment in water infrastructure, as illustrated in several upward trajectories after initial drop in $\mathbf{J}_{IA}$.

Figure 7-8 shows a negative correlation between agriculture and water objectives. Most sessions follow a path of increasing $\mathbf{J}_{MoA}$ and decreasing $\mathbf{J}_{MoW}$ before transitioning to a path of increasing $\mathbf{J}_{MoW}$ and/or decreasing $\mathbf{J}_{MoA}$ as a concession between agriculture and water objectives.

Table 7.10 summarizes key process variables including number of data exchanges (N) in the session, maximum input change (Δ_{max}), and numbers of capacity increases (N_Δ^+) and decreases (N_Δ^-) for each of the four infrastructure sectors. The data exchange count is computed as the number of simulation executions using Variant 1 (A or B) or the number of file exchange periods using Variant 2. In most cases, all three subjects elected to exchange files around the same time.

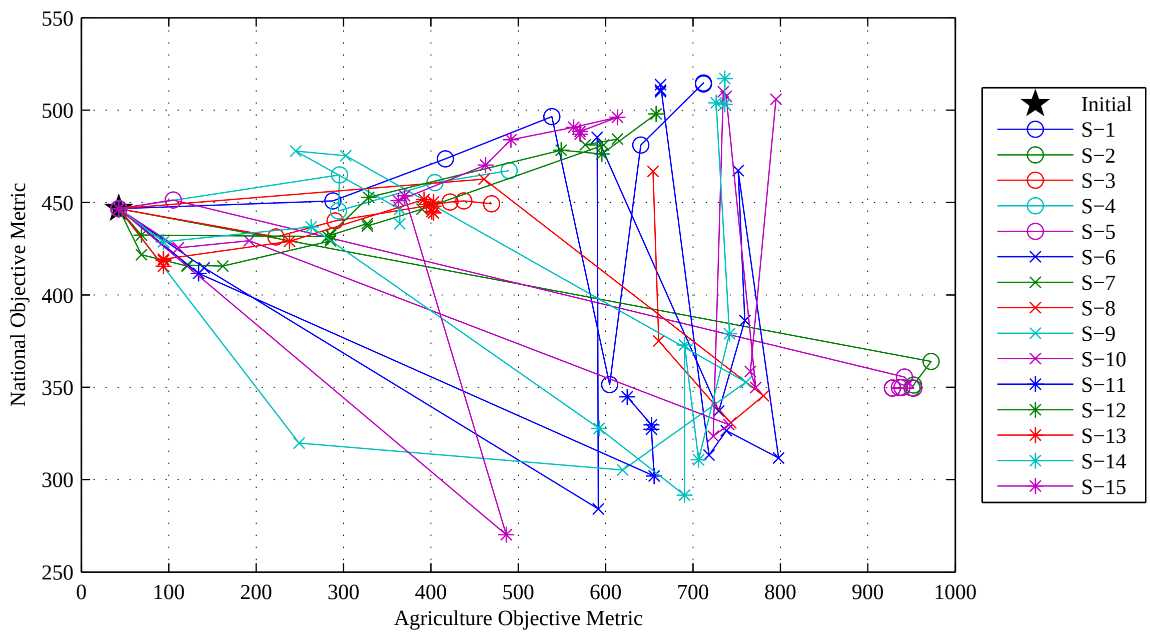

Figure 7-7: Design tradespace between agriculture and national objectives. Data show final objective metric values in successive rounds of the design session.

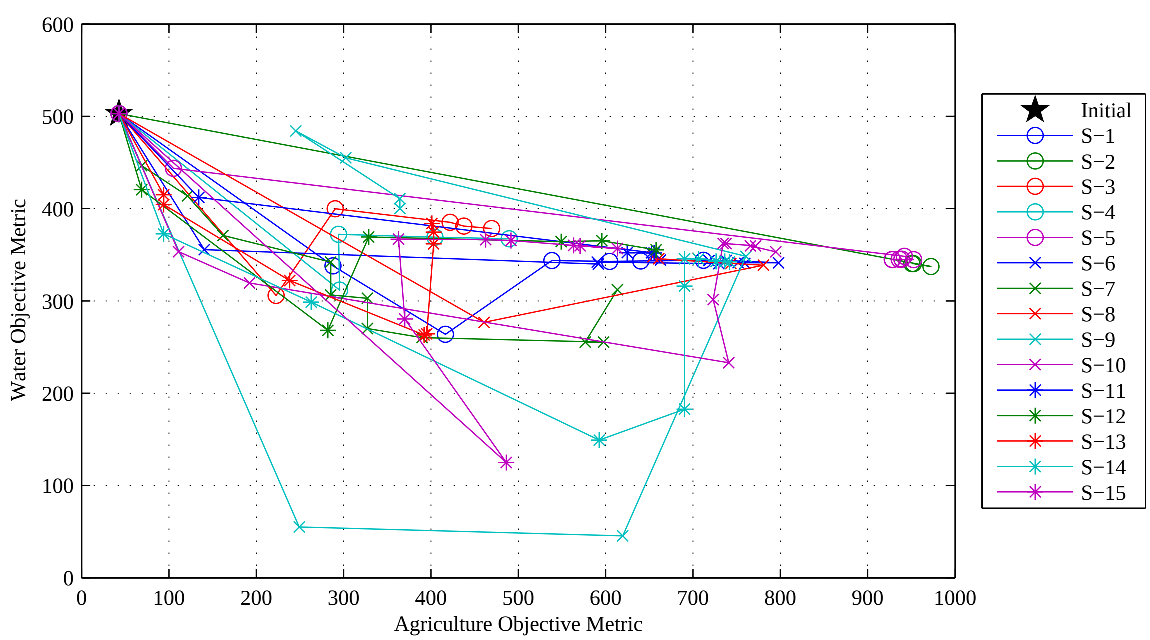

Figure 7-8: Design tradespace between agriculture and water objectives. Data show final objective metric values in successive rounds of the design session.

Table 7.10: Summary of SIPS-G process results by session

		Wheat (EJ/yr)			Desal. (MCM/yr)			Oil (Mtoe/yr)			Power (TWh/yr)			Energy
Session	N	Δ_{max}	N_{Δ}^{+}	N_{Δ}^{-}	Δ_{max}	N_{Δ}^{+}	N_{Δ}^{-}	Δ_{max}	N_{Δ}^{+}	N_{Δ}^{-}	Δ_{max}	N_{Δ}^{+}	N_{Δ}^{-}	N_{Δ}
S-1	7	35	5	1	3600	6	0	500	3	1	110	5	0	7
S-2	4	145	1	2	5700	3	0	0	0	0	70	1	0	1
S-3	5	25	4	0	9600	4	0	0	0	0	100	4	0	4
S-4	4	38	3	0	2400	2	0	300	1	0	110	3	0	4
S-5	7	150	5	1	3000	3	3	200	1	0	70	6	0	6
S-6	12	45	5	4	3600	6	5	400	6	0	50	6	0	8
S-7	13	25	8	1	750	6	0	100	1	0	50	4	0	5
S-8	5	50	3	2	4200	5	0	0	0	0	50	5	0	5
S-9	7	45	4	1	11250	3	1	50	0	1	170	6	1	7
S-10	9	105	4	3	4800	5	0	325	1	3	106	2	0	6
S-11	5	160	2	2	5400	4	0	100	2	0	40	3	0	4
S-12	6	60	5	1	4800	4	0	400	2	0	100	5	0	5
S-13	10	20	4	0	2550	7	0	0	0	0	20	3	0	3
S-14	11	35	5	1	1950	7	0	100	3	0	40	6	0	9
S-15	9	55	6	2	1800	9	0	200	3	0	60	7	0	9
Min	4	20	1	0	750	2	0	0	0	0	20	1	0	1
Median	7	45	4	1	3600	5	0	100	1	0	70	5	0	5
Max	13	160	8	4	11250	9	5	500	6	3	170	7	1	9

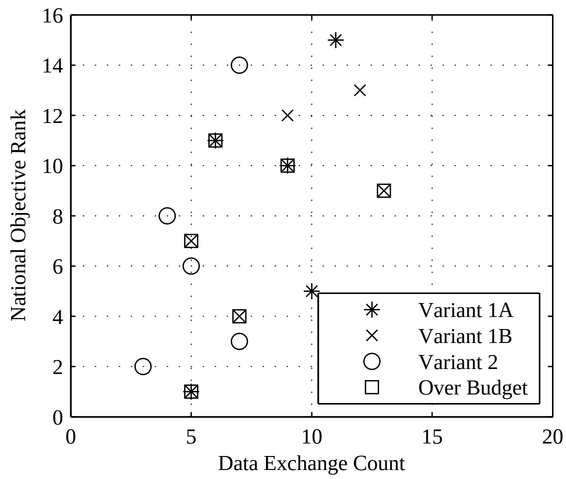

Figure 7-9: Scatter plot of SIPS-G ranked results by data exchange count. Better outcomes are correlated with higher exchange counts.

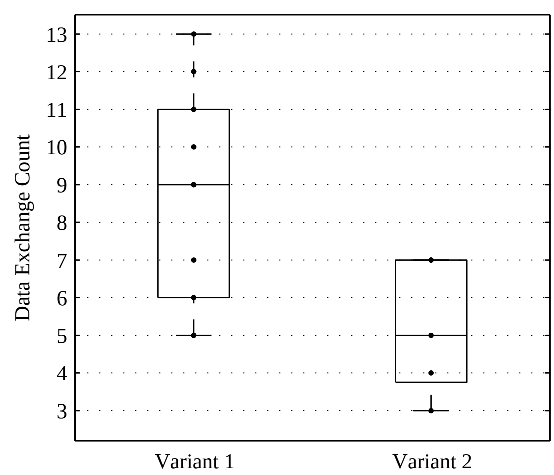

Figure 7-10: Box plot of data exchange count by tool variant. Boxes bound the first and third quartiles and whiskers bound extremes within 1.5 times the interquartile range.

The number of data exchanges as a measure of preference discourse is hypothesized to influence collaborative outcomes. The Spearman correlation between exchange count and national objective outcome is positive ($\rho = 0.544$) and significant ($p = 0.036$), illustrated in Figure 7-9 for ranked objectives. Furthermore, the mode of data exchange (Variant 1 or 2) has a significant impact on exchange count, $t(13) = -2.47$, $p = 0.028$), illustrated in Figure 7-10.

Similar results for the role of change counts on outcome metrics are observed within the energy role in Figure 7-11. The Spearman correlation between number of energy changes (N_Δ in Table 7.10) and the energy objective metric is positive ($\rho = 0.611$) and significant ($p = 0.016$). There is no significant difference in change counts between alternative data exchange modes (Variant 1 or 2) (ranksum 30, $p = 0.23$).

Maximum wheat expansion between data exchanges is inspected in Figure 7-12 as a possible predictor of national objective outcomes. While the Spearman correlation between maximum wheat expansion and national objective outcome is negative ($\rho = -0.360$), it is not statistically significant ($p = 0.188$). These results suggest a categorical relationship between large maximum wheat expansion and poor outcomes, rather than an ordinal relationship.

7.5 Discussion

This section discusses the results in terms of the hypothesized effects of design-in-the-small elements in the SIPS-G tool on collaborative design outcomes. Finally, it summarizes key implications for supporting collaborative design and outlines future work.

7.5.1 Evaluation of Hypotheses

This study varies the mode of data exchange, design station layout, and national objective form to create three variants of the SIPS-G tool. Hypothesis H1 proposed:

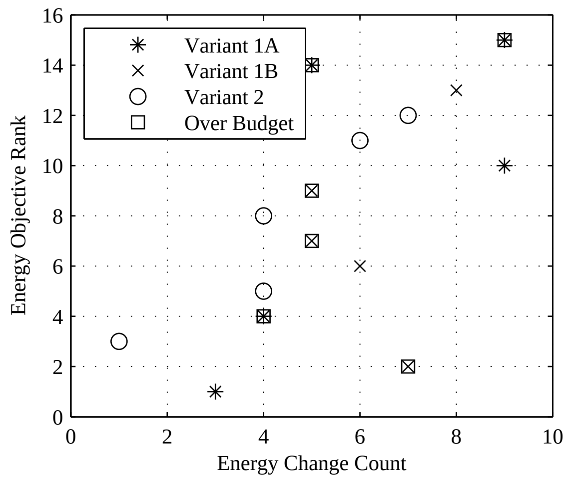

Figure 7-11: Scatter plot of SIPS-G ranked energy results by change count. Better outcomes are correlated with higher change counts.

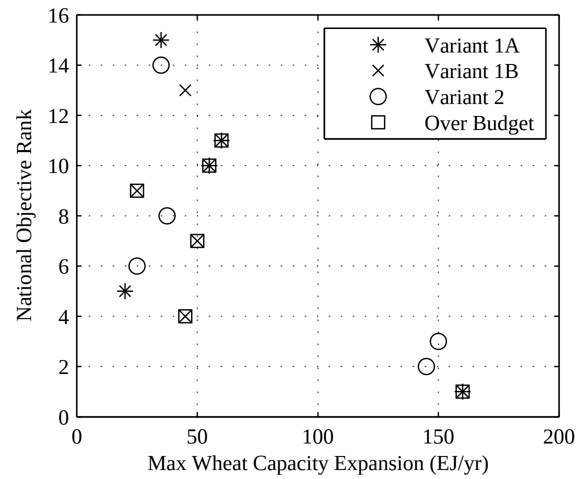

Figure 7-12: Scatter plot of SIPS-G ranked results by maximum wheat capacity expansion. All three poor outcomes demonstrate large wheat capacity expansions.

H1. Displaying an objective metric quantifying a consistent group preference leads to improved design activity outcomes.

Results do not show a statistical difference in design outcomes between Variant 1A and 1B which isolate the effects of the quantitative national objective metric, providing evidence against H1. These results are limited, however, by the capacity of participants to process and interpret task outcomes and their actual use of quantitative objective metrics. Observation and de-briefing suggest some subjects experienced data overload due to the task complexity, number of output displays, and time constraints which may limit their cognitive capacity for evaluation. Additionally, subjects may not have internalized the objective because it was stated in advance rather than an agreed upon during the session. Finally, the SIPS-G tool only provides outcomes from a single simulation execution, potentially limiting a subject's ability to evaluate a quantitative objective across rounds. Study extensions to further test H1 may consider simpler tasks or longer task durations, structured processes to enforce evaluation of objectives, visualizations comparing objectives across rounds, and group construction of a national objective during the design session.

Hypothesis H2 proposed:

H2. Integrated, distributed simulation tools enable more frequent information exchanges and lead to improved design activity outcomes.

While results do not show a statistical difference in design outcomes between tool variants, they do show a statistically-significant correlation between data exchange count and design outcomes, as well as a statistically higher data exchanges for Variant 1 compared to 2. Similar results for the effect of data exchange count on individual objective outcome are shown for the energy player as the most independent role. These results suggest a process-oriented model such as outlined by Kriz and Hense (2006) including input, process, and outcome variables. Under this approach, a revised hypothesis illustrated in Figure 7-13 suggests the tool variant providing integrated simulation allows

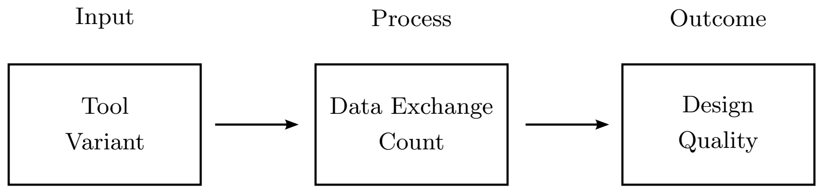

Figure 7-13: Logic model of SIPS-G input, process, and outcome variables.

higher data exchange counts which, in turn, results in higher quality of design outcomes. Future work must determine causation for observed correlations between data exchange count and outcome quality. It is certainly possible that a mediating factor such as team dynamics or capability results in higher data exchange counts and outcome quality. Approaches such as limiting the number of data exchanges may be able to isolate and control this particular process variable.

Results also show the three most poorly-performing design outcomes experienced large jumps in agricultural capacity with magnitudes about three times that of other sessions. This finding aligns with the effect of data exchange count as more frequent exchanges may limit the magnitude of changes by individual players. This also provides evidence for strategies similar to the hill-climbing behaviors discussed in Klein et al. (2003) which would create path dependencies in the design process. Although there is also evidence of concessions in the agriculture objective metric in several sessions in contrast to a pure hill-climbing strategy, there may be limits on maximum magnitude of concessions or timescales not feasible in the limited design session duration.

7.5.2 Implications for Collaborative Design

The results suggest a few important features for collaborative tools as well as useful approaches to study the design process. First, a tool providing frequent data exchanges among system models may lead to improved outcome quality. The integrated simulation tool in this study synchronizes design processes whereas the asynchronous tool may encourage local objective-optimizing behaviors. The concept of interoperable simulation gaming emphasizes enabling data exchanges across model boundaries and leverages the social interaction taking place in a game setting.

Results of this study are limited as a majority of variance between design sessions is not explained by the experimental conditions. In other words, the uncontrolled factors of group processes and information exchange outside of simulation may have a large impact on the outcomes of design sessions. This suggests that future work should either seek to study these factors in more depth or control them to a sufficient degree to further study the factors of tool design.

7.5.3 Future Work

There are a few directions for future work to extend and strengthen the findings of this study. First, no analysis has been performed of audio recordings of conversations during the design sessions. This additional mode of information exchange between participants is likely a source of unexplained variation across design sessions. Content analysis of recorded conversations may be able to identify key information or processes used to achieve improved design outcomes. Furthermore, analysis to determine centrality of conversations may be an additional direction for future work. Similar work by Broniatowski (2010) investigated transcripts of expert committees to determine information flow in social networks.

Second, several items were identified as possible extensions to improve the SIPS-G tool. First, a hybrid synchronous/asynchronous mode may combine the best features of both data exchange modes studied in this experiment. The ability to run local simulations at will while using the distributed simulation when necessary may improve outcomes by improving individual and group learning. Second, additional support should be provided for cross-round data visualization. Extensions of the tradespace plots used in this analysis would be a strong addition to the SIPS-G tool implementation. Extensions of the SIPS-G design scenario may consider alternative budget constraints, for example tied to net revenue, and objective metric formulations.

Finally, future work should determine causality between correlated variables including the frequency of data exchanges and design outcome quality. Alternative experimental designs may limit the number of allowed data exchanges or set a target objective metric without time limits. Additional process-oriented variables capturing the social layer of communication may also provide a richer model of collaborative design. For example, the logic model of Kriz and Hense (2006) includes process variables such as intensity of involvement, degree of challenge, time on-task, and intensity and quality of communication between participants.

References for Chapter 7

Broniatowski, D. A. (2010). "A Method for Analysis of Expert Committee Decision-Making Applied to FDA Medical Device Panels". PhD thesis. Cambridge, Massachusetts: Massachusetts Institute of Technology.

de Kraker, J., C. Kroeze, and P. Kirschner (2011). "Computer models as social learning tools in participatory integrated assessment". In: *International Journal of Agricultural Sustainability* 9.2, pp. 297–309. DOI: 10.1080/14735903.2011.582356.

Klein, M., P. Faratin, H. Sayama, and Y. Bar-Yam (2003). "Negotiating Complex Contracts". In: *Group Decision and Negotiation* 12.2, pp. 111–125. DOI: 10.1023/A:1023068821218.

Kolfschoten, G., S. Lukosch, and A. Mathijssen (2012). "Supporting Collaborative Design: Lessons from a case study at the ESA concurrent design facility". In: *Proceedings of the International Conference on Group Decision and Negotiation (GDN)*. Recife, Brazil.

Kriz, W. C. and J. U. Hense (2006). "Theory-oriented evaluation for the design of and research in gaming and simulation". In: *Simulation & Gaming* 37.2, pp. 268–283. DOI: 10.1177/1046878106287950.

Lu, S. C.-Y., W. Elmaraghy, G. Schuh, and R. Wilhelm (2007). "A Scientific Foundation of Collaborative Engineering". In: *CIRP Annals – Manufacturing Technology* 56.2, pp. 605–634. DOI: 10.1016/j.cirp.2007.10.010.

Ministry Commerce and Industry (2013). *Historical Overview*. URL: http://www.mci.gov.sa/en/AboutMinistry/Pages/HistoricalOverview.aspx (visited on 09/03/2013).

Ministry of Water and Electricity (2013). *A historic hint*. URL: http://www.mowe.gov.sa/English/electricitysectorhist.aspx (visited on 09/03/2013).

Rinaldi, S. M., J. P. Peerenboom, and T. K. Kelly (2001). "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies". In: *IEEE Control Systems Magazine*, pp. 11–25.

Rotmans, J. (1998). "Methods for IA: The challenges and opportunities ahead". In: *Environmental Modeling and Assessment* 3.3, pp. 155–179. DOI: 10.1023/A:1019019024003.

Chapter 8

Conclusion

> "A good organization devoted to Systems Analysis may, at any particular time, have a fairly large percent of its staff devoted to seemingly crack-pot projects. The only difficulty will be that the different individuals will put different projects into this category. Moreover, even demonstrably impractical projects can be justified because they may advance the state of the art. In any case they give people a chance to unburden themselves and to discover for themselves what is reasonable and what is not."
>
> Herman Kahn and Irwin Mann in *Ten Common Pitfalls* (1957)

This chapter outlines a summary of results to research questions, key contributions, and a critical review of interoperable simulation gaming. Finally, it concludes with topics for future work to continue developing and studying collaboration in design.

