Total System Modeling:
A System Engineering Application of the Higraph Formalism

Presented by Kevin Fogarty, SAIC

Total System Modeling

- This presentation will examine the concept of Total System Modeling to promote the development and use of a complete and correct system model that can be used to support the planning, design, integration, deployment, operation, and maintenance of a system.

Total System Modeling: Presentation Outline

- Introduction
- Systems of Systems
- Current Modeling Tools
- Use of Higraphs for System Modeling
- Practical Implementation of a Higraph-Based Model
- Benefits and Conclusions

Total System Modeling: Introduction

- Kevin Fogarty
  - Senior Systems Engineer with SAIC
  - Degrees from Virginia Tech and the University of Maryland, College Park
  - Experience supporting the Intelligence Community
  - Specializes in Software Systems Integration
- This presentation is an extension of thesis work and a paper published in the INCOSE Journal: Systems Engineering
  - Co-authored with Dr. Mark Austin, Institute for Systems Research, University of Maryland

Total System Modeling: Systems of Systems

- What is “Systems of Systems” (SoS)?
  - “Systems of systems are large scale concurrent and distributed systems that are comprised of complex systems” *[Kotov]*
  - “SoS is a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities” *[Department of Defense-4]*
  - “In relation to joint warfighting, system of systems is concerned with interoperability and synergism of Command, Control, Computers, Communications, and Information (C4I) and Intelligence, Surveillance, and Reconnaissance (ISR) Systems” *[Manthorpe]*

Total System Modeling: Systems of Systems (Cont'd)

- Within the Defense Industry:
  - Individual projects are themselves Systems of Systems:
    - Joint Strike Fighter (JSF)
    - KC-X Refueling Tanker
  - Defining the relationships between these two programs is a required System of Systems effort
    - From the basic, "The tanker will work with the JSF, right?"
    - To the complex, "What if the KC-X award is protested, and the protest is upheld, and the procurement schedule slips?"
  - Entire programs/policies/initiatives are defined as Systems of Systems
    - Future Combat Systems
    - Net Centric Warfare, "… generating combat power from the effective linking or networking of the warfighting enterprise …" *[Department of Defense-3]*
    - Net Centric Operations, "… generates increased combat power by networking sensors, decision makers, and shooters …" *[Alberts]*

Total System Modeling: Systems of Systems (Cont'd)

- As Systems Engineers we require a method to model these Systems of Systems—success depends on it!
- The model should be useful in all phases of the system lifecycle
  - From the usual requirements, design, integration, deployment, operations, and maintenance models,
  - To the budget, schedule, and staffing models,
  - To models of external dependencies such as technical domain rules, legislative realities, and modern political situations
- And all of these models need to be connected to form a Total System Model

Total System Modeling: Current Modeling Tools

- UML
  - "The objective of UML is to provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of software based systems as well as for modeling business and similar processes." *[The Object Management Group]*
  - A frequent criticism is that UML focuses too much on (software) system design
    - At a minimum, there is a need for better requirements modeling, as shown in *[Leitler]*
    - There is also a weakness in supporting lifecycle modeling

Total System Modeling: Current Modeling Tools (Cont'd)

- SysML
  - "SysML supports the specification, analysis, design, and verification and validation of a broad range of complex systems." *[SysML Partners]*
  - With SysML, there is support for connectivity between diagrams, and weak definition for hierarchy with the <<allocate>> tag *[Fogarty-5]*

- UML and SysML do an increasingly better job of modeling systems, but not necessarily Systems of Systems

Total System Modeling: Current Modeling Tools (Cont'd)

- DODAF
  - "The DoDAF provides the guidance and rules for developing, representing, and understanding architectures based on a common denominator across DoD, Joint, and multinational boundaries." *[Department of Defense-3]*
  - As such, it improves the connectivity issues seen in UML and SysML by providing a real framework to model Systems of Systems
    - Allows for Integrated and Federated architectures

Total System Modeling: Current Modeling Tools (Cont'd)

- DODAF Considerations
  - Critiques have been raised regarding non-specification of standard modeling notations and its impediment to "architecture coordination across programs" *[Department of Defense-2]*
    - UML, IDEF/IDEF1X, pure data repositories (databases), free-form documents (diagrams, pictures) may represent a view
  - *[Richards]* cites the practical limitations of using DODAF where a "companion architecture development process" is not specified, and contractors often treat DODAF models as "contract deliverable as opposed to a central communications tool in the design process"
    - "The U.S. Army Future Combat System's architecture requires interoperability of 1,540 systems, 10,000 DoDAF work products, and 800,000 information exchanges" *[Jain]*

Total System Modeling: Current Modeling Tools (Cont'd)

- What we really need is a way to link modeling languages and frameworks because:
  - No language or framework will ever be universally used
  - All of our current modeling tools and frameworks will continue to evolve… and will likely be replaced!
    - *C4ISR Architecture Framework v1.0, 7 June 1996*
    - *C4ISR Architecture Framework v2.0, 18 December 1997*
    - *DoD Architecture Framework v1.0, 30 August 2003*
    - *DoD Architecture Framework v1.5, 23 April 2007*
    - *DoD Architecture Framework v2.0, TBD*
  - Our current modeling tools and frameworks could be considered "complex"
    - UML 2.1.2 Superstructure: 738 pages
    - SysML 1.0: 258 pages
    - DODAF 1.5, Volumes I-III: 553 pages
- **We need to link current models, legacy models, and unique data repositories, and we must be able to adjust and customize our Total System Modeling framework to support lifecycle system engineering activities for a System of Systems**