8.1 Research Summary

Chapter 2 posed three main research questions to guide the work in this dissertation. This section revisits these questions to summarize the completed research.

1. What are the relative costs of technical and social complexity in design activities with barriers to collaboration?

Design of socio-technical systems such as infrastructure involve both technical and social sources of complexity. Most existing literature only focuses on one source, for example, psychology and management on social factors and engineering on technical factors. Chapter 3 performed a controlled human subjects experiment to study multi-actor design tasks with interactions limited to verbal communication. Tasks have variable levels of technical complexity, measured in the number of variables and degree of coupling, and social complexity, measured in the number of inter-dependent designers participating in a task, and the task completion time measures design efficiency.

Results show that over the range of variables considered, social and technical contributions are independent factors. Task completion time in coupled problems increases by a geometric factor 2.9 for each additional variable which agrees with previous results. One model shows task completion time increases by a geometric factor 2.4 for each additional designer and another model shows it increases by a decreasing factor, 2.9 for the second designer and 1.4 for the third. While task completion time grows with team size, costs grow even faster as more person-hours are accrued during the design task.

2. How can interoperable simulation gaming addresses the dual challenges of integration and collaboration in infrastructure systems?

Chapter 2 introduced existing methods for systems design. Integrated modeling approaches such as that used in concurrent engineering address integration challenges in systems design but often rely on a centralized authority which may not present in infrastructure systems. Other gaming approaches such as those used for "infra-gaming" incorporate perspectives of multiple roles but rely on a development studio to develop models.

Interoperable simulation gaming draws from technology used in military wargames to provide decentralized authority over constituent system models while enabling collaborative work environments for designers. Chapters 4 and 5 outlined an approach to develop interoperable simulation models described in detail in sub-questions (a) and (b). Using these approaches, Chapter 6 developed a prototype simulation model and software implementation in the context of infrastructure in Saudi Arabia. Finally, Chapter 7 formulated a simulation game based on the prototype application to demonstrate system model integration in an interactive, collaborative design environment.

2. (a) What generalized modeling framework represents the structure and dynamics of infrastructure systems for integrated modeling activities?

Chapter 4 introduced the infrastructure system-of-systems (ISoS) modeling framework. It includes structural and behavioral templates to model generalized infrastructure systems under a common format. The structural template represents infrastructure elements as edges between nodes as units of spatial aggregation. Elements include four state properties including location, resource contents, parent element, and operational state. The behavioral template defines four resource behaviors—storing, transforming, transporting, and exchanging—and three element behaviors—storing, transforming, and transporting—as formal state changes.

The ISoS modeling framework also defines an interoperability interface to enable interactions across system model boundaries. Interoperable models must use a common set of nodes and resource types, coordinate element locations, and communicate time-stamped resource exchanging behaviors during a simulation execution.

2. (b) What simulation architecture enables collaboration with decentralized authority over component infrastructure system models?

Chapter 5 applied the IEEE 1516 High Level Architecture (HLA) to the ISoS modeling framework interoperability interface to enable decentralized authority over component models. The HLA runtime infrastructure (RTI) manages data, objects, and time advancement during a federated simulation execution and allows interoperable system models to be implemented in any language and platform supported by the RTI. Required components for the HLA application include a federation object model (FOM) to define data structures and a federation agreement to define operational procedures for initializing, advancing, and reseting a federated simulation.

3. What design-in-the-small elements of an interoperable simulation game can lead to improved design activity outcomes?

Chapter 7 uses the prototype developed in Chapter 6 to perform a series of human subjects experiments to study collaborative design in a context-rich scenario. Players act as agriculture, water, and energy ministries to develop a 30-year plan for infrastructure to meet individual and national objectives under time and budget constraints. Results indicate a positive correlation between number of data exchanges and effectiveness of national objective outcomes. The prototype variant supporting distributed simulation using the HLA experienced significantly more data exchanges compared to a prototype variant reliant on file-based data exchanges. Additionally, the worst-performing groups all had large investments in agriculture infrastructure in early rounds of the design session. These results suggest frequent exchange of information with small intermediate steps are elements which may produce improved design outcomes.

8.2 Key Contributions

This dissertation makes six key contributions:

1. *Derives scaling laws for design tasks considering technical and social sources of complexity.*

   No studies have previously quantified the combined effects of both technical and social sources of complexity in design tasks. The results in Chapter 3 establish initial scaling laws for collaborative tasks demonstrating large cost penalties for multi-actor design under barriers to collaboration. While limited in the simplicity of the surrogate task and ranges of variables considered, this contribution provides a benchmark from which to measure future extensions. In particular, the multi-actor design tool developed for the study can be adapted to new system models, user interfaces, or design team structures to investigate factors involved in collaborative design.

2. *Develops a generalizable modeling framework for infrastructure as a system-of-systems.*

   Existing infrastructure system modeling frameworks focus either on aggregated high-level dynamics or detailed low-level resource flows and do not support interoperability. The ISoS framework developed in Chapter 4 addresses these limitations for simulating infrastructure as a system-of-systems. First, it uses abstracted nodes as a unit of spatial aggregation to remove direct coupling between element models and allow multi-scale modeling. Second, its formal definition of resource and element behaviors are believed to be a complete in describing generalized infrastructure systems operations. Third, its interoperability interface defines requirements for interactions across system model boundaries in a time-managed simulation. Four descriptive application cases demonstrate the framework's applicability across multiple spatial scales and levels of operational detail.

3. *Applies the HLA standard for interoperable simulation of infrastructure system models.*

   The ISoS interoperability interface in Chapter 4 defines requirements for component simulation models but does not impose a particular implementation. Chapter 5 selects the HLA standard for federated simulation to coordinate information exchange between models. The standard does not require a particular simulation implementation, supporting models created and operated by autonomous organizations. A federation implementation adapts the general-purpose HLA to the ISoS application by defining a federation object model (FOM) and a federation agreement. The FOM defines data structures for the ISoS framework including element properties and resource exchange interactions. The federation agreement defines HLA services required to participate in an federated simulation including an iterative procedure to resolve cyclic dependencies between component models. Finally a sample Java-based federate implementation demonstrates the use of common HLA components to reduce the cost of developing federated simulation models.

4. *Demonstrates a prototype multi-sector simulation model and baseline instantiation.*

   Chapter 6 evaluates the feasibility of the ISoS modeling framework and HLA implementation in the prototype sustainable infrastructure planning simulation game (SIPS-G). Infrastructure are modeled in four sectors—agriculture, water, petroleum, and electricity—to supply demanded resources to the social system. The SIPS-G model is developed using the context of Saudi Arabia but is applicable to other applications as well. Extensive documentation describes the mathematical relationships between model parameters and outlines implementations of several modular components including resource demands, population, pricing, and infrastructure lifecycle and operations models. The prototype also establishes a baseline scenario of infrastructure instantiations describing capacity expansion and operations in Saudi Arabia with parameters based on historical data and fictionalized costs.

5. *Formulates and implements a simulation game using interoperable model components.*

   Chapter 7 uses the SIPS-G model from Chapter 6 as the foundation of an interactive simulation game based in a fictional nation similar to Saudi Arabia. Use of an interoperable and distributed model differs from existing infrastructure games which rely on customized models created by a studio or centralized developer. The SIPS-G game scenario defines objective metrics for three player roles representing agriculture, water, and energy ministries as well as national objectives. During a design session each player instantiates infrastructure elements over a planning horizon as a participatory modeling exercise.

6. *Draws insights for design-in-the-small elements in collaborative support tools.*

   The results in Chapter 7 highlight a few key insights to improve the design of collaborative support tools. First, the number of data exchanges is positively correlated with outcome quality. Furthermore, distributed simulation such as that provided by the HLA produces significantly more data exchanges compared to file-based data exchange variants. Second, large design changes in sectors with strong dependencies are negatively correlated with outcome quality. Finally, the results suggest the social layer of interaction and collaborative processes has a strong influence on outcomes and should be the focus of future study.

8.3 Critical Review of Interoperable Simulation Gaming

This dissertation develops the concept of interoperable simulation gaming for infrastructure systems design. Although probably not perceived by most as a "crack-pot" project in the words of Herman Kahn and Irwin Mann to open this chapter, the idea is quite ambitious in its goals of infusing infrastructure planning activities with collaborative modeling activities. The content of this dissertation motivates the dual challenges of integration and collaboration in infrastructure systems and demonstrates a prototype design tool for research activities. Much more work is required to move it from an exploratory research project to practice.

The main barriers to interoperable simulation gaming are the same as those in design-in-the-large: integration and collaboration among constituent designers. The findings in Chapter 3 of the high cost of collaboration apply equally to both infrastructure systems and models thereof. While the ISoS modeling framework in Chapter 4 identifies common features across a wide range of infrastructure systems, the operational details of implementing a simulation model are likely cost- and time-intensive, similar to challenges in related areas of integrated assessment (de Kraker et al. 2011). The HLA formulation in Chapter 5 provides decentralized authority over models at a high cost of complexity which has so far limited its adoption outside defense applications. Innovations in web-based applications are promising areas to lower the barrier to sharing information across infrastructure planning organizations but must be developed further to meet the needs of interoperable simulation.

As a cautionary note, there are countless examples of monstrous simulation projects failing to meet objectives due to unbounded scale and scope. This dissertation uses simulation models to represent resource flows in infrastructure—quantities which can be observed and quantified with a reasonable degree of confidence from a technical perspective. The proposed models use simple components and rely on emergence to generate complex system behavior, embracing the famous words commonly attributed to Einstein of *everything should be made as simple as possible, but not simpler.*[1] In future work there is significant risk in developing computer models of phenomena which are not as well-understood, especially those at the interface between people and society. Contentious models of socio-economic processes may be better-suited as factors to be decided upon by game players, possibly using a plurality of models when no consensus can be achieved.

Finally, this dissertation embraces the role of "engineering systems" as a multi-disciplinary field of study. Relevant literature in the preceding chapters crosses diverse domains of engineering, psychology, management, and computer sciences to synthesize new approaches. Furthermore, the research audience sits between communities of observers and practitioners. Chapters 3 and 7, for example, seek a balance between tightly-controlled experiments with strong conclusions but limited relevance and loosely-controlled experiments with stronger relevance but more reserved conclusions. Both approaches are necessary to distill and apply knowledge, and both are supported in continued study of simulation games.

8.4 Future Work

Based on the contributions and critical review of this dissertation, this section explores the following topics reserved for future work:

1. *Expand the study of scaling laws for collaborative design with technical and social complexity.*

   There are many opportunities to expand the results from the study in Chapter 3 to consider different aspects of the collaborative design problem. Future work may seek to improve the realism of the surrogate design task through expanded problem and team sizes, nonlinear or partially-coupled system models, and design evaluation based on effectiveness. Other factors may investigate the structure or nature of teams, for example physically distributed teams or partially automated design exploration. Another dimension that could be explored is the role of improved user interfaces, for example to study the impact of quantitative objectives. Another area of future work would combine the results of uncoupled and coupled problems under a single structural complexity metric.

[1]While commonly attributed to Albert Einstein, there is evidence this quote may be from an article by Roger Sessions to paraphrase a comment by Einstein (O'Toole 2011).

2. *Benchmark performance of alternative ISoS model implementations.*

   The simulation implementations in this dissertation have not been evaluated or optimized for performance. Execution time for the 60-year baseline scenario with 1-year time steps and 5 iterations varies by an order of magnitude for a standalone simulation without a user interface ($<$ 1 second), standalone with user interface updates ($\approx$ 2 seconds), and distributed simulation with user interface updates ($\approx$ 20 seconds). Performance improvements, especially optimization of user interface components, are likely to contribute large improvements in execution time. Furthermore, error analysis should consider the impact of data exchange mode (integrated simulation vs. file-based), time step size, and number of iterations on the magnitude of inconsistencies arising from cyclic model dependencies

3. *Develop and distribute an ISoS software toolkit and modeling tool.*

   The ISoS modeling framework provides a common logical template for generalized infrastructure systems. Its efficiency or effectiveness for modeling activities cannot be directly compared to other formalisms for lack of a supporting tool for building models. An ISoS software toolkit and modeling tool would allow non-programmers to build and execute models of infrastructure systems. It must be flexible enough to define custom operational behaviors for infrastructure elements as well as modular component models to add new functionality.

   An ISoS modeling tool could also serve as a centralized platform for distributed simulation. While the HLA benefits from a federated simulation architecture, the practical issues of model implementation may prevent wide-scale use. A simpler, centralized platform using a common structure may provide enough generality for wide use with fewer barriers to adoption.

4. *Explore web-based platforms for improved simulation interoperability.*

   Chapter 5 identified web-based architectures as promising approaches for interoperability with lower barriers to adoption compared to the HLA. Future work may explore the rapidly-developing capabilities used in web applications for use in infrastructure modeling and simulation. System models implemented in JavaScript may enable browser-based simulation, however future work must establish methods for communication between system models including time synchronization.

5. *Evaluate practical development of interoperable simulation models.*

   The prototype SIPS-G model and simulation game in this dissertation was developed by a single entity and does not evaluate the practical ability of multiple organizations to develop an interoperable simulation. A first step may invite a second organization to contribute an interoperable infrastructure model to evaluate the ability of the method to support joint modeling activities.

6. *Implement new component modules for the SIPS-G model.*

   The prototype SIPS-G model developed in Chapter 6 provides an initial set of component models for the design scenario which could be expanded in future work. Additions the social system model may include policy actions to control resource demands or trade tariffs, economics-based resource pricing such as supply/demand elasticity, inter-regional population movement, and demands based on socio-economic factors. Additions to all infrastructure sectors may include technological innovation effects and a capacity sizing model to generate elements of varying sizes.

Additions to the agriculture sector include additional food types (e.g. cereals, fruits, and livestock), strategic reserves, and distribution models based on distance traveled. Additions to the water sector include reservoirs and other surface sources, more accurate aquifer abstraction models, and wastewater treatment. Additions to the petroleum sector include more accurate reservoir models with both oil and natural gas and refining activities. Additions to the electricity sector include more generation types, distribution losses as a function of distance traveled, and multiple voltage levels with transformers.

7. *Improve design features of the prototype SIPS-G tool.*

   The human subjects experiment in Chapter 7 identified several potential areas of improvement for the SIPS-G tool. First, a hybrid tool capable of both synchronous simulation (using the HLA) and asynchronous using the static data from the last synchronous simulation may result in improved outcomes. This structure provides each player independent workflows for rapid iteration and individual learning while maintaining the ease of data exchange with other players for collaboration and social learning.

   Second, improved data display and visualizations across simulation executions would likely improve outcomes. In particular, the tradespaces used in Chapter 7 analysis may help understand relative values of alternative designs. This future extension aligns with existing work in multi-attribute tradespace exploration (MATE), through evaluates only a few designs rather than enumerating a large portion of the design space. Integration of automated search tools to evaluate portions of the design space may also help avoid simple hill-climbing strategies of designers.

8. *Analyze content of audio recordings as sources of unexplained variance in design experiments.*

   Factors quantified in outcome and process results from the study in Chapter 7 only explain a portion of the variance across groups. The social layer of collaboration likely offers greater explanation of differences in outcome design quality. Analysis of the content in audio recordings such as the quantity or centrality of verbal exchanges may contribute new explanations of collaborative outcomes. In particular, development of process-oriented models may be used to evaluate the ECN hypothesis.

References for Chapter 8

de Kraker, J., C. Kroeze, and P. Kirschner (2011). “Computer models as social learning tools in participatory integrated assessment”. In: *International Journal of Agricultural Sustainability* 9.2, pp. 297–309. DOI: 10.1080/14735903.2011.582356.

Kahn, H. and I. Mann (1957). *Ten Common Pitfalls.* Tech. rep. RM-1937. Santa Monica, California: The RAND Corporation.

O’Toole, G. (2011). *Everything Should Be Made as Simple as Possible, But Not Simpler.* URL: http://quoteinvestigator.com/2011/05/13/einstein-simple/ (visited on 03/30/2014).

This page is intentionally left blank.

Appendix A

Collaborative Design Experimental Data

The following tables present the raw and normalized data (using procedures 1 and 2) for the collaborative design experiments conducted in Chapter 3. Results are presented in random session order to preserve confidentiality of the subjects. Abbreviations are as follows:

i: Unit of analysis (individual, group)

j: Task type (characterized by n, N, and C/U)

k: Task replication

n: Team size

N: Problem size

C/U: Coupled/uncoupled

$C_{i,X}$: Normalization factor for unit i (and X)

Table A.1: Task ordering for sessions S-1 through S-6

Task						Order in Session					
Code Name	j	k	n	N	C/U	S-1	S-2	S-3	S-4	S-5	S-6
Breezy Rain	1	1	1	3	U	19	11	21	1	6	9
Chief Government	2	1	1	4	U	8	14	23	23	8	4
Thinkable Ink	3	1	1	6	U	9	1	9	13	7	20
Hallowed Sign	4	1	1	2	C	10	15	7	7	1	8
Husky Verse	4	2	1	2	C	7	23	19	8	15	7
Flat Sleep	5	1	1	3	C	22	19	2	16	5	18
Statuesque Name	5	2	1	3	C	15	5	5	20	4	1
Brainy Damage	6	1	1	4	C	23	24	13	15	19	12
Silky Waste	6	2	1	4	C	18	16	20	24	21	13
Alert Burst	7	1	2	3	U	4	9	10	19	9	23
Hard Development	7	2	2	3	U	5	10	14	4	14	14
Onerous Effect	7	3	2	3	U	2	2	22	3	16	5
Murky Mass	8	1	2	2	C	6	18	11	14	23	21
Unwritten Experience	8	2	2	2	C	3	13	1	6	24	16
Wistful Act	8	3	2	2	C	16	8	6	2	17	17
Absorbed Copper	9	1	2	3	C	17	21	16	11	22	22
Arrogant Flame	9	2	2	3	C	1	22	18	10	13	19
Befitting Plant	9	3	2	3	C	13	7	15	17	11	15
Better Behavior	10	1	2	4	C	20	12	8	5	3	24
Staking System	11	1	3	3	U	11	6	24	12	12	3
Towering Test	11	2	3	3	U	24	20	4	22	2	6
Wide Growth	12	1	3	6	U	14	4	17	21	20	10
Economic Motion	13	1	3	3	C	21	3	3	9	10	2
Noiseless Stone	14	1	3	4	C	12	17	12	18	18	11

Table A.2: Raw times (s) for individual design tasks

	j	4	4	5	5	6	6	1	2	3
	k	1	2	1	2	1	2	1	1	1
	n	1	1	1	1	1	1	1	1	1
	N	2	2	3	3	4	4	3	4	6
i	C/U	C	C	C	C	C	C	U	U	U
1.R		10.44	16.68	35.85	31.88	51.91	n/a	12.20	16.34	45.18
1.G		29.55	39.98	129.50	53.55	175.16	n/a	16.01	26.23	32.44
1.B		13.05	30.15	85.98	46.71	124.89	n/a	15.31	15.99	43.41
2.R		8.00	7.08	33.36	201.94	59.12	154.54	14.20	17.06	46.26
2.G		8.96	17.99	36.12	57.28	137.25	325.76	9.77	14.64	25.34
2.B		14.40	7.05	33.46	22.20	148.08	368.03	10.69	11.60	25.46
3.R		24.23	12.59	51.10	166.30	110.20	86.31	31.99	13.90	n/a
3.G		27.35	8.08	130.62	52.99	46.01	50.13	12.34	10.73	n/a
3.B		13.42	15.66	58.38	30.07	164.52	123.64	14.79	11.03	n/a
4.R		14.67	13.84	27.45	20.41	174.50	62.53	19.96	19.32	52.52
4.G		9.54	15.85	35.28	57.25	182.14	169.07	16.72	19.25	23.20
4.B		18.83	4.14	81.76	38.71	110.93	60.72	13.21	11.03	25.03
5.R		20.78	9.79	108.80	59.05	126.11	n/a	20.54	25.95	43.59
5.G		33.83	16.26	44.27	41.56	64.61	n/a	14.05	15.89	27.62
5.B		21.10	14.36	14.40	36.63	74.54	n/a	7.90	10.04	14.51
6.R		29.02	11.87	29.13	32.17	242.73	138.01	31.19	31.06	53.59
6.G		8.37	30.64	51.41	70.68	139.04	105.92	33.51	51.19	49.14
6.B		8.18	11.85	26.42	29.49	127.71	74.94	16.20	27.50	33.14

Table A.3: Raw times (s) for coupled group design tasks

	j	8	8	8	9	9	9	10	13	14
	k	1	2	3	1	2	3	1	1	1
	n	2	2	2	2	2	2	2	3	3
	N	2	2	2	3	3	3	4	3	4
i	C/U	C	C	C	C	C	C	C	C	C
1		56.28	32.92	53.88	267.99	95.03	154.42	447.32	104.80	1014.76
2		45.61	28.09	17.85	83.03	164.31	243.35	192.22	446.68	631.11
3		48.03	75.95	16.25	60.56	476.74	127.69	348.75	129.86	129.92
4		28.37	17.59	38.97	127.82	114.20	91.00	82.89	783.30	652.50
5		23.55	58.08	24.23	223.77	189.52	360.17	350.54	238.29	718.08
6		14.05	61.81	22.17	91.50	107.22	54.56	124.62	355.61	179.81

Table A.4: Raw times (s) for uncoupled group design tasks

	j	7	7	7	11	11	12
	k	1	2	3	1	2	1
	n	2	2	2	3	3	3
	N	3	3	3	3	3	6
i	C/U	U	U	U	U	U	U
1		54.02	32.51	61.17	24.47	43.89	85.92
2		41.07	29.91	44.38	42.78	60.33	144.63
3		30.91	36.70	32.07	29.40	65.27	113.92
4		23.65	33.60	46.91	32.75	28.01	44.75
5		50.89	20.69	23.47	35.51	79.86	83.59
6		28.05	34.83	36.05	89.44	33.31	126.14

Table A.5: Normalized times (s) for individual tasks using procedure 1

		j	4	4	5	5	6	6	1	2	3
		k	1	2	1	2	1	2	1	1	1
		n	1	1	1	1	1	1	1	1	1
		N	2	2	3	3	4	4	3	4	6
i	C_i	C/U	C	C	C	C	C	C	U	U	U
1.R	1.38		14.42	23.03	49.50	44.02	71.68	n/a	16.85	22.56	62.39
1.G	0.61		17.91	24.23	78.47	32.45	106.14	n/a	9.70	15.89	19.66
1.B	0.81		10.58	24.45	69.71	37.87	101.26	n/a	12.41	12.96	35.20
2.R	0.79		6.35	5.62	26.48	160.32	46.94	122.69	11.27	13.54	36.73
2.G	0.68		6.08	12.22	24.53	38.90	93.21	221.22	6.63	9.94	17.21
2.B	0.67		9.66	4.73	22.44	14.89	99.33	246.86	7.17	7.78	17.08
3.R	0.80		19.36	10.06	40.83	132.88	88.05	68.96	25.56	11.11	n/a
3.G	1.17		32.08	9.48	153.23	62.16	53.97	58.81	14.48	12.59	n/a
3.B	0.92		12.34	14.40	53.68	27.65	151.29	113.70	13.60	10.14	n/a
4.R	1.06		15.57	14.69	29.13	21.66	185.16	66.35	21.18	20.50	55.73
4.G	0.81		7.76	12.90	28.71	46.59	148.23	137.59	13.61	15.67	18.88
4.B	1.18		22.22	4.89	96.48	45.68	130.90	71.65	15.59	13.02	29.54
5.R	0.73		15.26	7.19	79.89	43.36	92.60	n/a	15.08	19.05	32.01
5.G	1.18		39.91	19.18	52.22	49.02	76.21	n/a	16.57	18.74	32.58
5.B	1.57		33.20	22.60	22.66	57.64	117.29	n/a	12.43	15.80	22.83
6.R	0.72		20.84	8.52	20.92	23.10	174.29	99.10	22.40	22.30	38.48
6.G	0.80		6.67	24.40	40.94	56.29	110.72	84.35	26.69	40.76	39.13
6.B	1.21		9.89	14.33	31.96	35.67	154.48	90.65	19.60	33.27	40.09

Table A.6: Normalized times (s) for coupled group design tasks using procedure 1

		j	8	8	8	9	9	9	10	13	14
		k	1	2	3	1	2	3	1	1	1
		n	2	2	2	2	2	2	2	3	3
		N	2	2	2	3	3	3	4	3	4
i	C_i	C/U	C	C	C	C	C	C	C	C	C
1	0.78		43.73	25.58	41.86	208.22	73.84	119.98	347.56	81.43	788.46
2	0.89		40.46	24.92	15.84	73.66	145.76	215.88	170.52	396.26	559.87
3	1.14		54.82	86.68	18.55	69.12	544.09	145.73	398.02	148.21	148.27
4	0.92		25.98	16.11	35.68	117.04	104.57	83.33	75.90	717.24	597.47
5	0.79		18.66	46.02	19.20	177.31	150.17	285.39	277.76	188.82	568.99
6	1.45		20.32	89.37	32.06	132.30	155.04	78.89	180.19	514.20	260.00

Table A.7: Normalized times (s) for uncoupled group design tasks using procedure 1

		j	7	7	7	11	11	12
		k	1	2	3	1	2	1
		n	2	2	2	3	3	3
		N	3	3	3	3	3	6
i	C_i	C/U	U	U	U	U	U	U
1	0.78		41.97	25.26	47.53	19.01	34.10	66.76
2	0.89		36.43	26.53	39.37	37.95	53.52	128.31
3	1.14		35.28	41.88	36.60	33.55	74.49	130.01
4	0.92		21.66	30.77	42.95	29.99	25.65	40.98
5	0.79		40.32	16.39	18.60	28.14	63.28	66.24
6	1.45		40.56	50.36	52.13	129.33	48.16	182.39

Table A.8: Normalized times (s) for individual tasks using procedure 2

			j	4	4	5	5	6	6	1	2	3
			k	1	2	1	2	1	2	1	1	1
			n	1	1	1	1	1	1	1	1	1
			N	2	2	3	3	4	4	3	4	6
i	$C_{i,C}$	$C_{i,U}$	C/U	C	C	C	C	C	C	U	U	U
1.R	1.64	0.87		17.08	27.29	58.65	52.15	84.92	n/a	10.65	14.26	39.44
1.G	0.56	0.86		16.59	22.44	72.69	30.06	98.32	n/a	13.80	22.60	27.95
1.B	0.80	0.86		10.42	24.07	68.63	37.29	99.69	n/a	13.19	13.77	37.39
2.R	0.79	0.83		6.30	5.58	26.28	159.10	46.58	121.75	11.79	14.16	38.40
2.G	0.63	1.29		5.62	11.27	22.64	35.90	86.01	204.15	12.64	18.94	32.78
2.B	0.62	1.35		8.87	4.34	20.62	13.68	91.26	226.81	14.41	15.63	34.31
3.R	0.81	0.68		19.65	10.21	41.45	134.89	89.38	70.01	21.76	9.45	n/a
3.G	1.16	1.35		31.72	9.37	151.51	61.47	53.37	58.15	16.70	14.52	n/a
3.B	0.90	1.21		12.09	14.11	52.61	27.10	148.26	111.42	17.88	13.33	n/a
4.R	1.17	0.70		17.11	16.14	32.02	23.81	203.56	72.94	13.99	13.54	36.82
4.G	0.78	1.09		7.43	12.35	27.49	44.61	141.94	131.76	18.19	20.94	25.23
4.B	1.16	1.31		21.85	4.80	94.86	44.91	128.71	70.45	17.25	14.41	32.69
5.R	0.74	0.71		15.37	7.24	80.49	43.69	93.30	n/a	14.67	18.54	31.14
5.G	1.20	1.12		40.50	19.47	53.00	49.76	77.36	n/a	15.71	17.77	30.88
5.B	1.49	1.98		31.46	21.41	21.47	54.61	111.14	n/a	15.67	19.91	28.78
6.R	0.76	0.56		21.97	8.99	22.05	24.35	183.75	104.48	17.33	17.26	29.77
6.G	0.90	0.48		7.54	27.59	46.29	63.64	125.18	95.36	16.11	24.61	23.63
6.B	1.31	0.84		10.73	15.55	34.67	38.70	167.59	98.34	13.57	23.03	27.76

Table A.9: Normalized times (s) for coupled group design tasks using procedure 2

		j	8	8	8	9	9	9	10	13	14
		k	1	2	3	1	2	3	1	1	1
		n	2	2	2	2	2	2	2	3	3
		N	2	2	2	3	3	3	4	3	4
i	$C_{i,C}$	C/U	C	C	C	C	C	C	C	C	C
1	0.76		42.56	24.90	40.75	202.66	71.87	116.78	338.28	79.25	767.40
2	0.91		41.48	25.55	16.23	75.51	149.42	221.30	174.81	406.21	573.93
3	1.19		57.23	90.49	19.36	72.16	568.02	152.14	415.52	154.72	154.80
4	0.87		24.68	15.30	33.90	111.17	99.33	79.15	72.10	681.29	567.53
5	0.77		18.14	44.75	18.67	172.41	146.02	277.50	270.08	183.60	553.26
6	1.67		23.40	102.95	36.92	152.40	178.58	90.87	207.56	592.28	299.48

Table A.10: Normalized times (s) for uncoupled group design tasks using procedure 2

		j	7	7	7	11	11	12
		k	1	2	3	1	2	1
		n	2	2	2	3	3	3
		N	3	3	3	3	3	6
i	$C_{i,U}$	C/U	U	U	U	U	U	U
1	0.93		50.24	30.24	56.89	22.76	40.82	79.91
2	0.77		31.77	23.14	34.33	33.09	46.66	111.87
3	0.91		28.16	33.44	29.22	26.79	59.47	103.79
4	1.34		31.68	45.01	62.84	43.87	37.52	59.94
5	0.96		48.61	19.76	22.42	33.92	76.29	79.85
6	0.81		22.65	28.12	29.11	72.22	26.90	101.85

This page is intentionally left blank.

Appendix B

SIPS-G Software Implementation

This appendix discusses the software implementation of the SIPS-G model formulated in Chapter 6. The federation implementation is based on the generic ISoS federation described in Chapter 5 with a few simplifying assumptions. The federates are implemented in Java using a common SIPS-G library and include a graphical user interface (GUI) for player interaction with the simulation model. Section B.1 discusses the federation implementation including object models and agreements. Section B.2 discusses the federate implementation including each of the infrastructure system models. Finally, Section B.3 discusses the GUI implementation.

B.1 Federation Implementation

The SIPS-G federation implementation is based on the generic ISoS federation discussed in Chapter 5 using the attribute option for resource exchanging behaviors, text-based node representation, and omitted state and contents attributes. It implements several simplifying assumptions specific to the SIPS-G framing.

First, nodal infrastructure systems serve as the only federation elements. As there is only one of each system at each node region, the node-based attribute can be used in lieu of a location (node pair). Second, rather than composing all resource exchanges in a single attribute, the SIPS-G federation uses separate attributes for each resource type. For example, the electricity system has separate attributes for water and oil input which would be components of the `ExchangeOutputs` attribute[1] in the more general ISoS federation. Furthermore, there is a single source of each resource type (e.g. food from the agriculture system) such that there is no need for directed resource flows specifying resources and the source or target element. Third, as each infrastructure system has a mechanism for unbounded resource production (i.e. private production or import), there cannot be a shortfall in resource supply and the equivalent of a `ExchangeInputs` attribute is included only for completeness. Finally, the SIPS-G federation includes resource unit prices as infrastructure system attributes which can be interpreted in the ISoS framework as information exchanging behaviors.

The SIPS-G FOM in Table B.1 defines object classes and attributes using standard HLA data types. Each quantitative data item is also assigned units of measurement: food energy in gigajoules

[1]To clarify potentially confusing notation, outputs from the exchanging process (`ExchangeOutputs`) are inputs to an infrastructure system (R_{input}). Similarly, inputs to the exchanging process (`ExchangeInputs`) are outputs from an infrastructure system (R_{output}).

(GJ), water in cubic meters (m^3), oil in tonnes of oil equivalent (toe), and electricity in megawatt hours (MWh). The `InfrastructureSystem` object class inherits from the `HLAobjectRoot` base class and defines the common attributes across all the systems including a name, region, and cash flow attributes. The `AgricultureSystem`, `WaterSystem`, `PetroleumSystem`, `ElectricalSystem`, and `SocialSystem` object classes include attributes for each associated resource input and unit price. Finally, `SocialSystem` includes population and net revenue attributes to match the logical model variables.

The SIPS-G federation agreement extends the generic ISoS federation agreement. It also specifies the roles and responsibilities of each federate with object instances assigned in Table B.2 following the player roles. All federates must adhere to the start-up, advance, reset, and shut-down processes and establish common operational timing parameters using the `HLAinteger64Time` data type including

1. $t_0 = 1950000$ the initial time in 1/1000 years,
2. $\xi = 200$, the recommended pseudo-step size in 1/1000 years,
3. $\Delta t = 1000$, the step size in 1/1000 years, and
4. $t_f = 2010000$ the final time in 1/1000 years.

B.2 Federate Implementation

A single SIPS-G federate is instantiated with different local model components for each player role. The federate is implemented as a Java application using a portion of the common ISoS library discussed in Chapter 5. Figure B-1 illustrates a simplified object class diagram emphasizing the common interfaces for the SIPS-G application (blue boxes) extending some of the standard ISoS classes (gray boxes).

The SIPS-G federate implementation `FederateSIPSG` composes a `Country` object which, in turn, composes a list of `Region` objects as nodes. Both `Country` and `Region` classes implement a common `Society` interface which includes methods to access member infrastructure systems and aggregated resource inputs, net and cumulative revenues for the composed agriculture, water, petroleum, electricity, and social systems.

Given the simplifying assumptions in SIPS-G, the main interface `InfrastructureSystem` extends `SimObject` (rather than `Element`) with methods to access the associated name, net revenue, and society. The associated HLA model implementations (orange boxes) implement the core SIPS-G interfaces using standard HLA data types. The local model implementation also includes infrastructure "super system" (`*SSystem`) to aggregate behaviors of all component systems. For example, the `InfrastructureSSystem` aggregates the total net revenue of all nested `InfrastructureSystem` objects. These objects can be considered convenience classes; however the sector-specific classes also manage flow optimization across component systems.

B.2.1 Social System Model Implementations

Figure B-2 presents a class diagram of the social system model implementations. The `SocialSystem` interface specifies methods to access population, revenue, and food, water, oil, and electricity

Table B.1: SIPS-G federation object model definition

Object Class	Attribute	Data	Updating	Model Property	Units
InfrastructureSystem	Name	[1]	Static	–	–
	Region	[1]	Static	–	–
	NetRevenue	[2]	Periodic	$Q_{netRevenue}(\mathbf{E}_i^n)$	§/yr
AgricultureSystem	LocalFoodPrice	[2]	Static	π_{local}^{food}	§/GJ
	FoodOutput	[2]	Periodic	$Q_{output}^{food}(\mathbf{E}_{agricul.}^n)$	GJ/yr
	WaterInput	[2]	Periodic	$Q_{input}^{water}(\mathbf{E}_{agricul.}^n)$	m^3/yr
WaterSystem	LocalWaterPrice	[2]	Static	π_{local}^{water}	§/m^3
	WaterOutput	[2]	Periodic	$Q_{output}^{water}(\mathbf{E}_{water}^n)$	m^3/yr
	ElectricityInput	[2]	Periodic	$Q_{input}^{elect}(\mathbf{E}_{water}^n)$	MWh/yr
	OilInput	[2]	Periodic	$Q_{input}^{oil}(\mathbf{E}_{water}^n)$	toe/yr
PetroleumSystem	LocalOilPrice	[2]	Static	π_{local}^{oil}	§/toe
	OilOutput	[2]	Periodic	$Q_{output}^{oil}(\mathbf{E}_{petrol.}^n)$	toe/yr
	ElectricityInput	[2]	Periodic	$Q_{input}^{water}(\mathbf{E}_{petrol.}^n)$	MWh/yr
ElectricalSystem	LocalElectricityPrice	[2]	Static	π_{local}^{elect}	§/MWh
	ElectricityOutput	[2]	Periodic	$Q_{output}^{elect}(\mathbf{E}_{elect.}^n)$	MWh/yr
	WaterInput	[2]	Periodic	$Q_{input}^{water}(\mathbf{E}_{elect.}^n)$	m^3/yr
	OilInput	[2]	Periodic	$Q_{input}^{oil}(\mathbf{E}_{elect.}^n)$	toe/yr
SocialSystem	Population	[3]	Periodic	$Q_{stock}^{population}(\mathbf{E}_{social}^n)$	people
	CumulativeNetRevenue	[2]	Periodic	$Q_{stock}^{cash}(\mathbf{E}_{social}^n)$	§
	FoodInput	[2]	Periodic	$Q_{input}^{food}(\mathbf{E}_{social}^n)$	GJ/yr
	OilInput	[2]	Periodic	$Q_{input}^{oil}(\mathbf{E}_{social}^n)$	toe/yr
	WaterInput	[2]	Periodic	$Q_{input}^{water}(\mathbf{E}_{social}^n)$	m^3/yr
	ElectricityInput	[2]	Periodic	$Q_{input}^{elect}(\mathbf{E}_{social}^n)$	MWh/yr

[1] HLAunicodeString encoding
[2] HLAfloat64BE encoding
[3] HLAinteger64BE encoding

Table B.2: SIPS-G federate object instance assignments

Object Classes	Water Federate	Energy Federate	Agriculture Federate
Social System	1 (Urban node)	1 (Industrial node)	1 (Rural node)
Water System	3 (Each node)	0	0
Electricity System	0	3 (Each node)	0
Petroleum System	0	3 (Each node)	0
Agriculture System	0	0	3 (Each node)

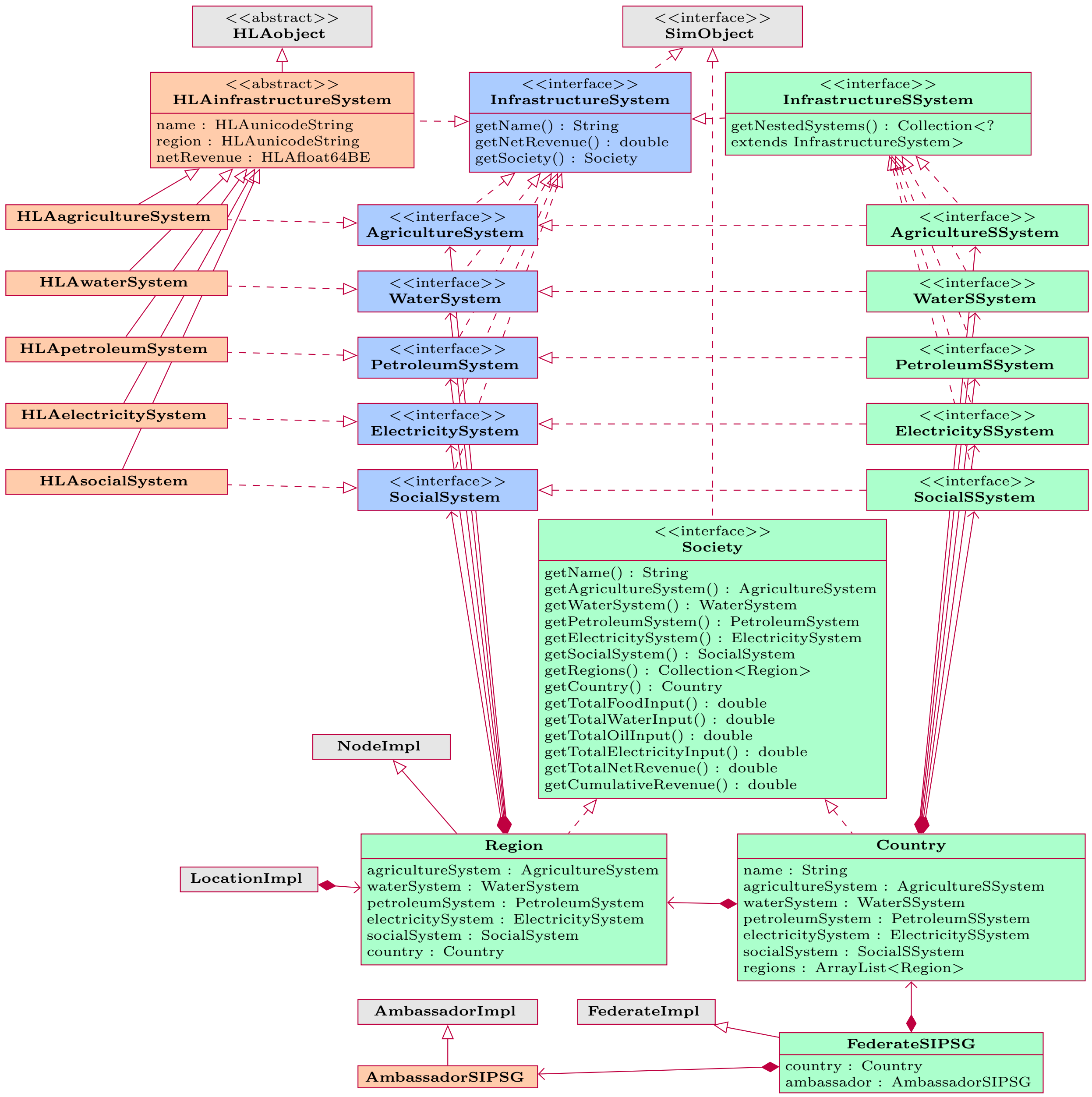

Figure B-1: Class diagram of SIPS-G objects. Gray boxes are common ISSystem object classes, blue boxes are SIPS-G model interfaces, orange boxes are HLA model implementations, and green boxes are local model implementations.