Total System Modeling: Use of Higraphs for System Modeling

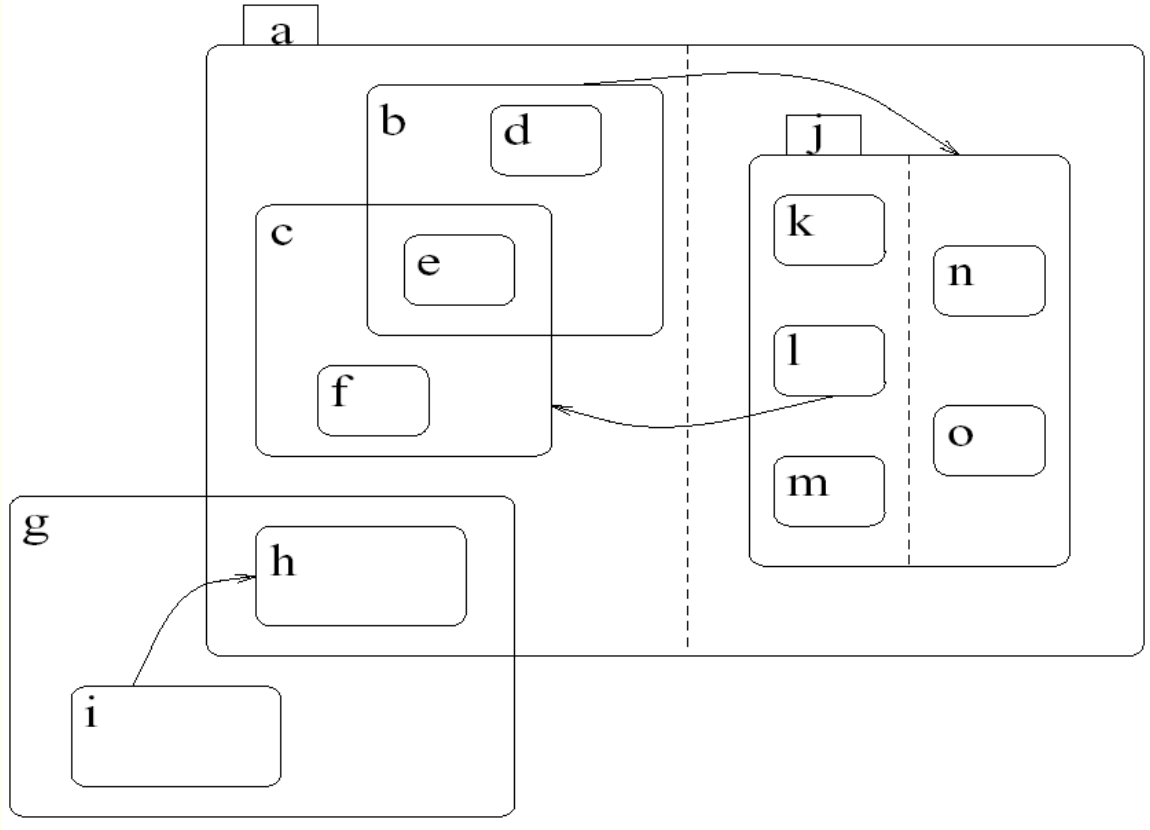
- A Higraph is a mathematical graph that combines depth and orthogonality:

  Higraph = Graph + Depth + Orthogonality *[Harel]*

- A Higraph can be defined by: *[Harel]*
  - B is the set of blobs [nodes], b, that make up a higraph
  - E is the set of edges, e, that make up a higraph
  - ρ is the hierarchy function
  - Π is the orthogonality (or partitioning) function
  - The quadruple (B, E, ρ, Π) defines a higraph, H

Total System Modeling: Use of Higraphs for System Modeling (Cont'd)

- An example from *[Harel]* is shown below:

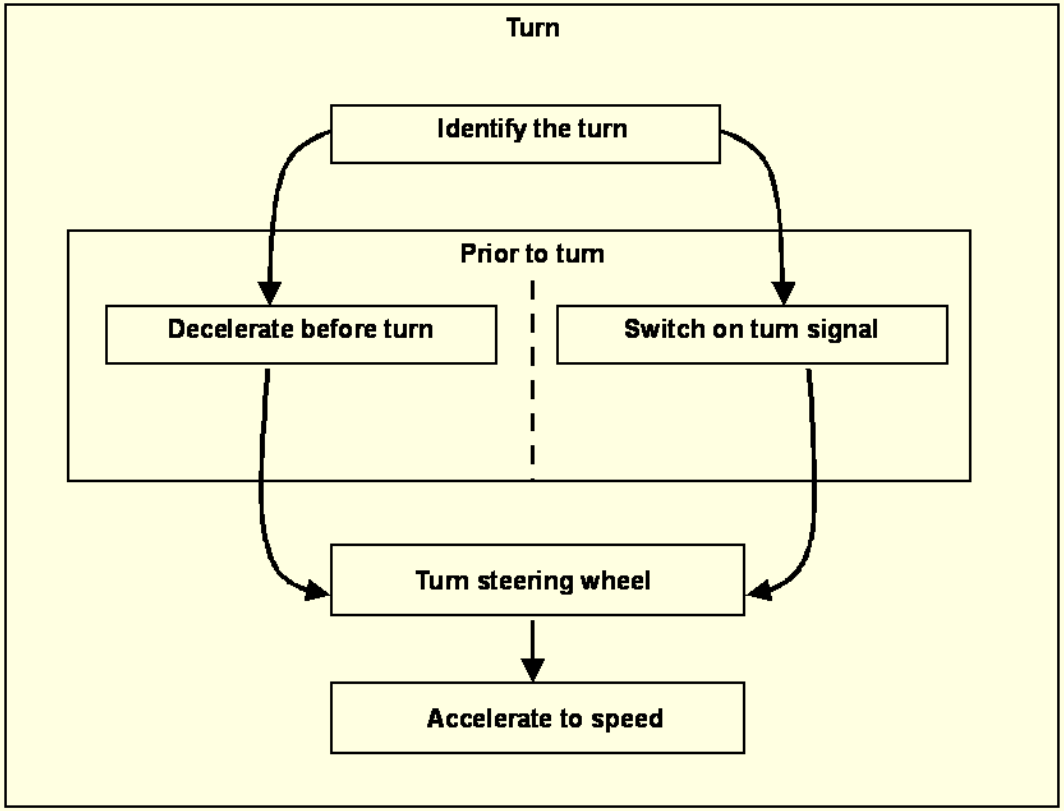


Total System Modeling: Use of Higraphs for System Modeling (Cont'd)

- The mathematical definition of that Higraph:
  - B = {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o}
  - E = {(i, h), (b, j), (l, c)}
    - e(l, c) = {(l, f), (l, e)}
  - ρ(H) = Σ ρ(b ε B)
    - ρ(a) = {b, c, h, j}
    - ρ(b) = {d, e}
    - ρ(c) = {e, f}
    - ρ(g) = {h, i}
    - ρ(j) = {k, l, m, n, o}
    - ρ(d) = ρ(e) = ρ(f) = ρ(h) = ρ(i) = ρ(k) = ρ(l) = ρ(m) = ρ(n) = ρ(o) = 0
  - Π(H) = Σ π_n(b ε B)
    - π_1(a) = {b, c, h}
    - π_2(a) = {j}
    - π_1(j) = {k, l, n}
    - π_2(j) = {n, o}
    - π_1(b) = π1(c) = π1(d) = π1(e) = π1(f) = π1(g) = π1(h) = π1(i) = π1(k) = π1(l) = π1(m) = π_1(n) = π1(o) = 0

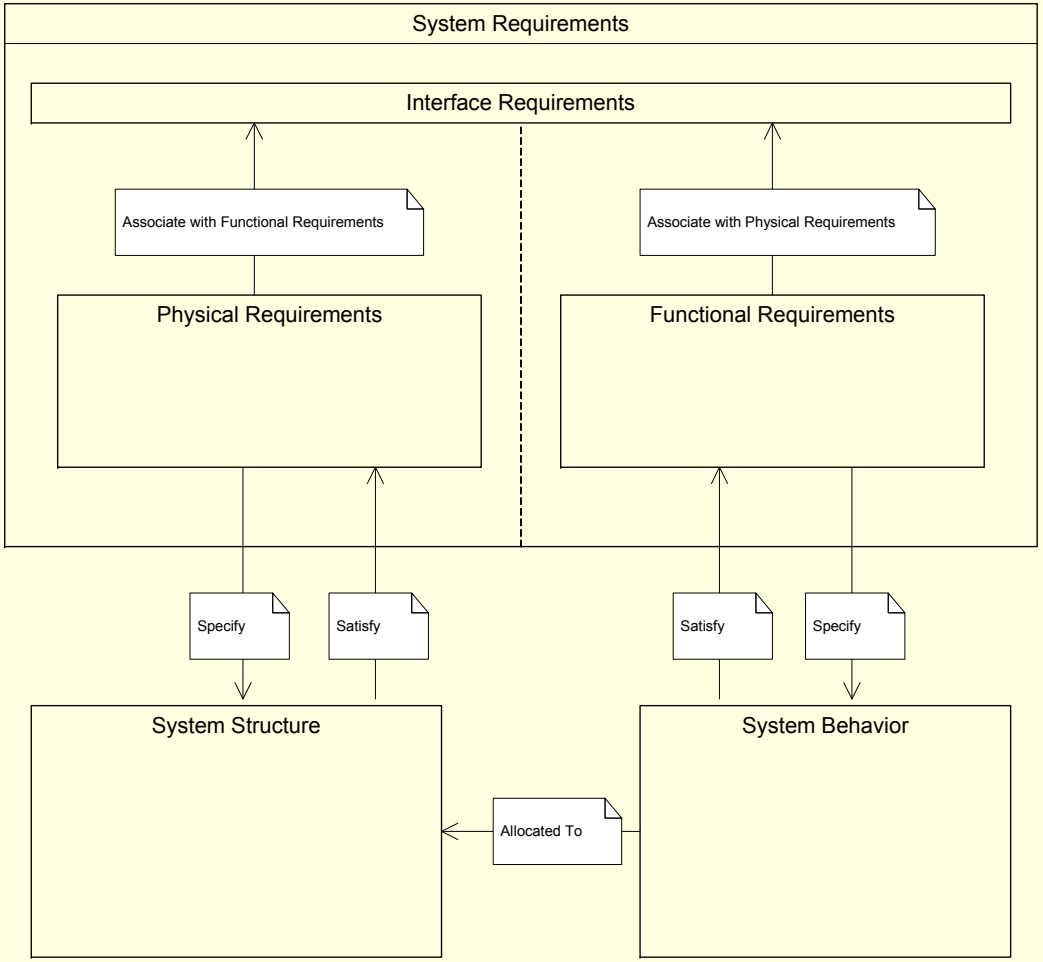
Total System Modeling: Use of Higraphs for System Modeling (Cont'd)

- In *[Fogarty-5]* and *[Fogarty-6],* we showed how Higraphs could theoretically be used as a system modeling language



Total System Modeling: Use of Higraphs for System Modeling (Cont'd)

- "Component" models could be linked to show a system model…



Total System Modeling: Use of Higraphs for System Modeling (Cont'd)

- …given rules for nodes…

- **B is the set of blobs [nodes]**
  - E is the set of edges
  - ρ is the hierarchy function
  - Π is the orthogonality (or partitioning) function
  - The quadruple (B, E, ρ, Π) defines a higraph, H

Area	Higraph Nodes (B)	Symbol
Requirements		
	Requirements Higraph	B_1
	Structure Requirements Higraph	B_{1-1}
	Requirement Number	B_{1-1-1}
	Requirement Area	B_{1-1-2}
	Requirement Type	B_{1-1-3}
	Requirement Text	B_{1-1-4}
	Requirement Owner	B_{1-1-5}
	Behavior Requirements Higraph	B_{1-2}
	Use Cases	B_{1-2-1}
	Actors	$B_{1-2-1-1}$
	System Behavior Requirements	$B_{1-2-1-2}$
Structure		
	Structure Higraph	B_2
	Components	B_{2-1}
	Attributes	B_{2-1-1}
	Functions	B_{2-1-2}
	Instances	B_{2-2}
Behavior		
	Behavior Higraph	B_3
	System States	B_{3-1}

Total System Modeling: Use of Higraphs for System Modeling (Cont'd)

- …edges…

- B is the set of blobs [nodes]
- **E is the set of edges**
- ρ is the hierarchy function
- Π is the orthogonality (or partitioning) function
- The quadruple (B, E, ρ, Π) defines a higraph, H

Area	Higraph Edges (E)	Symbol
Requirements		
	Allocation of a User to a Behavior	E_1
Structure		
	Inheritance	E_2
	Multiplicity Association	E_3
Behavior		
	State Transition	E_4
System Level		
	Assignment	E_5
	Assignment of a Structure Requirement to a Component Attribute	E_{5-1}
	Assignment of a Behavior requirement to a Use Case	E_{5-2}
	Assignment of a Use Case to a System State	E_{5-3}
	Assignment of a State Transition to a Component Function	E_{5-4}
	Satisfaction of a Domain Requirement by a System Requirement	E_6

Total System Modeling: Use of Higraphs for System Modeling (Cont'd)