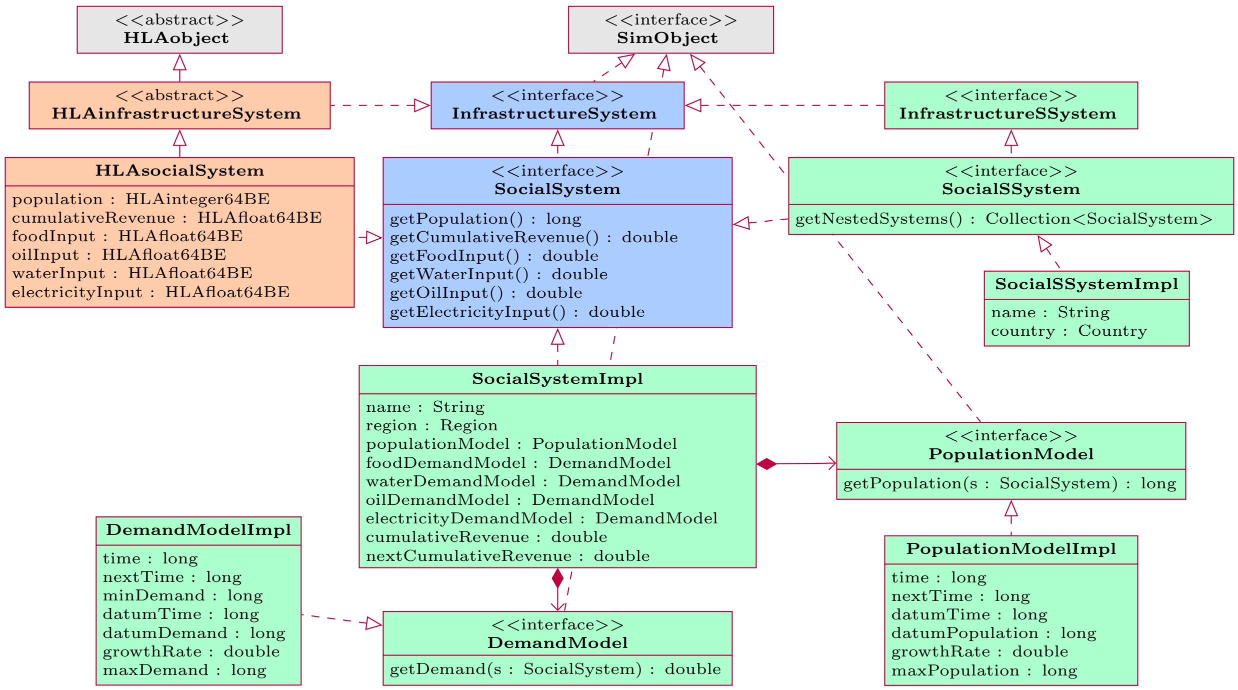

Figure B-2: Class diagram of SIPS-G social system implementation. Gray boxes are common ISSystem object classes, blue boxes are SIPS-G interfaces, orange boxes are HLA social system model implementations, and green boxes are local social system model implementations.

resource demands. The `HLAsocialSystem` model implementation uses standard HLA data types for each attribute. The `SocialSystemImpl` local model implementation stores the cumulative revenue stock as an attribute and composes a `PopulationModel` object to determine population and several `DemandModel` objects to determine inputs for each resource type. The population and demand models `PopulationModelImpl` and `DemandModelImpl` implement the logistic growth population function in Eq. 6.13 and logistic per-capita demand function in Eq. 6.14.

B.2.2 Generic Infrastructure System Model Implementations

Figure B-3 presents an object class diagram of the generic infrastructure system model implementations. In addition to the methods implemented from the `InfrastructureSystem` interface in Fig. B-1, the `InfrastructureSystemImpl` local model defines methods for common SIPS-G model properties including capital, operations, decommission, input, and distribution expense and output, distribution, and export revenue. Three additional methods access collections of elements: internal elements having an origin within the system ($\mathbf{E}_i^n$ in Eq. 6.5), external elements having a destination within the system but an origin elsewhere ($\mathbf{E}_i^{n\star}$ in Eq. 6.6), and the union of both sets.

The `InfrastructureElement` interface defines methods for SIPS-G element properties including the name, location, and capital, operations, and decommission expense. It also includes a method to check if it is operational. The abstract implementation `InfrastructureElementImpl` includes data members for the name and location and composes a lifecycle model defined by the interface `LifecycleModel` for the remaining properties. The implemented lifecycle model `LifecycleModelImpl` follows the distinct states described in Table 6.3.

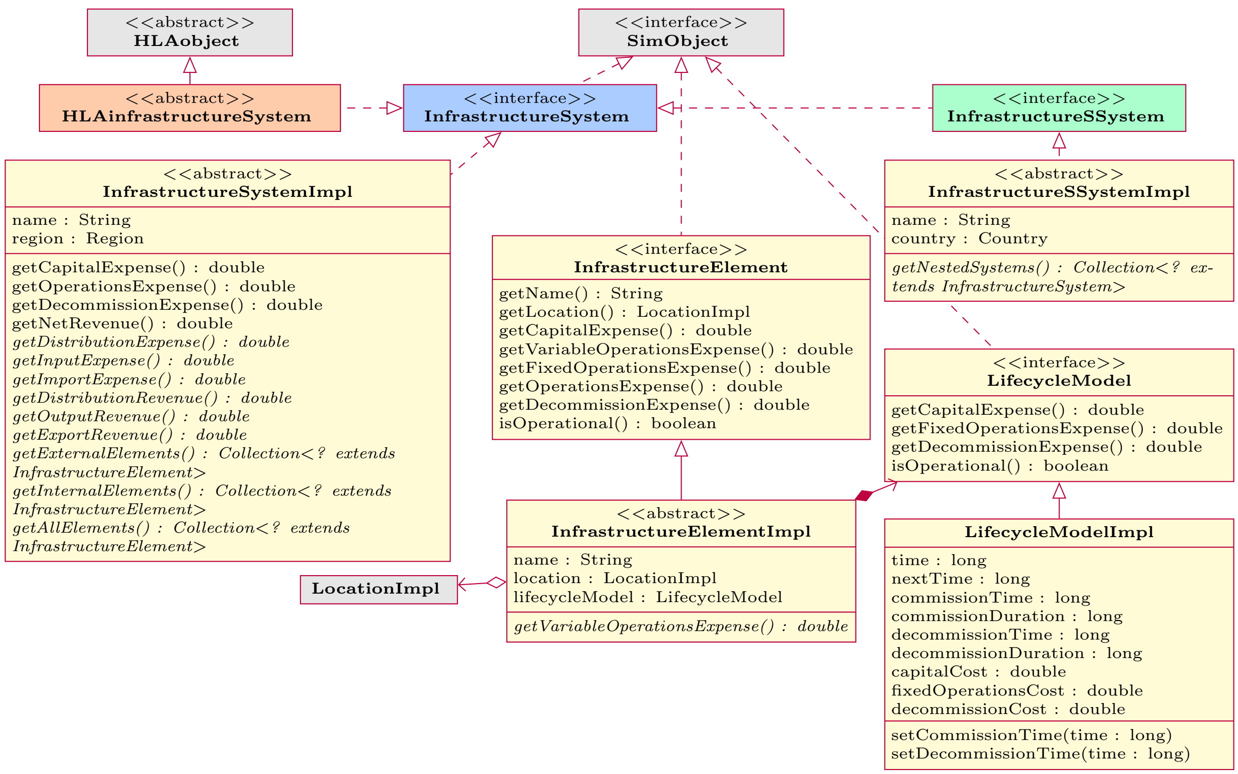

Figure B-3: Class diagram of SIPS-G generic infrastructure system implementation. Gray boxes are common ISSystem object classes, blue boxes are SIPS-G interfaces, orange boxes are HLA model implementations and yellow boxes are generic infrastructure model implementations.

B.2.3 Agriculture System Model Implementations

Figure B-4 presents an object class diagram of the agriculture system model implementations. The `AgricultureSystem` interface includes methods to access the local food price, food output, and water input. The `HLAagricultureSystem` model implementation uses standard HLA data types for each attribute. The `AgricultureSystemImpl` implementation extends `InfrastructureSystemImpl` to add data members for local, import, and export pricing models extending the `PricingModel` interface, arable land area, labor force and participation rate (with updates per Eq. 6.77), and a list of internal agricultural elements. It also defines methods for SIPS-G model properties including arable land area (available and used), labor force (available and used), import and export prices, food resource flows (exported, imported, received, sent, produced, supplied, and wasted), expenses (distribution, input, and import), and revenues (distribution, output, and export).

The `AgricultureSSystem` interface defines methods to optimize food production and distribution. The `AgricultureSSystemImpl` class implements these methods using Eq. 6.84–6.90 and the `SimplexSolver` LP solver from the Apache Commons Math 3.2 library (Apache 2013).

The `AgricultureElement` interface defines common methods for all agriculture elements including accessing food sent, received, and produced, input water resources, land area and labor used, and the variable operations expense. The `AgricultureProductionElement` implementation includes data members for current and maximum land area used and for properties in Eq. 6.78–6.81. The `AgricultureDistributionElement` implementation includes data members for current and maximum food sent and for properties in Eq. 6.82–6.83

B.2.4 Water System Model Implementations

Figure B-5 presents an object class diagram of the water system model implementations. The `WaterSystem` interface includes methods to access the local water price, water output, and electricity input. The `HLAwaterSystem` model implementation uses standard HLA data types for each attribute. The `WaterSystemImpl` implementation extends `InfrastructureSystemImpl` to add data members for local and import pricing models extending the `PricingModel` interface, initial and current aquifer volume and recharge rate (with updates per Eq. 6.112), a private production element with properties per Eq. 6.113–6.114, and a list of internal water elements. It also defines methods for SIPS-G model properties including aquifer volume, water resource flows(imported, received, sent, produced, supplied, and wasted), expenses (distribution, input, and import), and revenues (distribution and output).

The `WaterSSystem` interface defines methods to optimize water production and distribution. The `WaterSSystemImpl` class implements these methods using Eq. 6.121–6.127 and the LP solver `SimplexSolver` from the Apache Commons Math 3.2 library (Apache 2013).

The `WaterElement` interface defines common methods for all water elements including accessing water sent, received, and produced, input electricity and aquifer resources, and variable operations expense. The `WaterProductionElement` implementation includes data members for current and maximum water produced and for properties in Eq. 6.115–6.117. The `WaterDistributionElement` implementation includes data members for current and maximum water sent and for properties in Eq. 6.118–6.120.

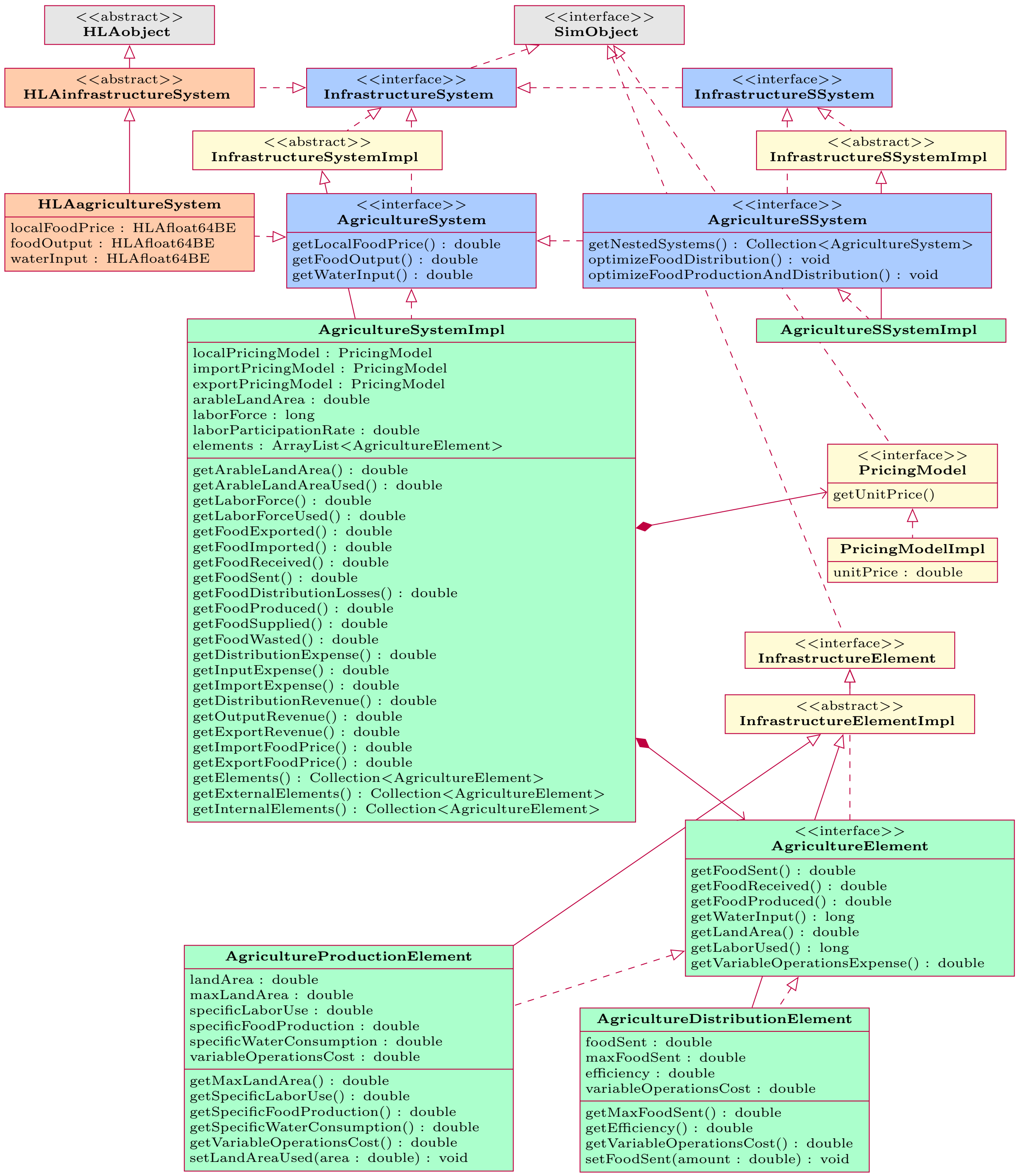

Figure B-4: Class diagram of SIPS-G agriculture system implementation. Gray boxes are common ISSystem object classes, blue boxes are SIPS-G interfaces, orange boxes are HLA model implementations, yellow boxes are generic model implementations, and green boxes are local agriculture model implementations.

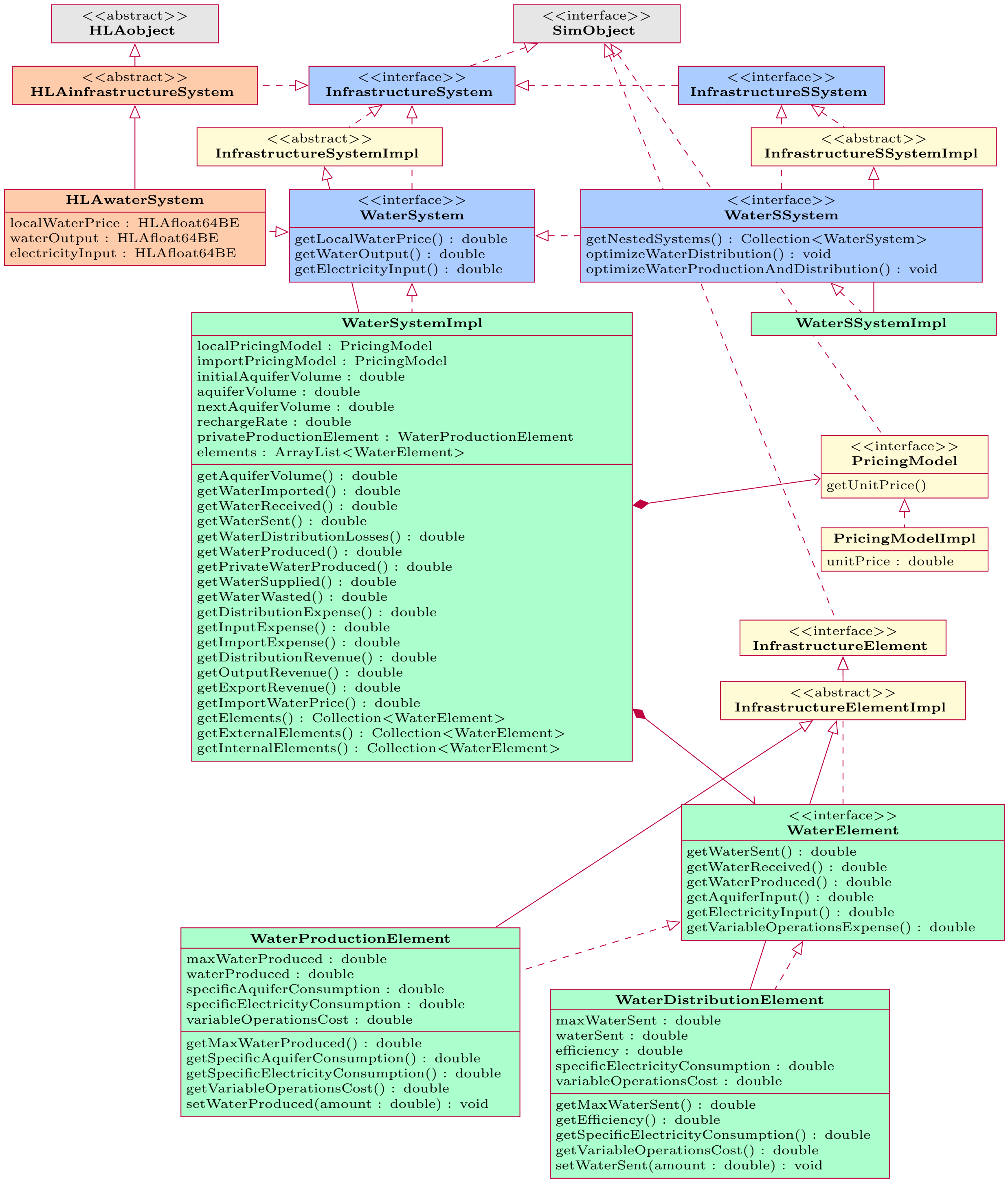

Figure B-5: Class diagram of SIPS-G water system implementation. Gray boxes are common ISSystem object classes, blue boxes are SIPS-G interfaces, orange boxes are HLA water model implementations, yellow boxes are generic infrastructure model implementations, and green boxes are local water model implementations.

B.2.5 Petroleum System Model Implementations

Figure B-6 presents an object class diagram of the petroleum system model implementations. The `PetroleumSystem` interface includes methods to access local oil prices, oil output, and electricity input. The `HLApetroleumSystem` model implementation uses standard HLA data types for each attribute. The `PetroleumSystemImpl` implementation extends `InfrastructureSystemImpl` to add data members for local, import, and export pricing models extending the `PricingModel` interface, initial and current reservoir volume, and a list of internal petroleum elements. It also defines methods for SIPS-G model properties including reservoir volume, oil resource flows (imported, exported, received, sent, produced, supplied, and wasted), input electricity resources, expenses (distribution, input, and import) and revenues (distribution, output, and export).

The `PetroleumSSystem` interface defines additional methods to optimize oil production and distribution. The `PetroleumSSystemImpl` class implements these methods using Eq. 6.152–6.157 and the `SimplexSolver` LP solver from the Apache Commons Math 3.2 library (Apache 2013).

The `PetroleumElement` interface defines common methods for all petroleum elements including accessing oil sent, received, and produced, input electricity and reservoir resources, and variable operations expense. The `PetroleumProductionElement` implementation includes data members for current and maximum oil produced and for other properties in Eq. 6.147–6.148. The `PetroleumDistributionElement` implementation includes data members for current and maximum oil sent and for properties in Eq. 6.149–6.151.

B.2.6 Electricity System Model Implementations

Figure B-7 presents an object class diagram of the electricity system model implementations. The `ElectricitySystem` interface includes methods to access local electricity prices, output electricity, and input water and oil. The `HLAelectricitySystem` model implementation uses standard HLA data types for each attribute. The `ElectricitySystemImpl` implementation extends `InfrastructureSystemImpl` to add data members for a local pricing model using the `PricingModel` interface, a private production element with properties per Eq. 6.175, and a list of internal electricity elements. It also defines methods for SIPS-G model properties including electricity resource flows (received, sent, produced, supplied, and wasted), expenses (distribution and input) and revenues (distribution and output).

The `ElectricitySSystem` interface defines additional methods to optimize electricity production and distribution. The `ElectricitySSystemImpl` class implements these methods using Eq. 6.181–6.186 and the `SimplexSolver` LP solver from the Apache Commons Math 3.2 library (Apache 2013).

The `ElectricityElement` interface defines common methods for all electricity elements including accessing electricity sent, received, and produced, input water and oil resources, and variable operations expense. The `ElectricityProductionElement` implementation includes data members for current and maximum electricity produced and for properties in Eq. 6.176–6.178. The `ElectricityDistributionElement` implementation includes data members for current and maximum electricity sent and for properties in Eq. 6.179–6.180.