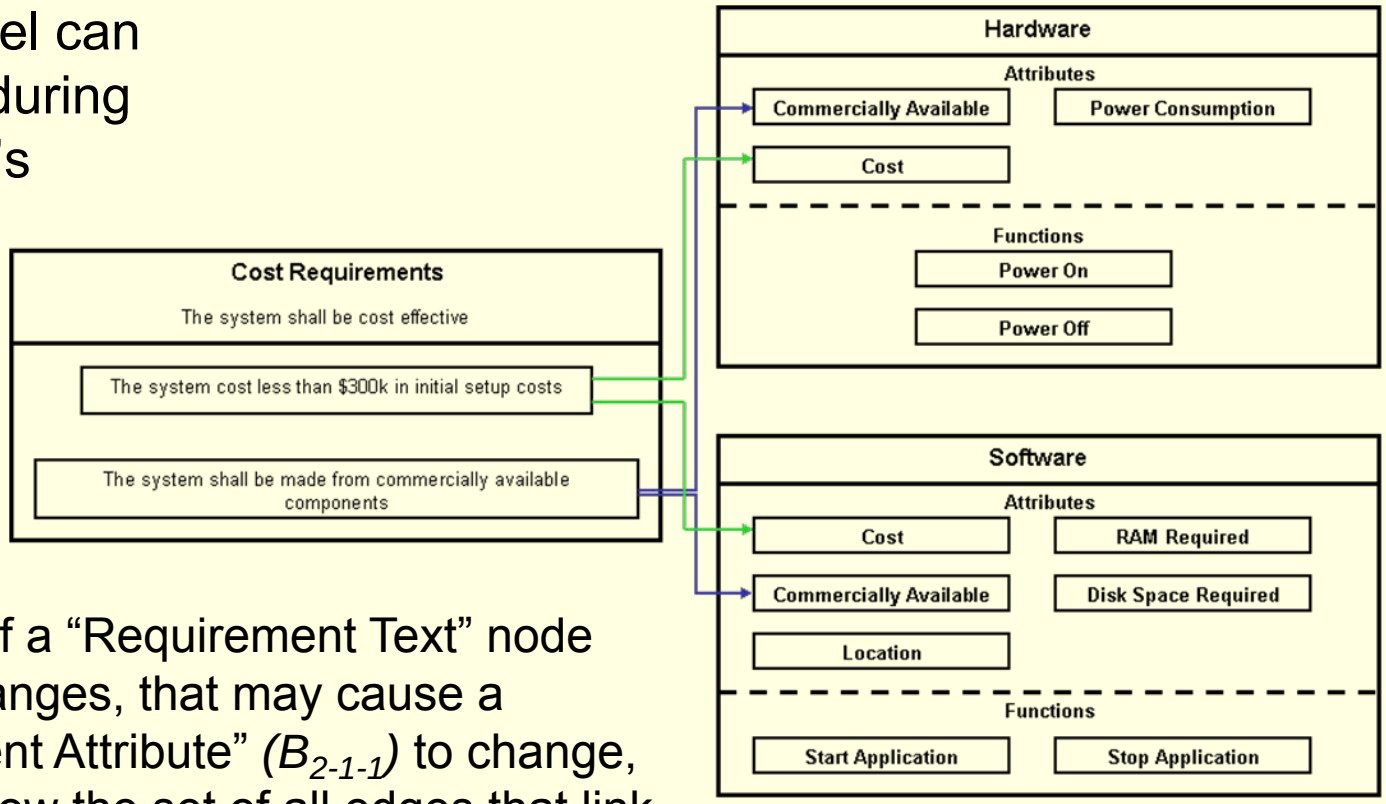
- …hierarchies, and orthogonality
  - B is the set of blobs [nodes]
  - E is the set of edges
  - **ρ is the hierarchy function**
  - **Π is the orthogonality (or partitioning) function**
  - The quadruple (B, E, ρ, Π) defines a higraph, H

Area	Higraph Hierarchy (ρ)	Symbol
Requirements		
	Requirements Hierarchy	ρ_1
	Use Case Hierarchy	ρ_2
Structure		
	Association of Attributes with a Component	ρ_3
	Association of Functions with a Component	ρ_4
Behavior		
	Behavior Hierarchy (States/Substates)	ρ_5

Area	Higraph Orthogonality (Π)	Symbol
Requirements		
	Requirements Domain	Π_1
Structure		
	Hardware Component or Software Component	Π_2
	Component Attribute or Component Function	Π_3
Behavior		
	Allowed Concurrent Behavior	Π_4

Total System Modeling: Use of Higraphs for System Modeling (Cont'd)

- This model can be used during a system's lifecycle

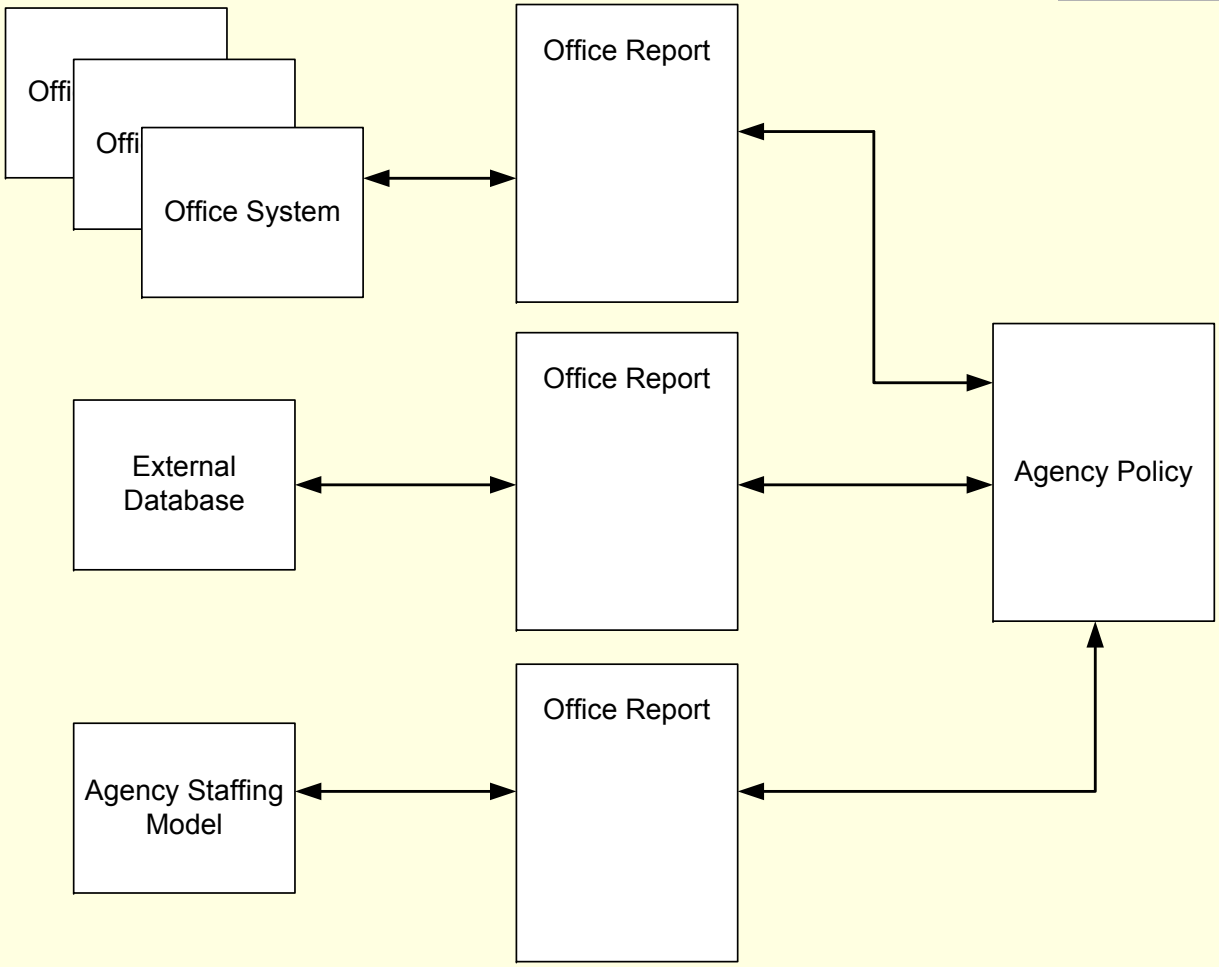


- We know if a "Requirement Text" node ($B_{1\text{-}1\text{-}4}$) changes, that may cause a "Component Attribute" ($B_{2\text{-}1\text{-}1}$) to change, and we know the set of all edges that link requirements and attributes ($E_{5\text{-}1}$)

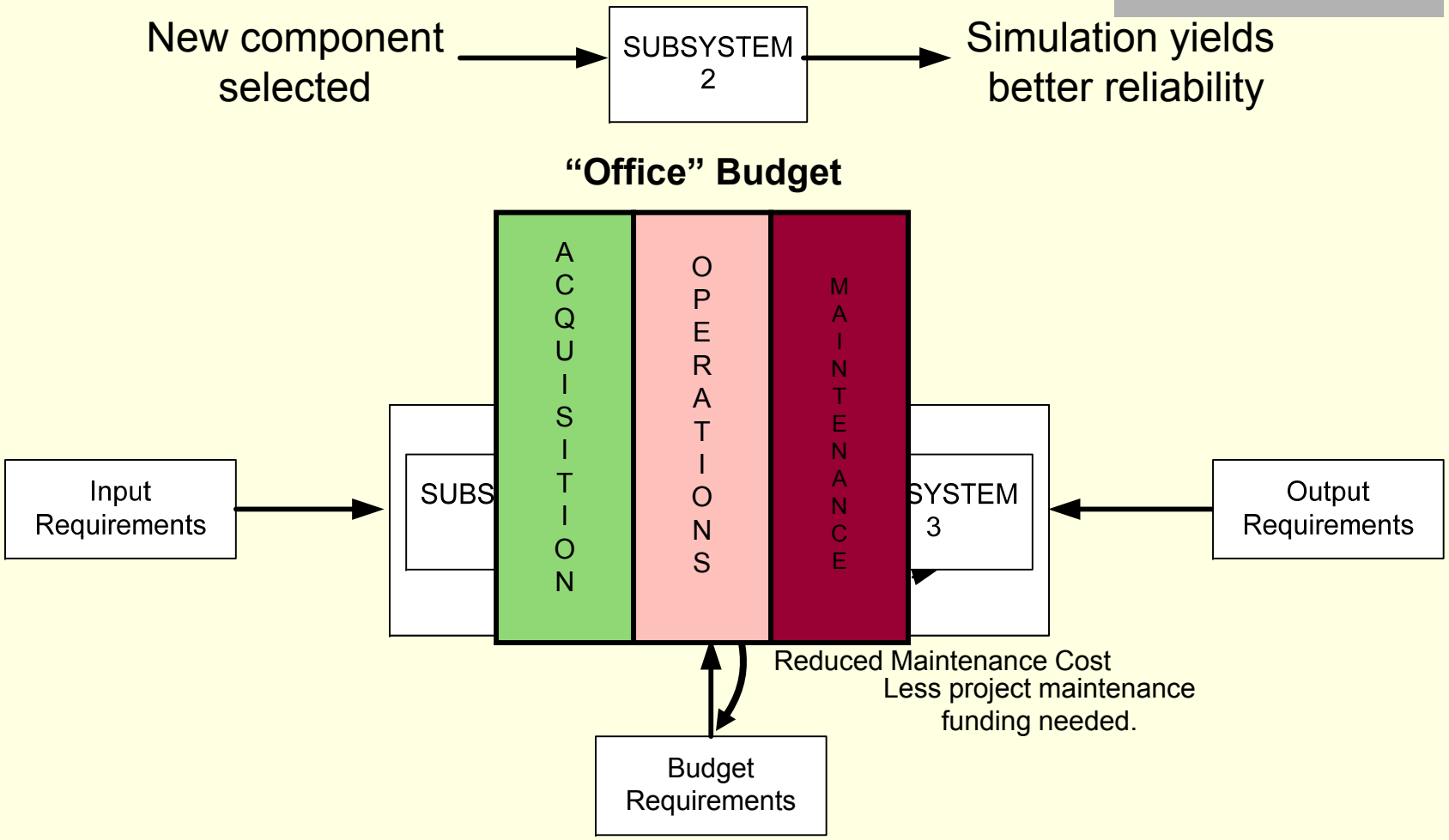
Total System Modeling: Use of Higraphs for System Modeling (Cont'd)

- But an even better use for Higraphs might be as a modeling language used to define and connect Systems of Systems
  - Each system utilizes its own modeling method
  - Higraph nodes become those models, or components/views from those models
    - Edges are used to show relationships between the models
    - Nodes can be arranged in a hierarchy (implying inheritance and traceability)
    - Orthogonality can be used to show time/schedule dependencies (among others)