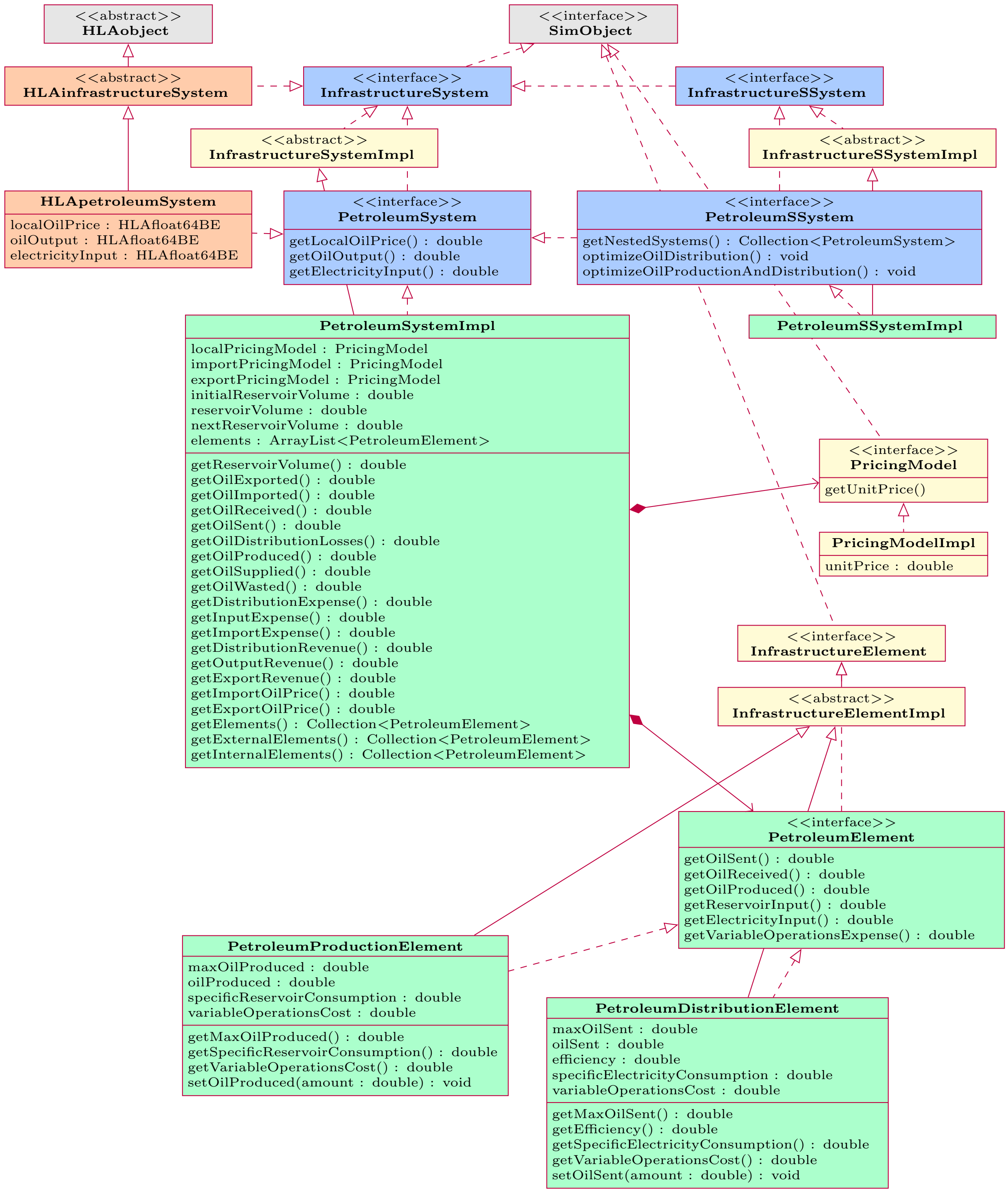

Figure B-6: Class diagram of SIPS-G petroleum system implementation. Gray boxes are common ISSystem object classes, blue boxes are SIPS-G interfaces, orange boxes are HLA petroleum model implementations, yellow boxes are generic infrastructure model implementations, and green boxes are local petroleum model implementations.

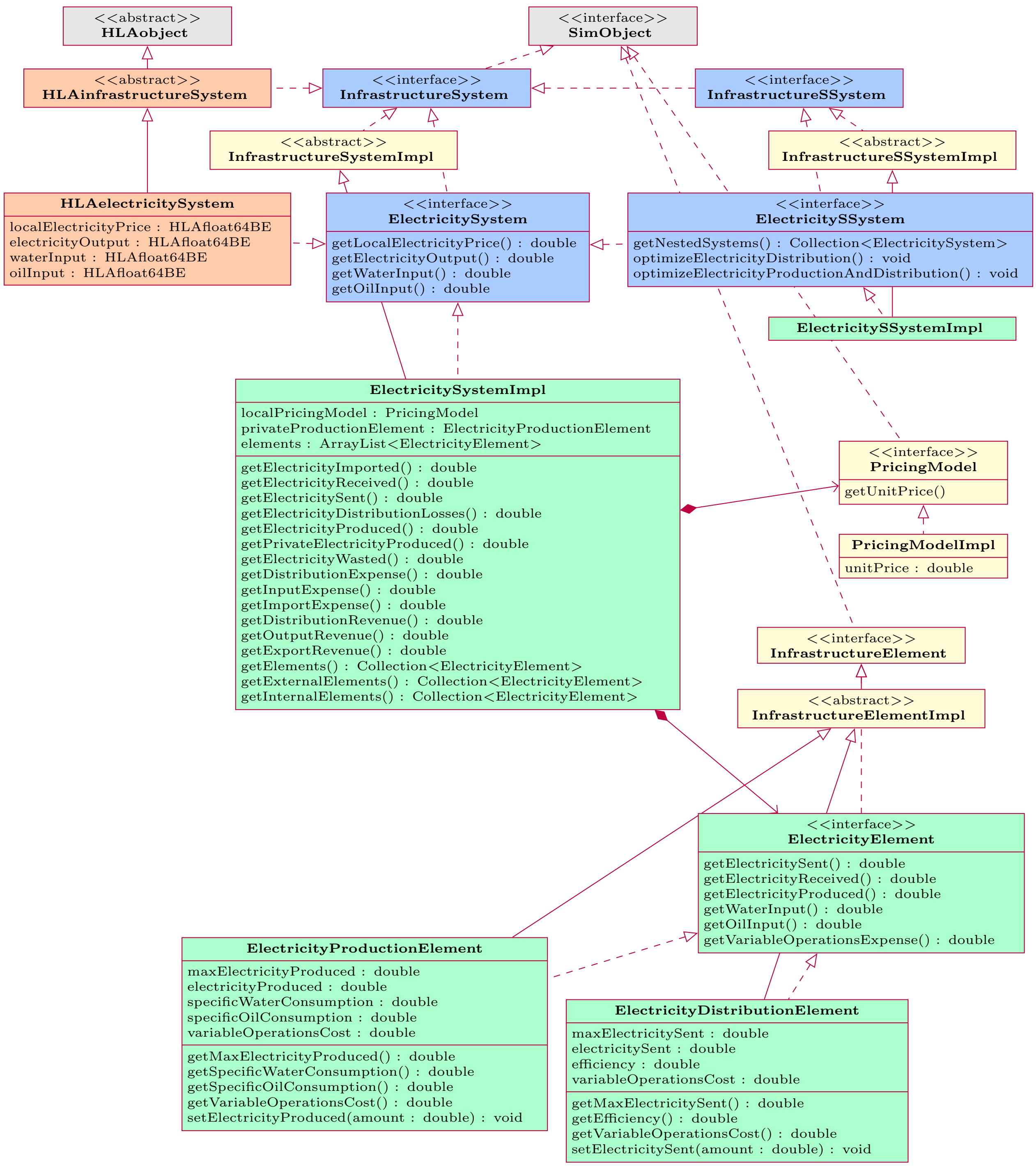

Figure B-7: Class diagram of SIPS-G electricity system implementation. Gray boxes are common ISSystem object classes, blue boxes are SIPS-G interfaces, orange boxes are HLA electricity model implementations, yellow boxes are generic infrastructure model implementations, and green boxes are local electricity model implementations.

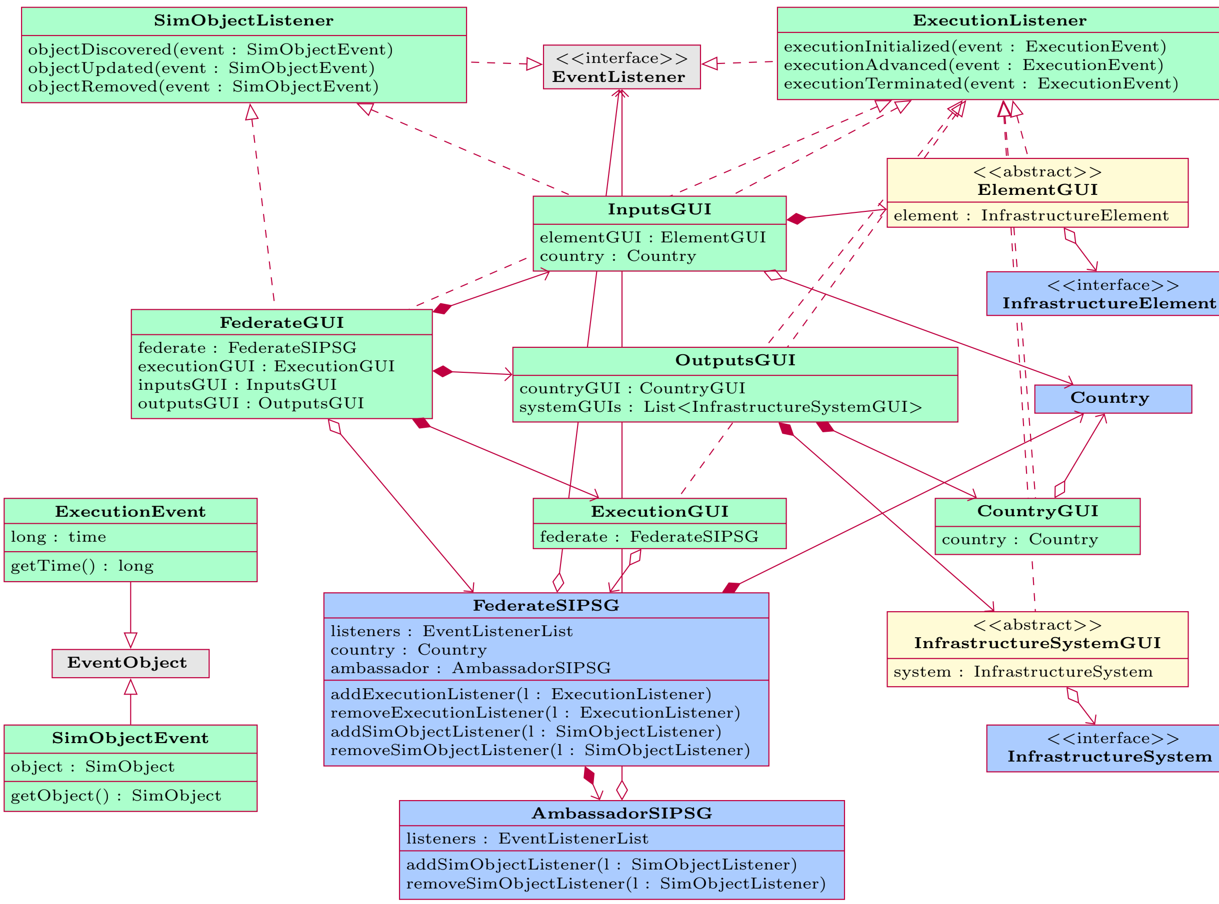

Figure B-8: Class diagram of SIPS-G graphical user interface implementation. Gray boxes are standard Java classes, blue boxes are SIPS-G model components, green boxes are SIPS-G GUI components, and yellow boxes are generic SIPS-G GUI components. Specific infrastructure system GUI components are omitted.

B.3 Graphical User Interface

A graphical user interface (GUI) provides a player access to the infrastructure system models during a game session, both to specify simulation inputs (where and when to build which infrastructure elements) and to visualize outputs including resource stocks and flows computed in the simulation results.

Figure B-8 illustrates a simplified object class diagram of the GUI implementation. Many components rely on an observer pattern to notify and execute simulation object events. The classes `ExecutionEvent` and `SimObjectEvent` extend the Java `EventObject` class for use as method arguments in the `ExecutionListener` and `SimObjectListener` listener classes. Each GUI component implements the required `*Listener` interface and is registered as a listener with both the federate (for local events) and the ambassador (for remote events). For example, when the ambassador receives an attribute update service call for a remote infrastructure system, it would call the `objectUpdated` method for all associated listeners.

Figure B-9 shows a screen capture of the SIPS-G GUI with several components highlighted. The

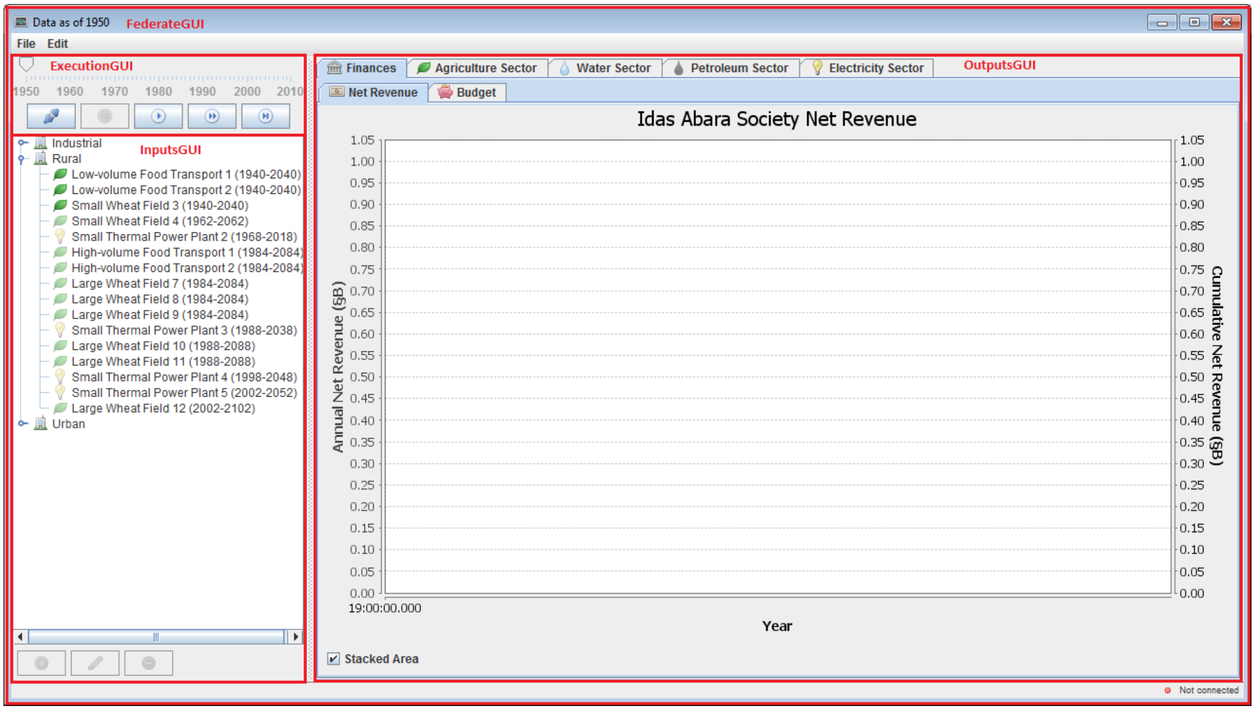

Figure B-9: Screen capture of the main SIPS-G graphical user interface includes the execution component (upper left), inputs component (left), and outputs component (right) with nested country and infrastructure system components as tabs.

class `FederateGUI` acts as the top-level GUI component and contains three main sub-components. The `ExecutionGUI` component at the top left provides user controls for the simulation execution. A slider bounds the initial and final simulation times. Buttons below the slider allow for user commands to connect to a federation, initialize or reset to initial conditions, and advance by one year, five years, or to the end of the simulation.

The `InputsGUI` component receives user input to define an infrastructure plan. A tree list component organizes infrastructure elements by region. In the example in Fig. B-9, the rural region node is expanded to show various infrastructure elements. Full-color icons illustrate operational elements while faded icons illustrate elements in an empty (pre-commission) or null (post-decommission) state. Buttons along the bottom allow for user commands to add new elements, edit existing elements, or delete existing elements. Individual elements can be edited in `ElementGUI` components. Figure B-10 shows a screen capture for agriculture production and distribution elements.

The `OutputsGUI` component displays outputs from the simulation execution. The `CountryGUI` component provides national financial data including net revenue and capital expenses by infrastructure sector and total cumulative revenue. Each sector-specific `InfrastructureSystemGUI` component shows national or regional data in Table B.3. Most data are displayed using stacked area and line charts implemented with the JFreeChart open source library (Object Refinery Limited 2013). Figure B-11 shows an example for a national-level `AgricultureSystemGUI` component including the Net Revenue chart (top) and Food Source chart (bottom). Figure B-12 shows an example Network Flow components for national and regional `AgricultureSystemGUI` components.

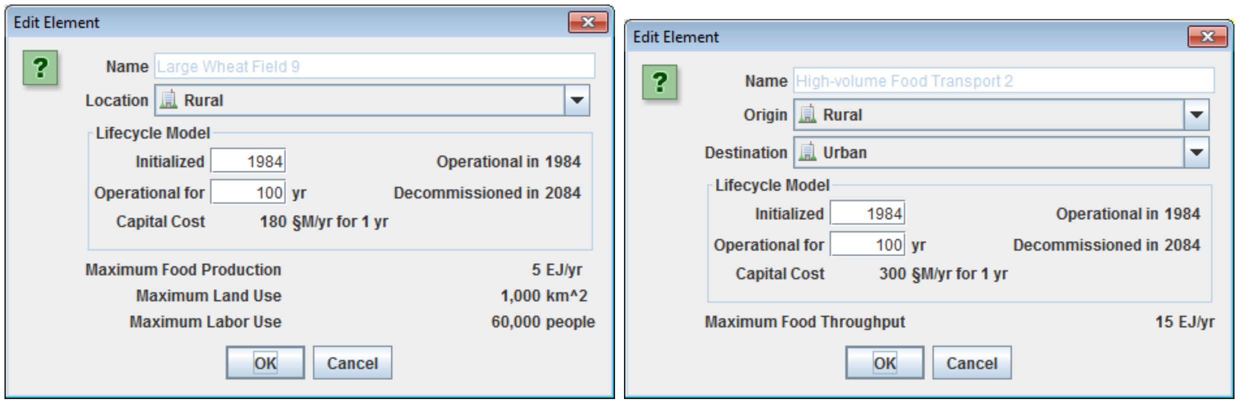

Figure B-10: Screen capture of the element GUI for an agriculture production element (left) and agriculture distribution element (right). Components show fixed attributes and editable attributes accepting user inputs.

References for Appendix B

Apache Software Foundation (2013). *Commons Math 3.2*. URL: `http://commons.apache.org/proper/commons-math/` (visited on 11/19/2013).

Object Refinery Limited (2013). *JFreeChart*. URL: `http://www.jfree.org/jfreechart/` (visited on 11/19/2013).