Total System Modeling: Use of Higraphs for System Modeling (Cont'd)



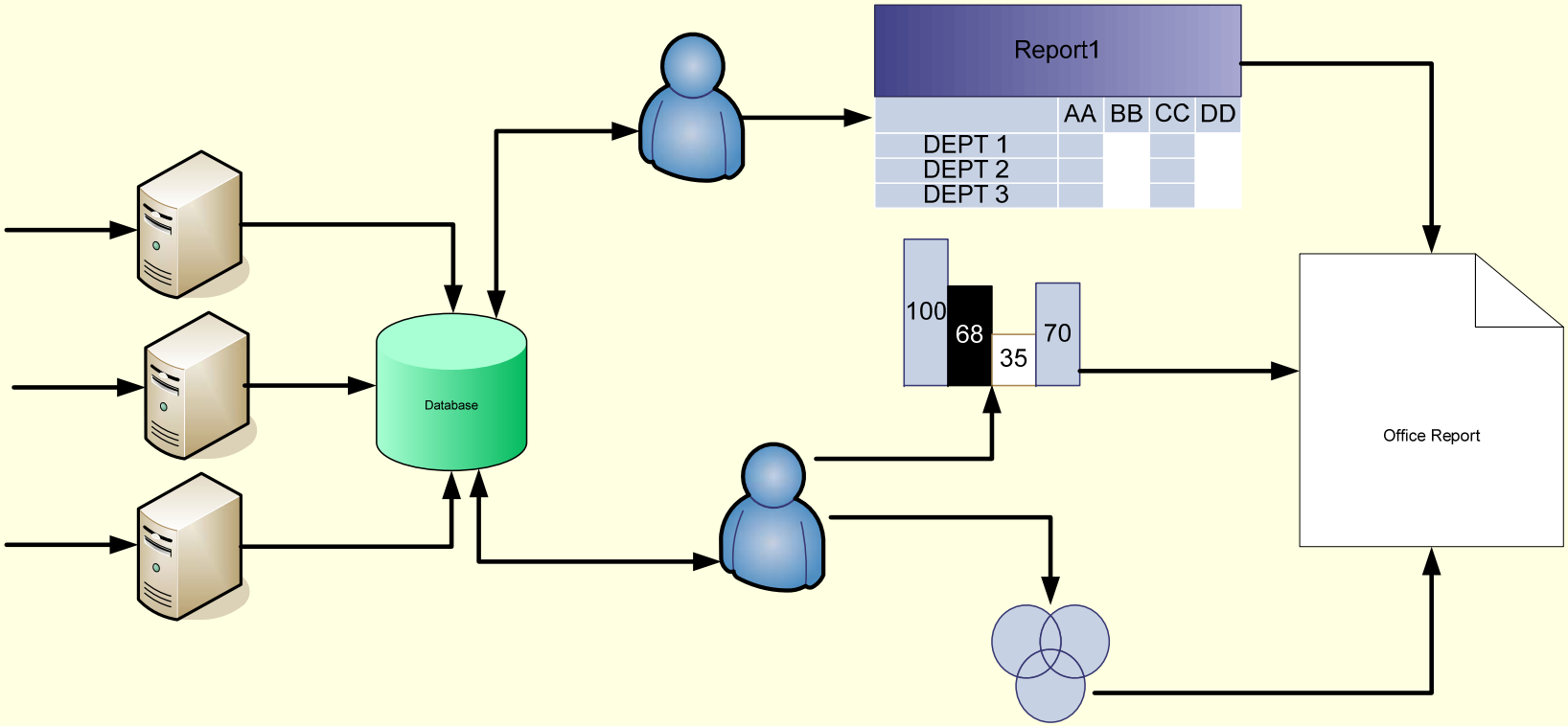
Total System Modeling: Use of Higraphs for System Modeling (Cont'd)



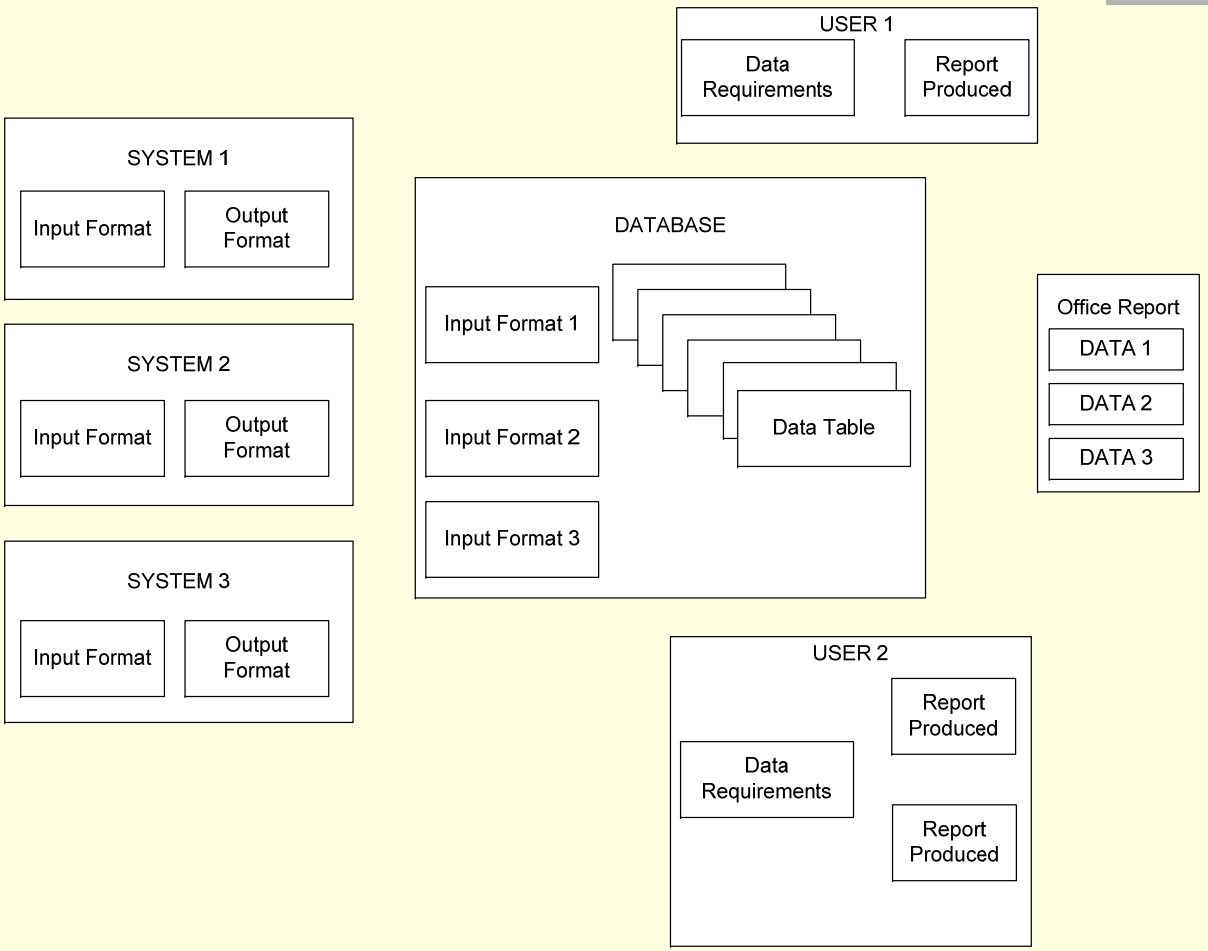
Total System Modeling: Practical Implementation of a Higraph-Based Model