Table B.3: Charts for data display in infrastructure system GUI components

Data Quantity	`AgricultureSystemGUI`	`WaterSystemGUI`	`PetroleumSystemGUI`	`ElectricitySystemGUI`
$P_{capital}$	Net Revenue	Net Revenue	Net Revenue	Net Revenue
$P_{decommission}$	Net Revenue	Net Revenue	Net Revenue	Net Revenue
$P_{operations}$	Net Revenue	Net Revenue	Net Revenue	Net Revenue
$R_{produced}$	Food Source, Network Flow	Water Source, Network Flow	Oil Source, Network Flow	Electricity Source Network Flow
R_{input}	Water Use, Land Use, Labor Use	Electricity Use, Aquifer Use	Electricity Use, Reservoir Use	Water Use, Oil Use
V_{output}	Net Revenue	Net Revenue	Net Revenue	Net Revenue
R_{output}	Food Use, Network Flow	Water Use, Network Flow	Oil Use, Network Flow	Electricity Use Network Flow
$P_{distribution}$	Net Revenue	Net Revenue	Net Revenue	Net Revenue
$R_{received}$	Food Source, Network Flow	Water Source, Network Flow	Oil Source, Network Flow	Electricity Source Network Flow
$V_{distribution}$	Net Revenue	Net Revenue	Net Revenue	Net Revenue
R_{sent}	Food Use, Network Flow	Water Use, Network Flow	Oil Use, Network Flow	Electricity Use Network Flow
P_{import}	Net Revenue	Net Revenue	Net Revenue	n/a
$R_{imported}$	Food Source, Network Flow	Water Source, Network Flow	Oil Source, Network Flow	n/a
V_{export}	Net Revenue	n/a	Net Revenue	n/a
$R_{exported}$	Food Use, Network Flow	n/a	Oil Use, Network Flow	n/a

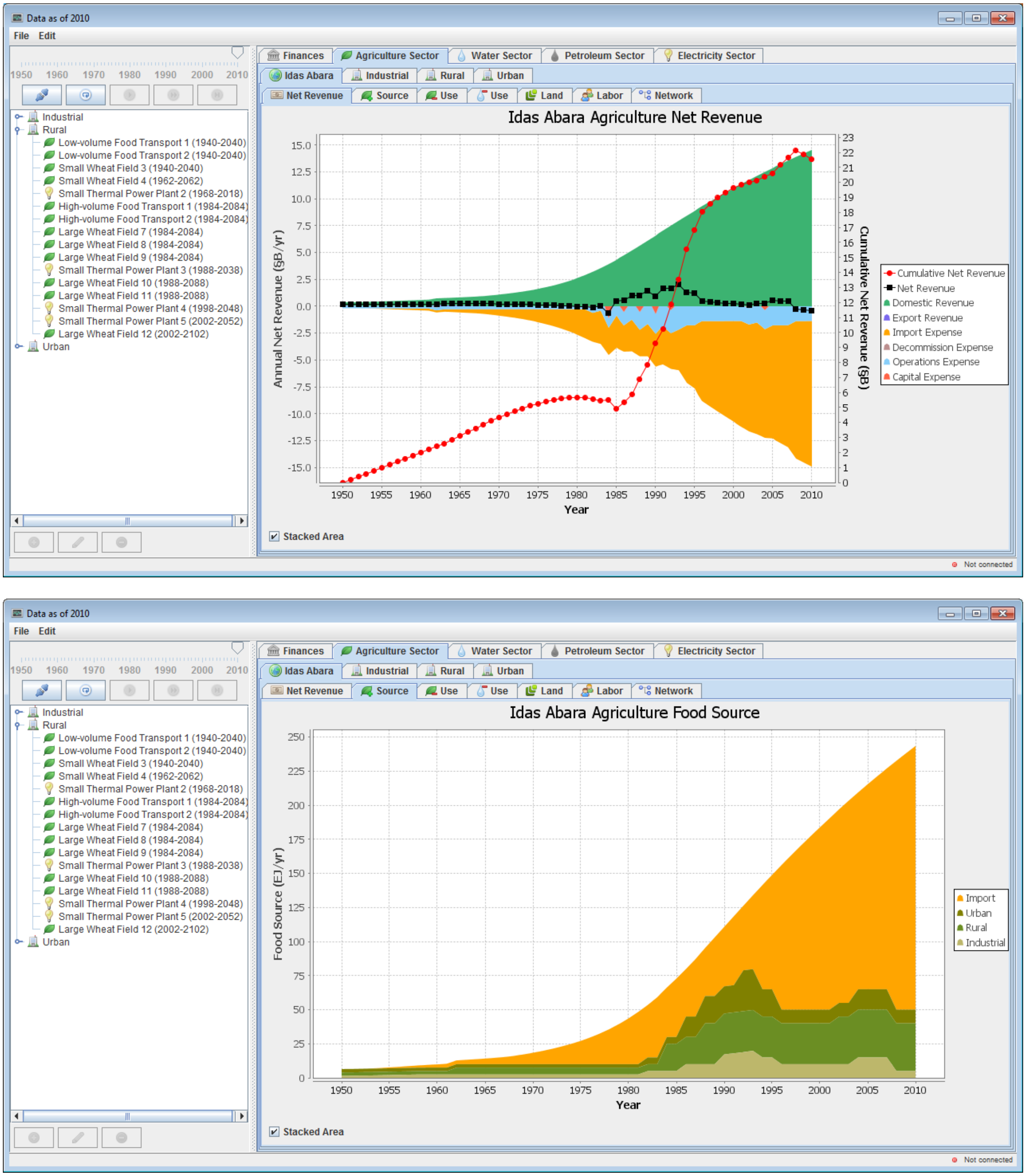

Figure B-11: Screen capture of the main SIPS-G graphical user interface includes the execution GUI component (upper left), inputs GUI component (left), and outputs GUI component (right) with nested country GUI and infrastructure system GUI components (tabs).

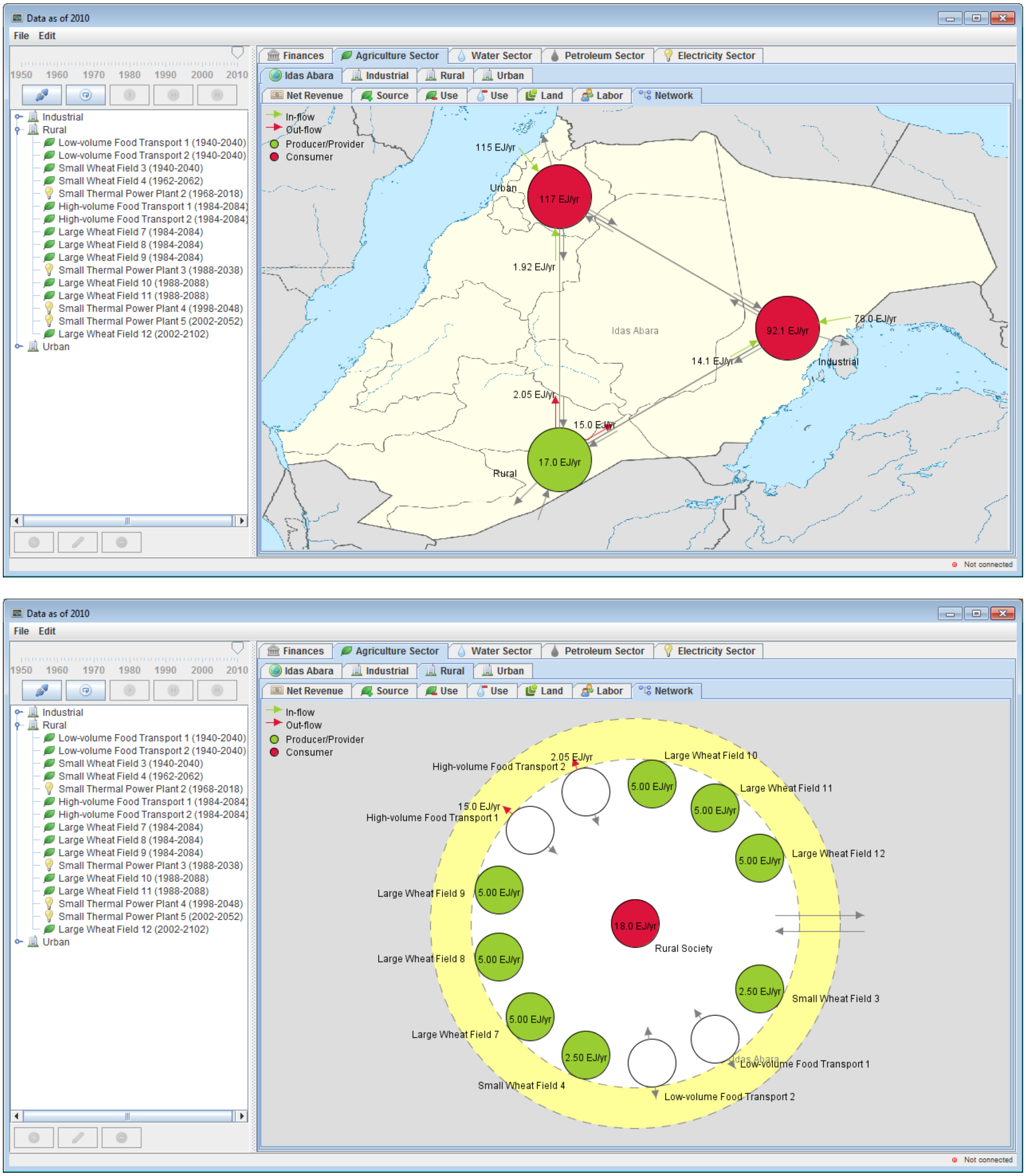

Figure B-12: Screen capture of the network flow output display for national (top) and regional (bottom) food flows. The national display shows resource flows between nodes while the regional display shows production and distribution by infrastructure elements.

Appendix C

SIPS-G Design Objective Metrics

This appendix details the objective metrics for each of the three roles—agriculture, water, and energy—and the nation as a whole. The following sections describe the formulation of each objective metric as a sum of components and illustrate the values under the baseline scenario.

C.1 Agriculture Role Objectives

The agriculture role's objectives include three components of equal weight:

1. *Food Security* ($\mathbf{S}_{food}$): measures the fraction of food supply from domestic sources.
2. *Capital Investment* ($\mathbf{I}_{MoA}$): measures the capital investment in the agriculture sector.
3. *Net Revenue* ($\mathbf{R}_{MoA}$): measures the net revenue from the agriculture sector.

The quantitative objective $\mathbf{J}_{MoA}(t)$ for year t is defined as the weighted sum

$$\mathbf{J}_{MoA}(t) = \frac{1}{3}\mathbf{S}_{food}(t) + \frac{1}{3}\mathbf{I}_{MoA}(t) + \frac{1}{3}\mathbf{R}_{MoA}(t). \tag{C.1}$$

C.1.1 Food Security

The food security metric $\mathbf{S}_{food}(t)$ measures the average fraction of domestic food supply between 1980 and year $t > 1980$ compared to a desired value of 75%. By averaging over a range of years, it frames food security as a trajectory rather than a target. It ranges between:

$\mathbf{S}_{food}(t) = 0$: no domestic food production in all years, and

$\mathbf{S}_{food}(t) = 1000$: at least 75% domestic food production in all years

and is computed for year t as

$$\mathbf{S}_{food}(t) = 1000 \cdot \frac{1}{t - 1980} \cdot \sum_{\tau=1980}^{t} F(\tau) \tag{C.2}$$

where domestic food factor $F(\tau)$ in year τ is given by a bounded linear interpolation

$$F(\tau) = \begin{cases} 1 & \text{if } D(\tau) = 0 \text{ or } S(\tau)/D(\tau) > 0.75 \\ 0 & \text{if } S(\tau)/D(\tau) < 0 \\ \frac{S(\tau)/D(\tau)}{0.75} & \text{otherwise} \end{cases} \tag{C.3}$$

where $S(\tau)$ is the domestic food supply in year τ

$$S(\tau) = Q^{food}_{produced}(\mathbf{E}_{agricul.}, \tau) \tag{C.4}$$

and $D(\tau)$ is the total food demand in year τ

$$D(\tau) = Q^{food}_{output}(\mathbf{E}_{agricul.}, \tau). \tag{C.5}$$

C.1.2 Agriculture Capital Investment

The capital investment metric $\mathbf{I}(t)$ measures the cumulative capital expenses in an infrastructure sector by year t compared to a desired value $I_{max}(t)$. It represents the motivation of a player to acquire funds from a limited national budget on capital expenditures and ranges between:

$\mathbf{I}(t) = 0$: no cumulative capital investment by year t, and

$\mathbf{I}(t) = 1000$: cumulative capital investments by year t exceeding $I_{max}(t)$.

It is computed for year t as a bounded linear interpolation

$$\mathbf{I}(t) = 1000 \cdot \begin{cases} 1 & \text{if } I(t) > I_{max}(t) \\ I(t)/I_{max}(t) & \text{otherwise} \end{cases} \tag{C.6}$$

where $I(t)$ is the cumulative capital investment in infrastructure sector i as of year t

$$I(t) = \sum_{\tau \le t} P_{capital}(\mathbf{E}_i, \tau) \tag{C.7}$$

and $I_{max}(t)$ is an upper bound on desired capital investment. The formulation of $I_{max}(t)$ uses an exponential growth computed as

$$I_{max}(t) = I_{2010} \cdot (1 + r_I)^{t-2010} \tag{C.8}$$

where I_{2010} is the desired cumulative capital investment in 2010 and r_I is the desired annual growth rate.

The agriculture capital investment metric $\mathbf{I}_{MoA}(t)$ uses the common form in Eq. C.6 in the agriculture sector with a desired value of $I_{2010} = §10$ billion and an annual growth rate of $r_I = 6\%$.

C.1.3 Agriculture Net Revenue

The net revenue metric $\mathbf{R}(t)$ measures the cumulative net revenue (i.e. cash flow) by year t compared to minimum and maximum desired values $R_{min}(t)$ and $R_{max}(t)$. It represents the motivation of a player to operate profitable infrastructure and ranges between:

$\mathbf{R}(t) = 0$: less than $R_{min}(t)$ in cumulative net revenue by year t, and

$\mathbf{R}(t) = 1000$: more than $R_{max}(t)$ in cumulative net revenue by year t.

It is computed for year t as a bounded linear interpolation

$$\mathbf{R}(t) = 1000 \cdot \begin{cases} 1 & \text{if } R(t) > R_{max}(t) \\ 0 & \text{if } R(t) < R_{min}(t) \\ \frac{R(t) - R_{min}(t)}{R_{max}(t) - R_{min}(t)} & \text{otherwise} \end{cases} \tag{C.9}$$

where $R(t)$ is the cumulative net revenue in infrastructure sector i as of year t

$$R(t) = \sum_{\tau \leq t} Q_{cashFlow}(\mathbf{E}_i, \tau) \tag{C.10}$$

and $R_{min}(t)$ and $R_{max}(t)$ are lower and upper bounds on desired net revenue. The formulation of $R_{min}(t)$ and $R_{max}(t)$ use an exponential growth computed as

$$R_{min}(t) = R_{min,2010} \cdot (1 + r_R)^{t-2010} \tag{C.11}$$

$$R_{max}(t) = R_{max,2010} \cdot (1 + r_R)^{t-2010} \tag{C.12}$$

where $R_{min,2010}$ and $R_{max,2010}$ are the lower and upper bounds on desired cumulative net revenue in 2010 and r_R is the desired annual growth rate.

The agriculture net revenue metric $\mathbf{R}_{MoA}(t)$ uses the common form in Eq. C.9 in the agriculture sector with a lower bound of $R_{min,2010} = 0$, an upper bound of $R_{max,2010} = \S 50$ billion and an annual growth rate of $r_R = 5\%$.

C.1.4 Baseline Agriculture Metrics

Figure C-1 shows the time history of agriculture objective metrics under the baseline scenario. The final value in 2010 is 43 with component scores of 119 for food security, 10 for investment, and 0 for net revenue.

C.2 Water Role Objectives

The water role's objectives include three components of equal weight:

1. *Aquifer Security* ($\mathbf{S}_{aquifer}$): measures the expected aquifer lifetime at current withdrawal rates.

2. *Capital Investment* ($\mathbf{I}_{MoW}$): measures the cumulative capital investment in the water sector.

3. *Net Revenue* ($\mathbf{R}_{MoW}$): measures the cumulative net revenue from the water sector.

The quantitative objective $\mathbf{J}_{MoW}(t)$ is defined as the weighted sum

$$\mathbf{J}_{MoW}(t) = \frac{1}{3}\mathbf{S}_{aquifer}(t) + \frac{1}{3}\mathbf{I}_{MoW}(t) + \frac{1}{3}\mathbf{R}_{MoW}(t). \tag{C.13}$$

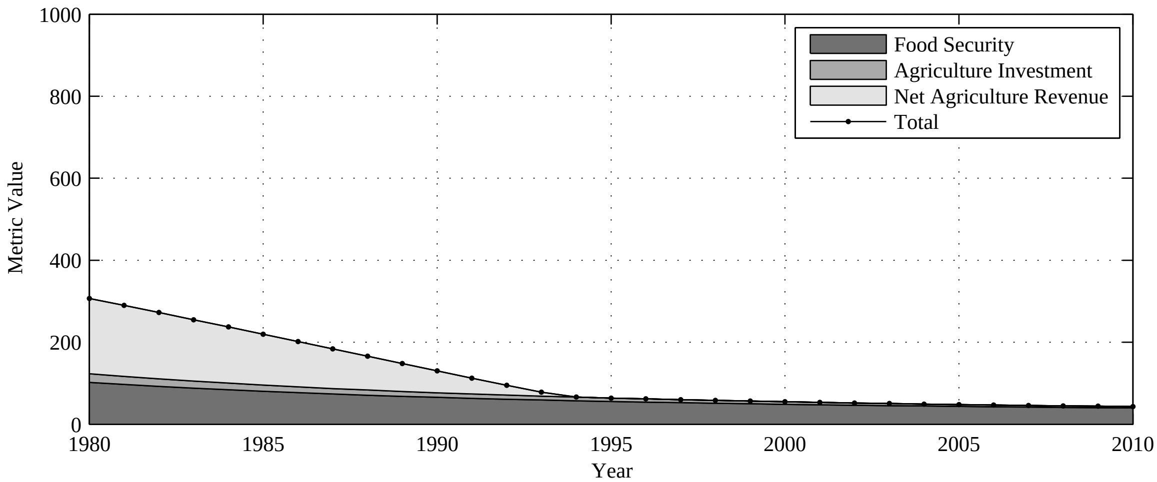

Figure C-1: Stacked area diagram for agriculture objective metrics under the baseline scenario.

C.2.1 Aquifer Security

The aquifer security metric $\mathbf{S}_{aquifer}(t)$ measures the average expected lifetime of an aquifer between 1980 and year $t > 1980$ compared to a desired values of 200 years. By averaging over a range of years, it frames aquifer security as a trajectory rather than a target. It ranges between:

$\mathbf{S}_{aquifer}(t) = 0$: expected lifetime is less than 20 years in each year between 1980 and t, and

$\mathbf{S}_{aquifer}(t) = 1000$: expected lifetime is more than 200 years in each year between 1980 and t

and is computed for year t as

$$\mathbf{S}_{aquifer}(t) = 1000 \cdot \frac{1}{t - 1980} \cdot \sum_{\tau=1980}^{t} L_a(\tau) \tag{C.14}$$

where the aquifer lifetime factor $L_a(\tau)$ for year τ is given by a bounded linear interpolation

$$L_a(\tau) = \begin{cases} 1 & \text{if } W_a(\tau) = 0 \text{ or } V_a(\tau)/W_a(\tau) > 200 \\ 0 & \text{if } V_a(\tau)/W_a(\tau) < 20 \\ \frac{V_a(\tau)/W_a(\tau) - 20}{200 - 20} & \text{otherwise} \end{cases} \tag{C.15}$$

where $V_a(\tau)$ is the total remaining aquifer volume in year τ

$$V_a(\tau) = Q_{stock}^{aquifer}(\mathbf{E}_{water}, \tau) \tag{C.16}$$

and $W_a(\tau)$ is the volume of water withdrawn from an aquifer in year τ

$$W_a(\tau) = Q_{retrieved}^{aquifer}(\mathbf{E}_{water}, \tau). \tag{C.17}$$

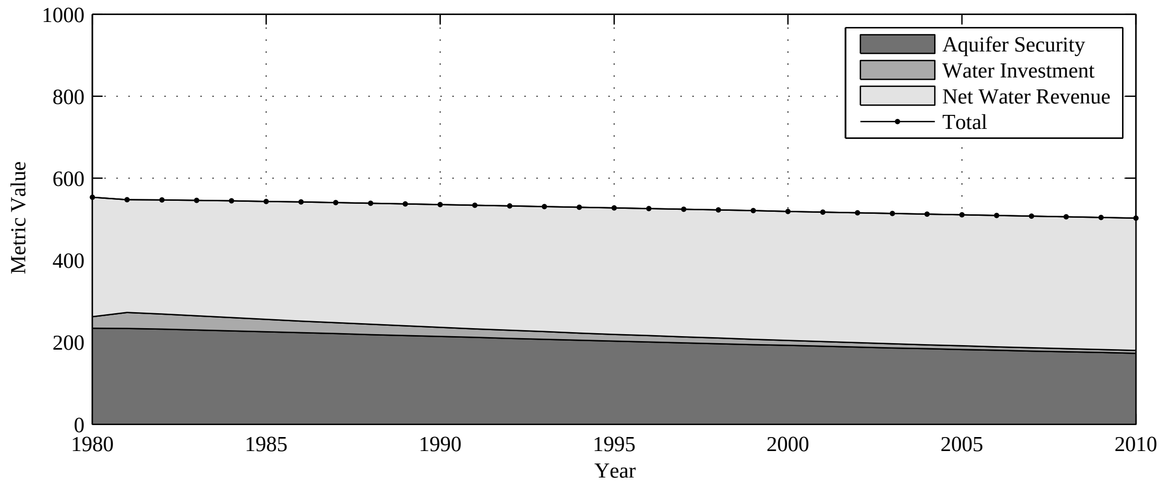

Figure C-2: Stacked area diagram for water objective metrics under the baseline scenario.

C.2.2 Water Capital Investment

The water capital investment metric $\mathbf{I}_{MoW}(t)$ uses the common form in Eq. C.6 in the water sector with a desired value of $I_{2010} = \S 15$ billion and an annual growth rate of $r_I = 6\%$.

C.2.3 Water Net Revenue

The water net revenue metric $\mathbf{R}_{MoW}(t)$ uses the common form in Eq. C.9 in the water sector with a lower bound of $R_{min,2010} = -\S 10$ billion, an upper bound of $R_{max,2010} = 0$ and an annual growth rate of $r_R = 6\%$. This formulation recognizes the national water sector as subsidy-based and prefers smaller losses.