- Each “project” documents all external (input and output) interfaces and dependencies related to that project in a format that could be imported into a higraph-based tool. Examples include:
  - Technical: HW Platform, SW Interface, Network Interface
  - Schedule: Delivery Dates, Test Cycles
  - Budget: Labor $$, Material $$, O&M $$, Funding Source
- Verified/validated at an “office” level
- These dependencies get “rolled-up” and combined with models from other projects/programs/offices

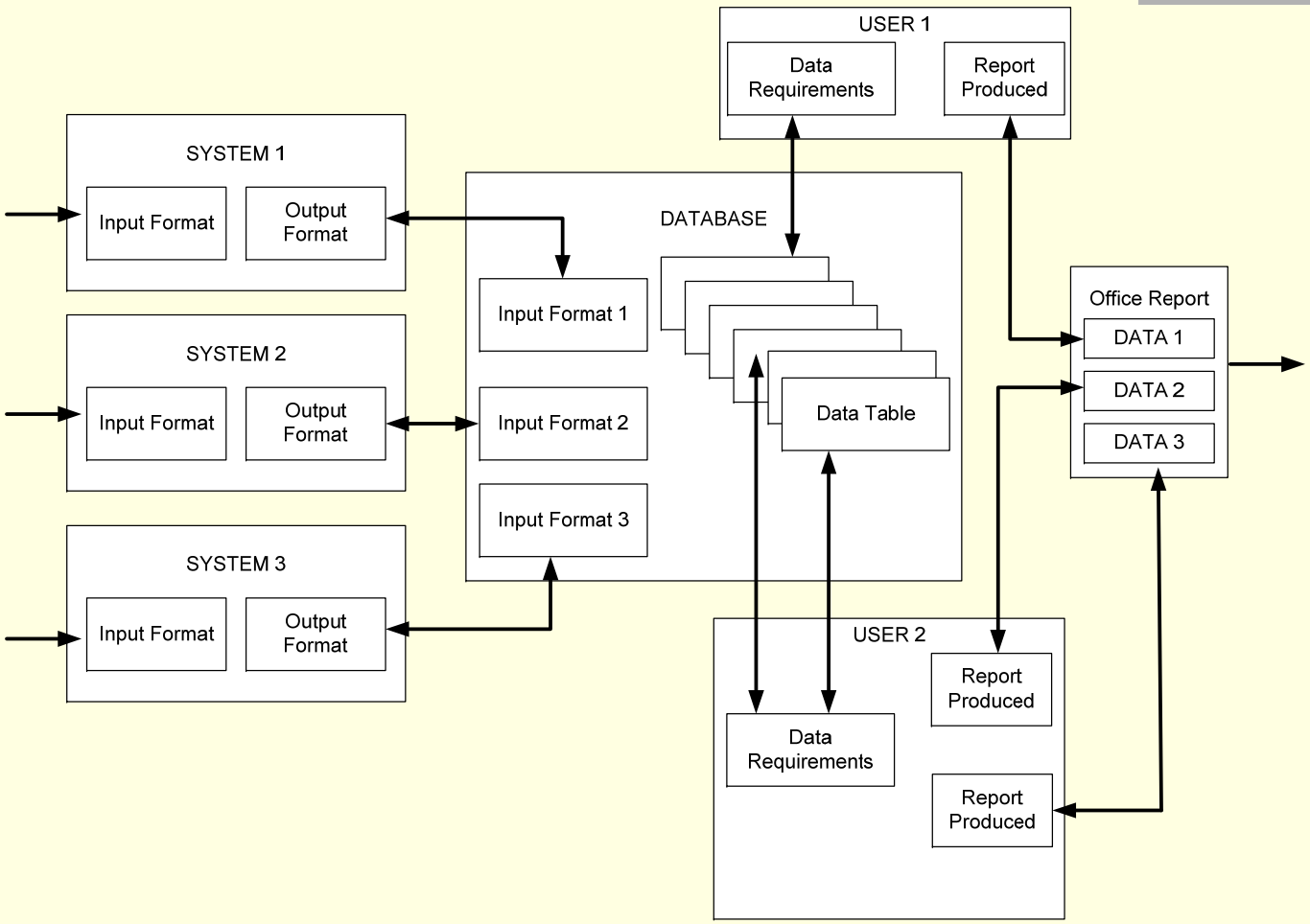
Total System Modeling: Practical Implementation of a Higraph-Based Model (Cont'd)