C.2.4 Baseline Water Metrics

Figure C-2 shows the time history of water objective metrics under the baseline scenario. The final value in 2010 is 503 with component scores of 519 for aquifer security, 20 for investment, and 969 for net revenue.

C.3 Energy Role Objectives

The energy role's objectives include three components of equal weight:

1. *Reservoir Security* ($\mathbf{S}_{reservoir}$): measures the expected oil reservoir lifetime at current extraction rates.

2. *Capital Investment* ($\mathbf{I}_{MoE}$): measures the cumulative capital investment in the oil and electricity sectors.

3. *Net Revenue* ($\mathbf{R}_{MoE}$): measures the cumulative net revenue from the oil and electricity sectors.

The quantitative objective $\mathbf{J}_{MoE}(t)$ is defined as the weighted sum

$$\mathbf{J}_{MoE}(t) = \frac{1}{3}\mathbf{S}_{reservoir}(t) + \frac{1}{3}\mathbf{I}_{MoE}(t) + \frac{1}{3}\mathbf{R}_{MoE}(t). \tag{C.18}$$

C.3.1 Oil Reservoir Security

The oil reservoir security metric $\mathbf{S}_{reservoir}(t)$ measures the average expected lifetime of the oil reservoir between 1980 and year $t > 1980$ compared to a desired value of 200 years. By averaging over a range of years, it frames reservoir security as a trajectory rather than a target. It ranges between:

$\mathbf{S}_{reservoir}(t) = 0$: no remaining reservoir in each year between 1980 and t, and

$\mathbf{S}_{reservoir} = 1000$: expected lifetime is more than 200 years in each year between 1980 and t.

and is computed for year t as

$$\mathbf{S}_{reservoir}(t) = 1000 \cdot \frac{1}{t - 1980} \cdot \sum_{\tau=1980}^{t} L_r(\tau) \tag{C.19}$$

where the reservoir lifetime factor $L_r(\tau)$ for year τ is given by a bounded linear interpolation

$$L_r(\tau) = \begin{cases} 1 & \text{if } W_r(\tau) = 0 \text{ or } V_r(\tau)/W_r(\tau) > 200 \\ 0 & \text{if } V_r(\tau)/W_r(\tau) < 0 \\ \frac{V_r(\tau)/W_r(\tau)}{200} & \text{otherwise} \end{cases} \tag{C.20}$$

where $V_r(\tau)$ is the total remaining reservoir volume in year τ

$$V_r(\tau) = Q_{stock}^{reservoir}(\mathbf{E}_{petrol.}, \tau) \tag{C.21}$$

and $W_r(\tau)$ is the volume of oil extracted from a reservoir in year τ

$$W_r(\tau) = Q_{retrieved}^{reservior}(\mathbf{E}_{petrol.}, \tau). \tag{C.22}$$

C.3.2 Energy Capital Investment

The energy capital investment metric $\mathbf{I}_{MoE}(t)$ uses the common form in Eq. C.6 in the petroleum and electricity sectors with a desired value of $I_{2010} = \S 500$ billion and an annual growth rate of $r_I = 4\%$.

C.3.3 Energy Net Revenue

The energy net revenue metric $\mathbf{R}_{MoE}(t)$ uses the common form in Eq. C.9 in the petroleum and electricity sectors with a lower bound of $R_{min,2010} = 0$, an upper bound of $R_{max,2010} = \S 50$ billion and an annual growth rate of $r_R = 4\%$.

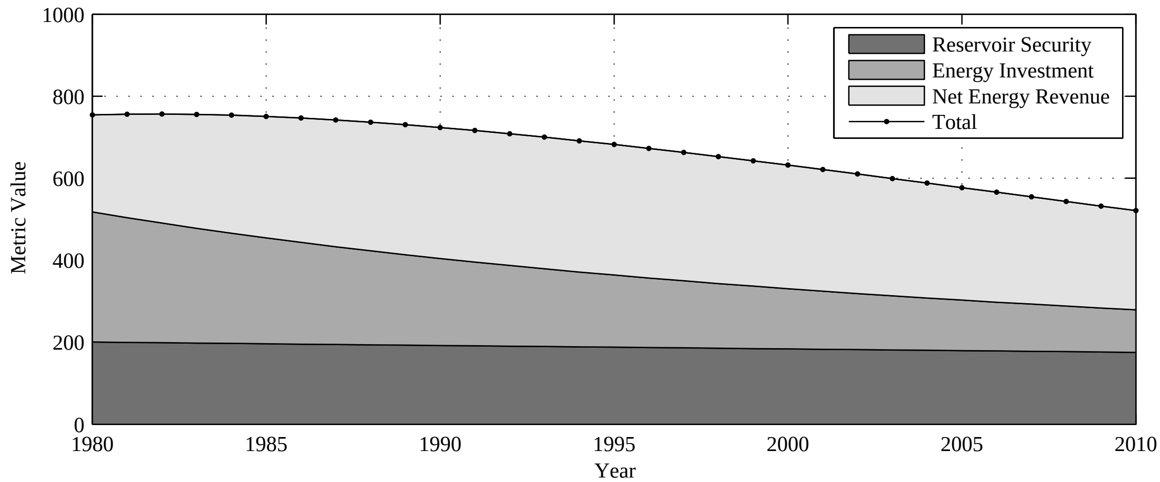

Figure C-3: Stacked area diagram for energy objective metrics under the baseline scenario.

C.3.4 Baseline Energy Metrics

Figure C-3 shows the time history of energy objective metrics under the baseline scenario. The final value in 2010 is 521 with component scores of 526 for reservoir security, 312 for investment, and 725 for net revenue.

C.4 National Objectives

The national objectives includes four components of equal weight:

1. *Food Security* ($\mathbf{S}_{food}$): measures the fraction of food supply from domestic sources.

2. *Aquifer Security* ($\mathbf{S}_{aquifer}$): measures the expected aquifer lifetime at current withdrawal rates.

3. *Reservoir Security* ($\mathbf{S}_{reservoir}$): measures the expected oil reservoir lifetime at current extraction rates.

4. *Net Revenue* ($\mathbf{R}_{IA}$): measures the total net revenue of all infrastructure sectors.

The quantitative objective $\mathbf{J}_{IA}(t)$ is defined as the weighted sum

$$\mathbf{J}_{IA}(t) = \frac{1}{4}\mathbf{S}_{food}(t) + \frac{1}{4}\mathbf{S}_{aquifer}(t) + \frac{1}{4}\mathbf{S}_{reservoir}(t) + \frac{1}{4}\mathbf{R}_{IA}(t) \tag{C.23}$$

where the resource security components are drawn from each role (i.e. $\mathbf{S}_{food}(t)$ in Eq. C.2, $\mathbf{S}_{aquifer}(t)$ in Eq. C.14, and $\mathbf{S}_{reservoir}(t)$ in Eq. C.19) and the net revenue $\mathbf{R}_{IA}(t)$ is computed using the common form in Eq. C.9 for all infrastructure sectors with a lower bound of $R_{min,2010} = -\S 10$ billion, an upper bound of $R_{max,2010} = \S 550$ billion and an annual growth rate of $r_R = 4\%$.

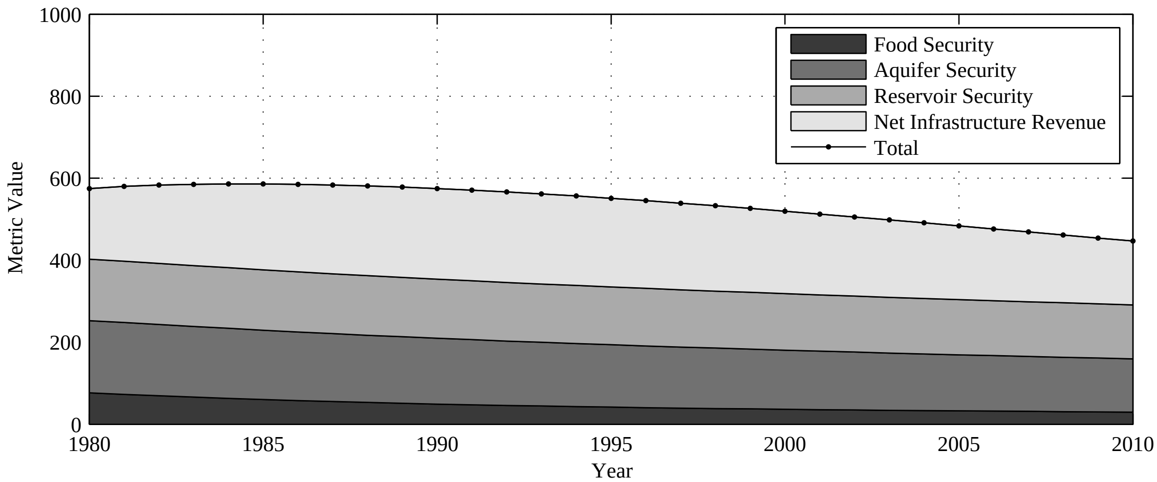

Figure C-4: Stacked area diagram for national objective metrics under the baseline scenario.

C.4.1 Baseline National Metrics

Figure C-4 shows the time history of national objective metrics under the baseline scenario. The final value in 2010 is 447 with component scores of 119 for food security, 519 for aquifer security, 526 for reservoir security, and 622 for net revenue.

C.5 Analysis of Objectives

Table C.1 summarizes causal effects between objective components based on the underlying infrastructure system models where "+" indicates a positive causation between row A and column B (i.e. an increase in A causes an increase in B) and "−" indicates a negative causation between row A and column B. Only nominal interactions are illustrated which may change at varying levels. For example, additional agriculture investment would not improve food security if it exceeds available land or labor capacities. Figure C-5 illustrates the data as a causal loop diagram.

Table C.1: Nominal causal interactions between objectives

	$\mathbf{S}_{food}$	$\mathbf{I}_{MoA}$	$\mathbf{R}_{MoA}$	$\mathbf{S}_{aquifer}$	$\mathbf{I}_{MoW}$	$\mathbf{R}_{MoW}$	$\mathbf{S}_{reservoir}$	$\mathbf{I}_{MoE}$	$\mathbf{R}_{MoE}$	$\mathbf{R}_{IA}$	$\mathbf{J}_{MoA}$	$\mathbf{J}_{MoW}$	$\mathbf{J}_{MoE}$	$\mathbf{J}_{IA}$
$\mathbf{S}_{food}$	1										+			+
$\mathbf{I}_{MoA}$	+	1	+	−	−			−		+	+			
$\mathbf{R}_{MoA}$			1								+			
$\mathbf{S}_{aquifer}$				1								+		+
$\mathbf{I}_{MoW}$		−		+	1	−		−	−	−		+		
$\mathbf{R}_{MoW}$						1						+		
$\mathbf{S}_{reservoir}$							1						+	+
$\mathbf{I}_{MoE}$		−			−		$-^{\star}$	1	+	+			+	
$\mathbf{R}_{MoE}$									1				+	
$\mathbf{R}_{IA}$										1				+
$\mathbf{J}_{MoA}$											1			
$\mathbf{J}_{MoW}$												1		
$\mathbf{J}_{MoE}$													1	
$\mathbf{J}_{IA}$														1

$\star$ only for investment in oil wells

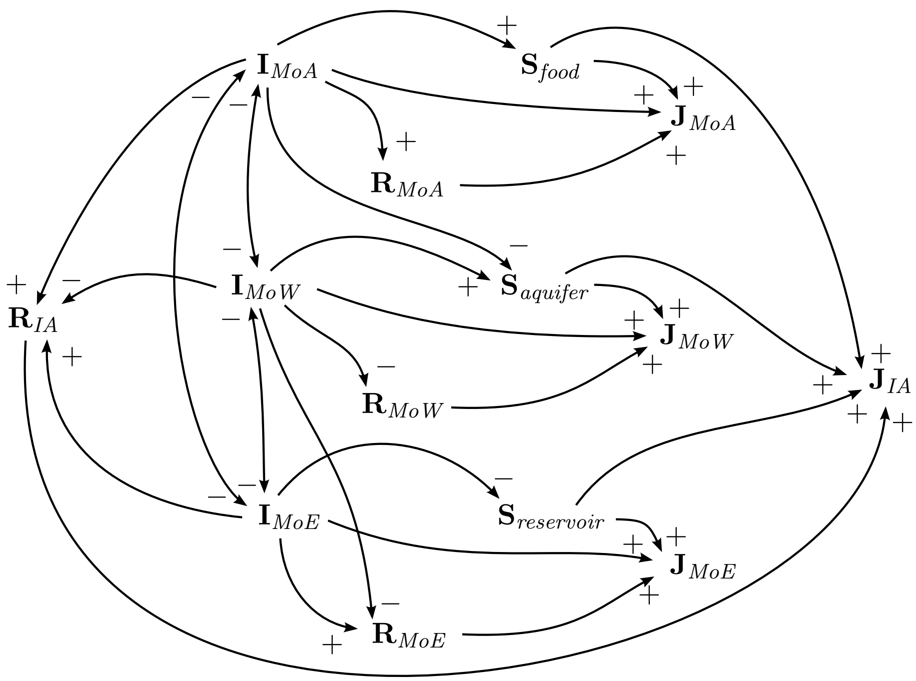

Figure C-5: Causal loop diagram of nominal interactions between objectives.

Investment in agriculture infrastructure ($\mathbf{I}_{MoA}$) contributes to greater food security ($\mathbf{S}_{food}$) from increased domestic production and higher revenues ($\mathbf{R}_{MoA}$ and $\mathbf{R}_{IA}$) by avoiding import expenses. Allocating budget to the agriculture sector limits investment in other infrastructure ($\mathbf{I}_{MoW}$ and $\mathbf{I}_{MoE}$) and consumes more water to reduce aquifer security ($\mathbf{S}_{aquifer}$). The net effect of agriculture investment on the individual objective ($\mathbf{J}_{MoA}$) is positive; however the effect on the national objective ($\mathbf{J}_{IA}$) is mixed due to indirect budget and aquifer security effects.

Investment in water infrastructure ($\mathbf{I}_{MoW}$) contributes to greater aquifer security ($\mathbf{S}_{aquifer}$) from increased desalination capacity and lower revenues ($\mathbf{R}_{MoW}$ and $\mathbf{R}_{IA}$) due to water subsidies. Allocating budget for the water sector limits investment in other infrastructure ($\mathbf{I}_{MoW}$ and $\mathbf{I}_{MoE}$) and consumes more electricity which generates less revenue than the alternative of oil export ($\mathbf{R}_{MoE}$). The net effects of water investment on both individual ($\mathbf{J}_{MoW}$) and national objectives ($\mathbf{J}_{IA}$) are mixed due to direct aquifer security and revenue effects and indirect budget and energy revenue effects.

Investment in energy infrastructure ($\mathbf{I}_{MoE}$) has different effects depending on the specific element selected. Oil wells contribute to lower reservoir security ($\mathbf{S}_{reservoir}$) from increased oil extraction and higher revenues ($\mathbf{R}_{MoE}$ and $\mathbf{R}_{IA}$) from to oil export. Thermal power plants contribute to higher oil export and electricity revenues while solar PV plants contribute to higher oil export revenues but lower electricity revenues. Both thermal and solar PV plants contribute a net positive effect on energy revenue ($\mathbf{R}_{MoE}$ and $\mathbf{R}_{IA}$). Allocating budget for the energy sector limits investment in other infrastructure ($\mathbf{I}_{MoA}$ and $\mathbf{I}_{MoW}$). The net effects of energy investment on the individual ($\mathbf{J}_{MoE}$) and national objectives ($\mathbf{J}_{IA}$) are mixed but generally positive due to direct reservoir security and revenue effects and indirect budget effects.

Tensions between achieving individual role versus joint national objectives arise from two key sources. The first tension emerges from interactions between the agriculture and water sectors. Increased domestic food production to improve food security has large impacts on water withdrawals, leading to lower aquifer security. While investment in desalination capacity can offset a portion of the water demand, it leads to the second tension between water and energy sectors. As an energy-intensive process, desalination increases electricity demands which is fueled by domestic oil consumption. Because increasing oil production reduces reservoir security, efficient sources electricity generation include thermal power plants or solar PV plants which trade higher capital expenses for reduced domestic oil consumption. As desalination is also a capital-intensive project, the annual budget limit is quickly reached with moderate investment in solar PV plants.

This page is intentionally left blank.

Appendix D

SIPS-G Experimental Data

Tables D.1–D.4 summarize results from the final round of each design session for Agriculture, Water, Energy, and National objectives. Each table shows the final resource production capacity summed across all regions and forms (e.g. thermal and solar power), its percentage of the maximum value encountered during the design session, and the resulting objective metric components.

Figures D-1–D-5 plot infrastructure inputs (summarized by normalized capacity in each sector) and objective metric outcomes after each data exchange. Infrastructure capacities are normalized by the following median values from Tables D.1–D.3 (Wheat: 100 EJ/yr, Desalination: 8400 MCM/yr, Oil: 700 Mtoe/yr, and Power: 204 TWh/yr).

Table D.1: Final round agriculture results by session

Session	Wheat Cap. (EJ/yr)	% of Max	$\mathbf{S}_{food}$	$\mathbf{I}_{MoA}$	$\mathbf{R}_{MoA}$	$\mathbf{J}_{MoA}$
1	85.0	100%	742.6	400.0	993.0	711.8
2	130.0	84%	851.2	1000.0	1000.0	950.4
3	75.0	100%	556.1	274.0	578.5	469.5
4	102.5	100%	515.4	442.0	511.6	489.6
5	210.0	98%	889.3	1000.0	918.8	936.0
6	90.0	69%	686.8	478.0	823.5	662.8
7	90.0	100%	622.4	448.0	769.7	613.4
8	95.0	90%	675.6	406.0	880.7	654.1
9	75.0	71%	427.5	304.0	361.4	364.3
10	130.0	93%	762.1	622.0	1000.0	794.7
11	150.0	79%	626.0	544.0	703.9	624.6
12	140.0	90%	644.6	568.0	760.6	657.7
13	80.0	100%	471.0	262.0	470.2	401.1
14	100.0	95%	784.9	424.0	1000.0	736.3
15	100.0	95%	595.6	394.0	722.5	570.7

Table D.2: Final round water results by session

Session	Desal. Cap. (MCM/yr)	% of Max	$\mathbf{S}_{aquifer}$	$\mathbf{I}_{MoW}$	$\mathbf{R}_{MoW}$	$\mathbf{J}_{MoW}$
1	10250	100%	32.0	1000.0	0.0	344.0
2	11900	100%	21.2	1000.0	0.0	340.4
3	14300	100%	135.2	1000.0	0.0	378.4
4	3200	100%	102.1	1000.0	0.0	367.4
5	3650	67%	35.5	1000.0	0.0	345.2
6	5300	49%	32.6	1000.0	0.0	344.2
7	2600	100%	66.3	870.0	0.0	312.1
8	11600	100%	43.5	1000.0	0.0	347.8
9	11600	95%	199.9	1000.0	0.0	400.0
10	7850	100%	59.1	1000.0	0.0	353.0
11	10850	100%	56.5	1000.0	0.0	352.2
12	8400	100%	65.5	1000.0	0.0	355.2
13	7400	100%	152.0	1000.0	0.0	384.0
14	6800	100%	28.2	1000.0	0.0	342.7
15	10250	100%	78.5	1000.0	0.0	359.5

Table D.3: Final round energy results by session

Session	Oil Cap. (Mtoe/yr)	% Max	Elect. Cap. (TWh/yr)	% Max	% Renew.	$\mathbf{S}_{reservoir}$	$\mathbf{I}_{MoE}$	$\mathbf{R}_{MoE}$	$\mathbf{J}_{MoE}$
1	1200	100%	234	100%	30%	415.1	967.0	953.8	778.6
2	500	100%	94	100%	80%	526.3	563.5	710.8	600.2
3	500	100%	234	100%	91%	526.3	878.5	725.2	710.0
4	800	100%	234	100%	4%	433.6	640.0	891.6	655.0
5	700	100%	234	100%	74%	474.6	947.5	805.8	742.7
6	1700	100%	204	100%	50%	338.5	1000.0	1000.0	779.5
7	600	100%	164	100%	88%	474.4	796.0	796.2	688.9
8	500	100%	204	100%	92%	526.3	923.5	717.6	722.4
9	450	90%	294	100%	3%	542.1	491.5	719.6	584.4
10	375	75%	150	100%	89%	621.0	743.5	626.5	663.7
11	700	100%	114	100%	36%	526.3	583.0	699.3	602.8
12	1100	100%	244	100%	40%	393.7	985.0	981.9	786.9
13	500	100%	74	100%	57%	526.3	496.0	712.8	578.4
14	800	100%	194	100%	63%	408.2	869.5	895.5	724.4
15	1000	100%	264	100%	43%	376.9	1000.0	1000.0	792.3