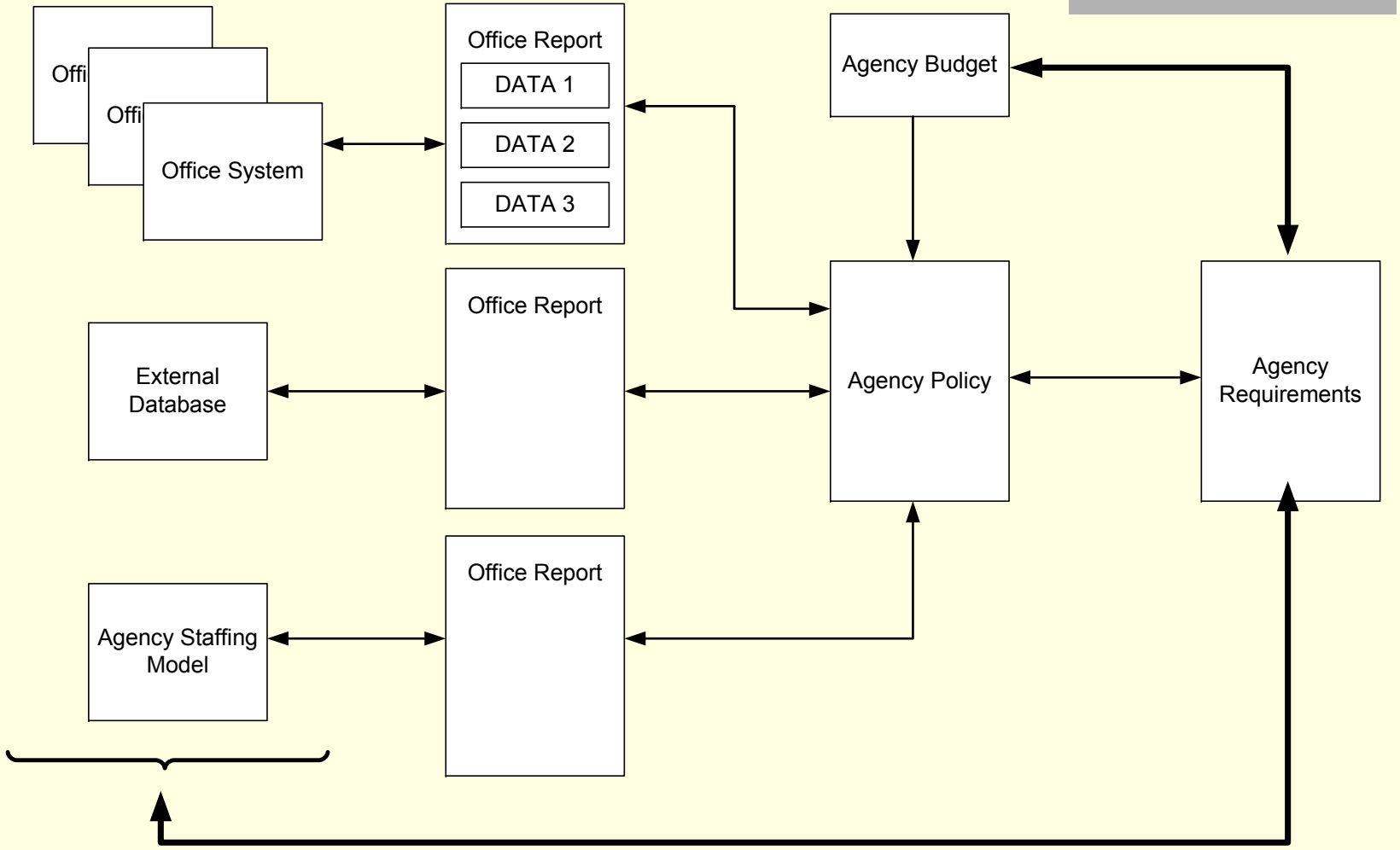
Total System Modeling: Practical Implementation of a Higraph-Based Model (Cont'd)



Total System Modeling: Practical Implementation of a Higraph-Based Model (Cont'd)



Total System Modeling: Practical Implementation of a Higraph-Based Model (Cont'd)



Total System Modeling: Benefits and Conclusions

- Benefits include:
  - Independent, mathematically based modeling language to connect existing models
  - Ability to customize rules for meanings behind nodes, edges, hierarchies, and orthogonalities
  - Ability to actively use the model for design validation, complete traceability, error checking, etc.

Total System Modeling: Benefits and Conclusions (Cont'd)

- Higraphs pose a simple, yet powerful, way to connect system models
  - Keeping a mathematical foundation allows for formality during tool implementation and inference based on patterns
  - Over time we can arrive at a true, Total System Model

Total System Modeling: Questions and Comments

Questions or Comments?

Total System Modeling: References

1. Alberts, D.S., Garstka, J.J., Stein, F.P., "Network Centric Warfare: Developing and Leveraging Information Superiority," CCRP Publ., 2nd Edition (Revised). Aug 1999, Second Print Feb 2000, p 2.
2. Department of Defense, "Acquisition Modeling and Simulation Master Plan," 17 April, 2006.
3. Department of Defense, "DoD Architecture Framework Version 1.5 Volume I: Definitions and Guidelines," 23 April, 2007, p ES-1.
4. Department of Defense, "Systems Engineering Guide for Systems of Systems: Version 1.0," August, 2008, p iii.
5. Fogarty K., "System Modeling and Traceability Applications of the Higraph Formalism," MS Thesis, Institute for Systems Research, University of Maryland, College Park, MD 20742. May 2006.
6. Fogarty K. and Austin M.A., "Systems Modeling and Traceability Applications of the Higraph Formalism," Systems Engineering: The Journal of the International Council on Systems Engineering, May 2008 (In Press)
7. Harel David. "Statecharts: A Visual Formalism for Complex Systems." Science of. Computer. Programming, 1987. Vol. 8, pp. 231-274.
8. Jain, P., and Dickerson, C., "Family-of-Systems Architecture Analysis Technologies," INCOSE Proceedings, July 2005.
9. Kotov, V., "Systems of Systems as Communicating Structures," Hewlett Packard Computer Systems Laboratory Paper HPL-97-124, (1997), pp. 1-15.
10. Letelier, Patricio. "A Framework for Requirements Traceability in UML Projects." In Proc. of 1st International Workshop on Traceability in Emerging Forms of Software Engineering. In conjunction with the 17th IEEE
11. Manthorpe, W.H., "The Emerging Joint System of Systems: A Systems Engineering Challenge and Opportunity for APL," John Hopkins APL Technical Digest, Vol. 17, No. 3 (1996), pp. 305-310.
12. The Object Management Group, Inc. "Unified Modeling Language Superstructure, Version 2.1.2." November 2007, p 1.
13. Richards, M., Shah, N., Hastings, D., and Rhodes, D., "Architecture Frameworks in System Design: Motivation, Theory, and Implementation," *17th* INCOSE Symposium, San Diego, CA, June, p 6.
14. SysML Partners. "Systems Modeling Language Specification, Version 1.0," September, 2007, p 1.