Table D.4: Final round national results by session

Session	$\mathbf{S}_{food}$	$\mathbf{S}_{aquifer}$	$\mathbf{S}_{reservoir}$	$\mathbf{R}_{IA}$	$\mathbf{J}_{IA}$
1	742.6	32.0	415.1	867.3	514.2
2	851.2	21.2	526.3	0.0	349.7
3	556.1	135.2	526.3	579.7	449.3
4	515.4	102.1	433.6	818.1	467.3
5	889.3	35.5	474.6	0.0	349.9
6	686.8	32.6	338.5	981.8	509.9
7	622.4	66.3	474.4	774.4	484.4
8	675.6	43.5	526.3	622.4	466.9
9	427.5	199.9	542.1	584.5	438.5
10	762.1	59.1	621.0	581.2	505.8
11	626.0	56.5	526.3	170.5	344.8
12	644.6	65.5	393.7	887.9	497.9
13	471.0	152.0	526.3	630.9	445.1
14	784.9	28.2	408.2	847.0	517.1
15	595.6	78.5	376.9	896.6	486.9

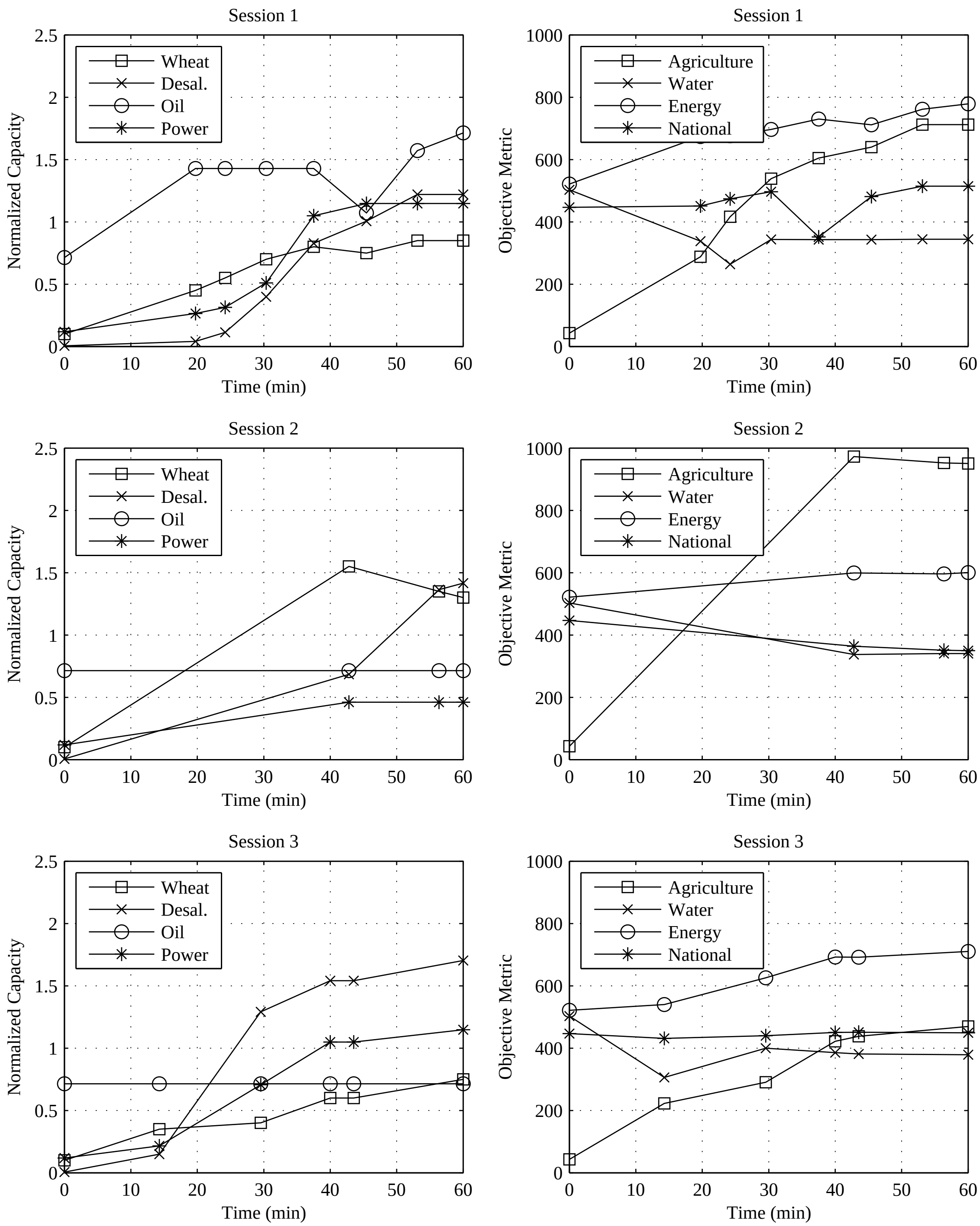

Figure D-1: Normalized input and objective metric time histories for sessions 1–3.

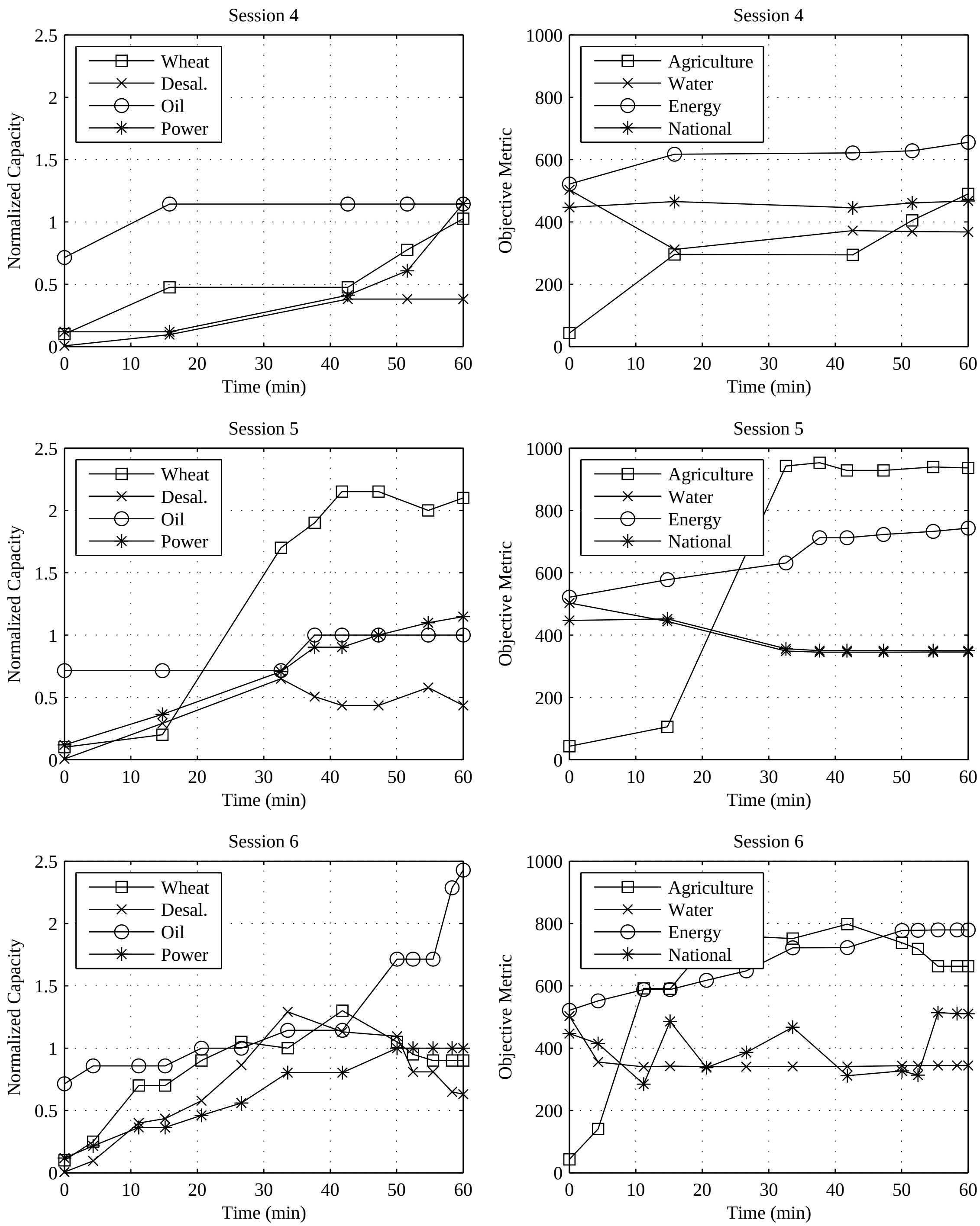

Figure D-2: Normalized input and objective metric time histories for sessions 4–6.

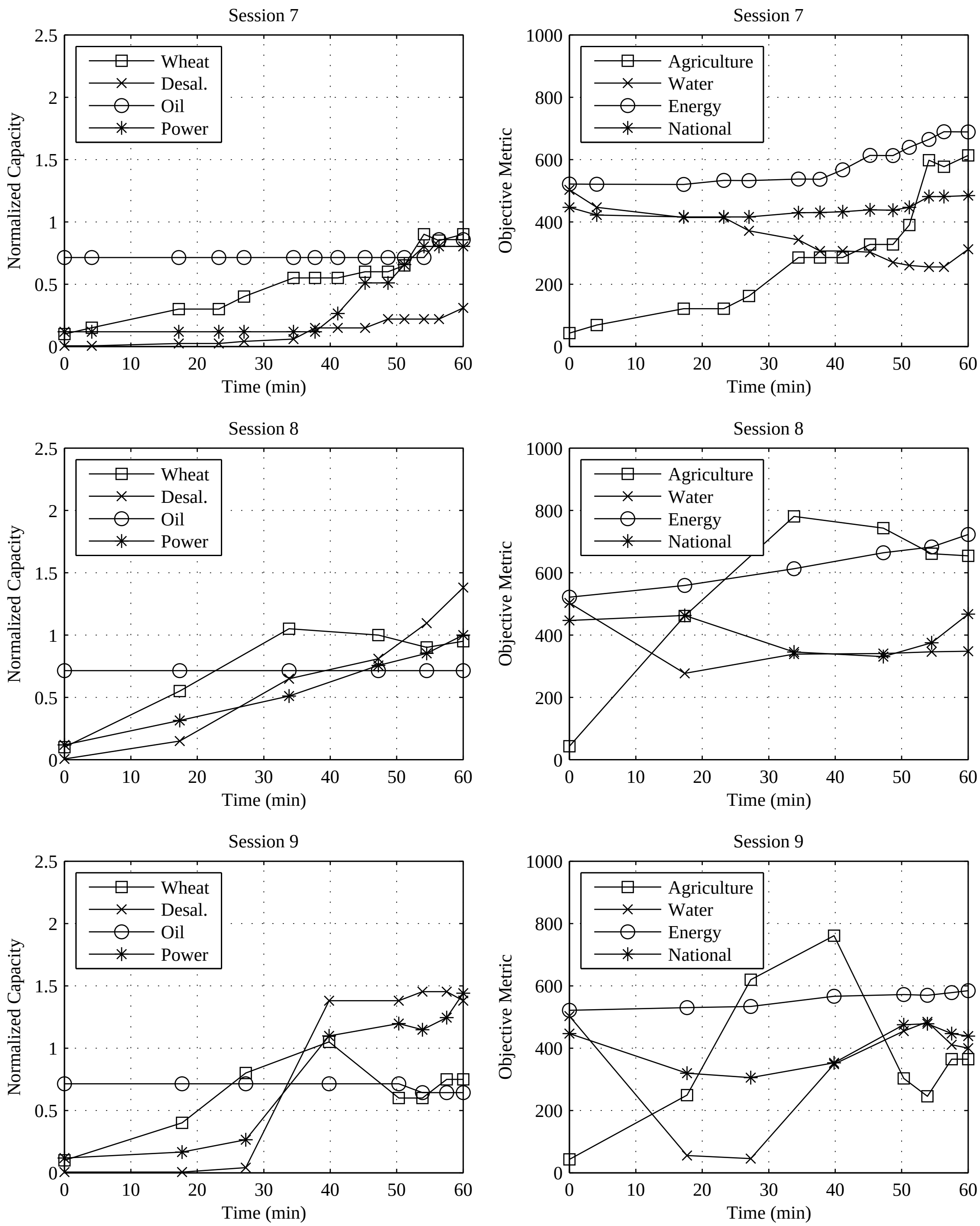

Figure D-3: Normalized input and objective metric time histories for sessions 7–9.

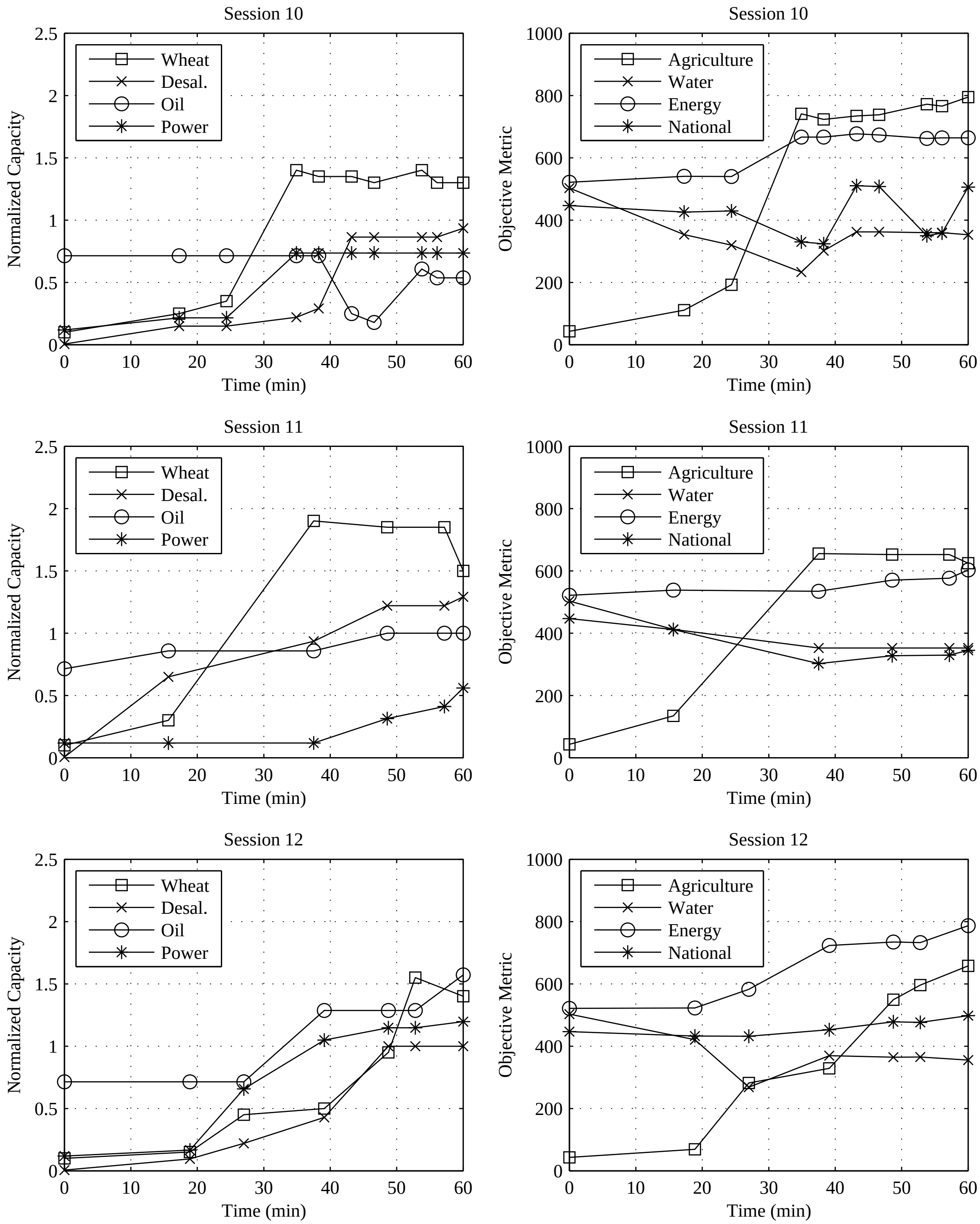

Figure D-4: Normalized input and objective metric time histories for sessions 10–12.

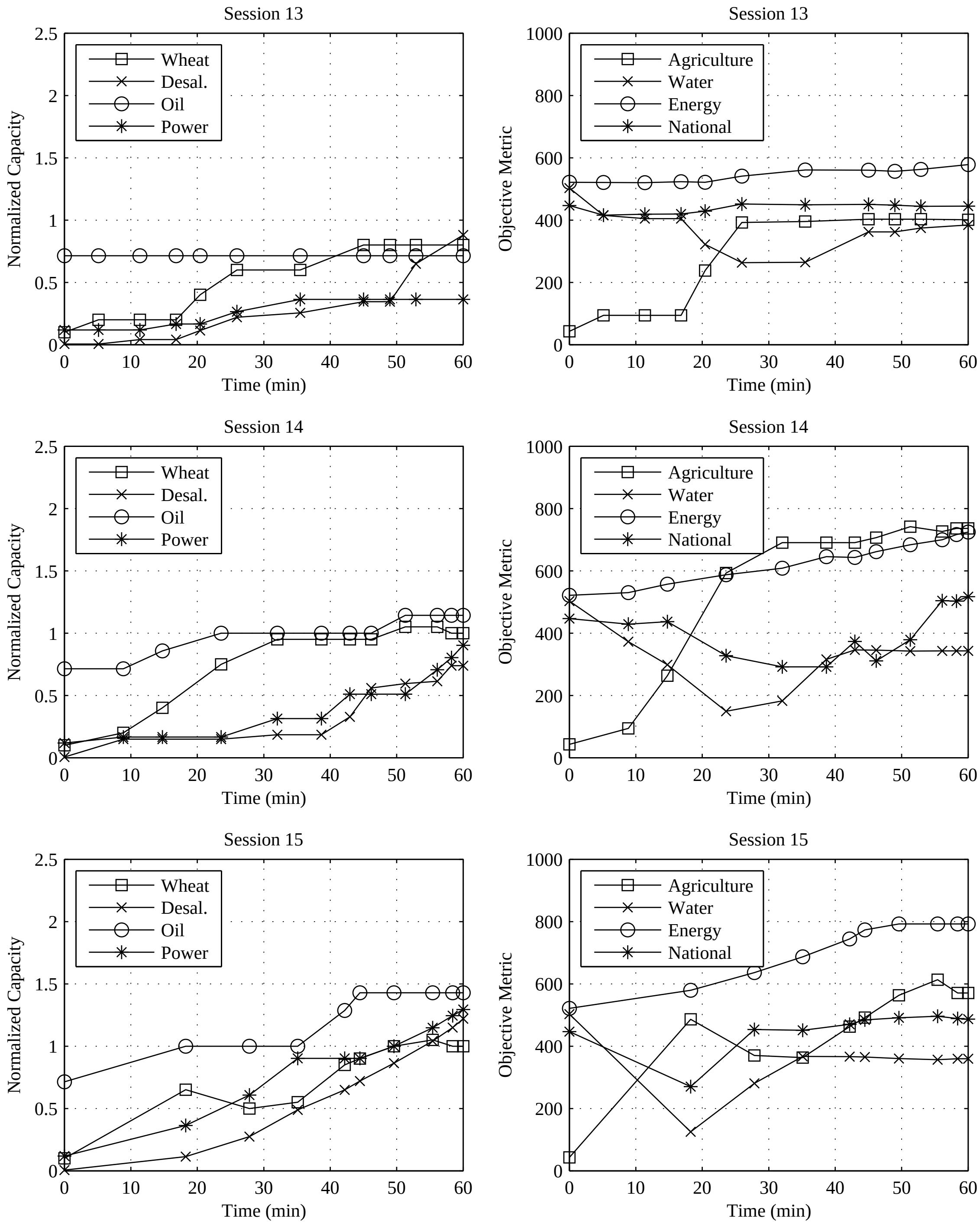

Figure D-5: Normalized input and objective metric time histories for sessions 13–15.

Biographical Note

Paul Thomas Grogan was born in Madison, Wisconsin in 1985. He attended primary school in Oshkosh, Wisconsin and high school in Middleton, Wisconsin. Paul graduated from Middleton High School in 2003 as an All-State Scholar, receiving a Robert C. Byrd Honors Scholarship. He enrolled at the University of Wisconsin–Madison and received a bachelor's degree in Engineering Mechanics & Astronautics and Mathematics in 2008 with engineering honors in liberal arts and a certificate in Computer Sciences.

Paul enrolled in the MIT Department of Aeronautics and Astronautics for graduate studies where he worked in support of the NASA Constellation program to develop a flexible modeling and simulation tool to study logistics of human space exploration campaigns. He received a master's degree in Aeronautics and Astronautics in 2010 and enrolled in the MIT Engineering Systems Division with the support of a National Defense Science and Engineering Graduate Fellowship from the U.S. Department of Defense. During his doctoral program he worked on infrastructure modeling projects for Masdar City and Saudi Arabia. His dissertation develops interoperable simulation games as a method to combine technical system models with social actors and address the major sources of complexity in infrastructure systems design. He received a doctoral degree in Engineering Systems in 2014.

Paul aspires to teach and lead research as a faculty member at a major university. He is most concerned for the future of critical energy, water, and food resources systems on which all societies rely. He hopes to continue applying interactive, model-oriented methods combining technical and social elements to address the dual challenges of integration and collaboration among systems.