

An Executable and Integrative Approach to Whole-System Modeling: Applying OpEMCSS to Simulation-Based and Model- Based Systems Engineering for Holonic Systems Engineering

Jose S. Garcia, Jr., M.S.E.E.

Systems Engineer – The Boeing Company
INCOSE-LA's Industrial Relations Chair
February 7, 2009

Abstract

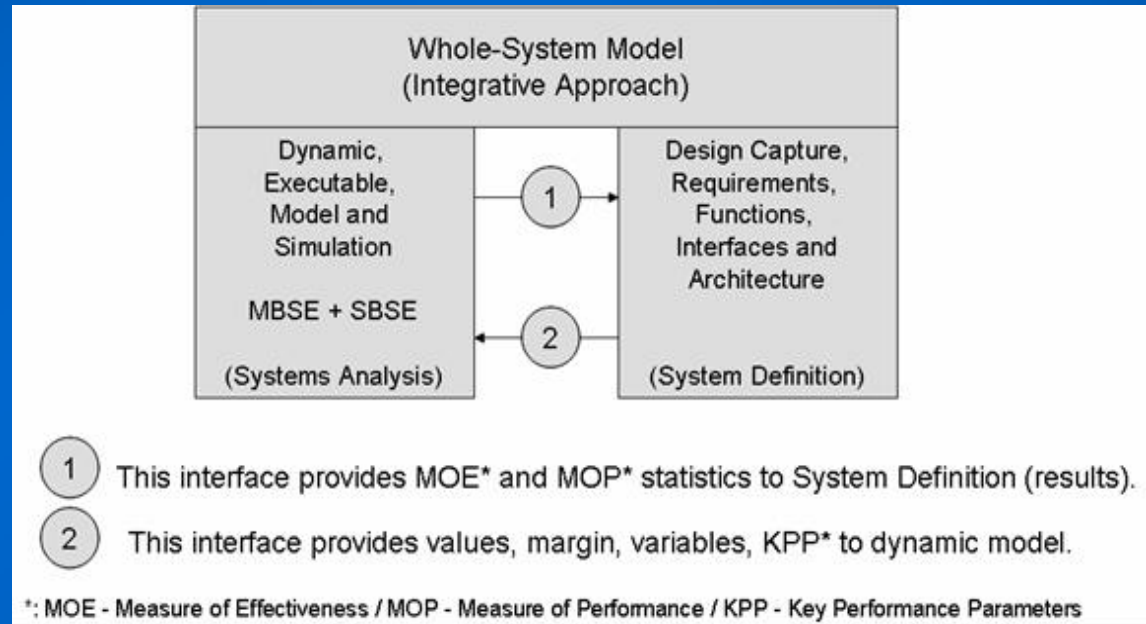
- A traditional systems engineering process emphasizes requirements, traceability, functional decomposition, architecture, optimization, and static modeling. The lack of an executable modeling paradigm in this process constricts the operational evaluation of requirements for agent-based systems.
- A holonic systems engineering process identifies requirements in terms of processes and interactions. It facilitates an executable and integrative approach to Whole Systems Modeling (WSM), by leveraging complex adaptive system techniques and simulations for pragmatic foresight into system behavior, via the application of Operational Evaluation Modeling for Context Sensitive Systems (OpEMCSS).
- The key enabler to WSM is applying OpEMCSS to simulation-based and model-based systems engineering.

Assumptions

- This study focuses on systems and System of Systems that contain agents. Not all systems contain agents. However, most complex system problems involve multi-agent (two or more) entities, systems, and subsystems.
- There is no single tool or methodology that enables Whole-System Modeling and design, that is both executable and integrative in nature.
- Model-Based Systems Engineering includes Simulation-Based tools.
- Current modeling tools do not provide multiple instantiations of processes or agents, as found in current System of Systems problems and situations.
- Hypothesis:
 - An executable and integrative approach to Whole-System Modeling can be accomplished by applying OpEMCSS to all aspects of model-driven design that includes both Model-Based Systems Engineering (MBSE) and Simulation-Based Systems Engineering (SBSE), herein collectively referred to as MBSE

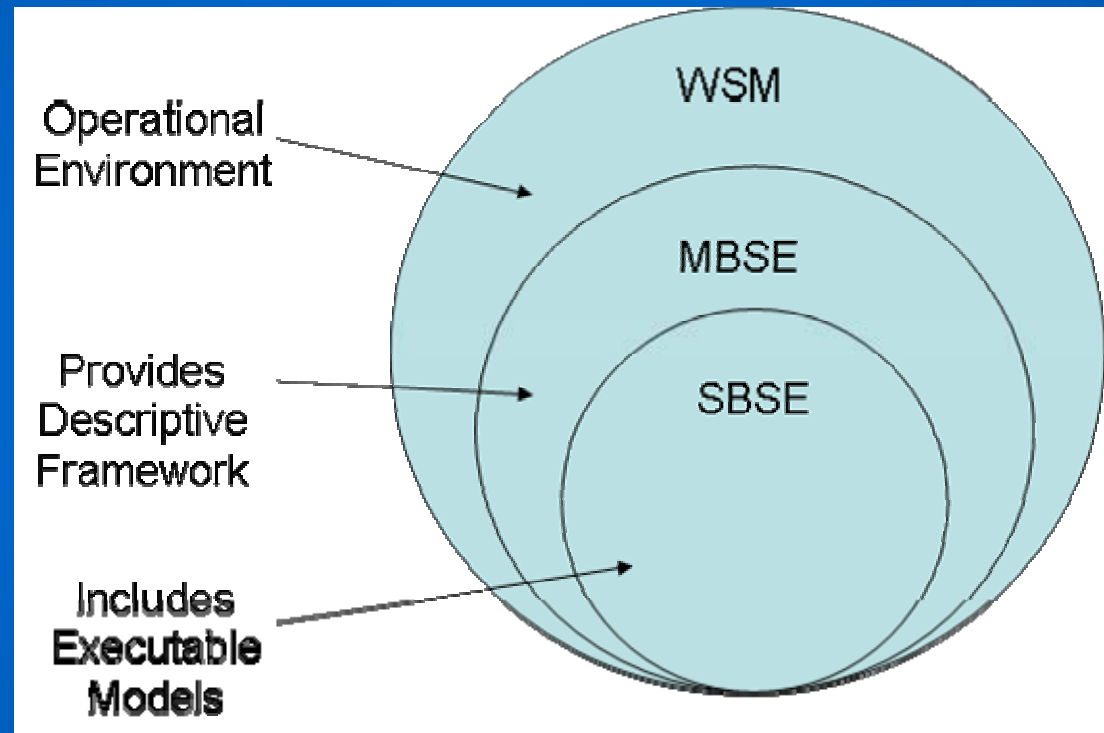
Integrative Modeling Paradigm

- The intent of this research is to also provide an understanding of the systems engineering process within the context of whole-systems modeling [5].
- This study, although focused on systems engineering, encompasses multi-disciplinary applications.
- The research and conclusions provided herein will help set the foundation for an integrative modeling paradigm,



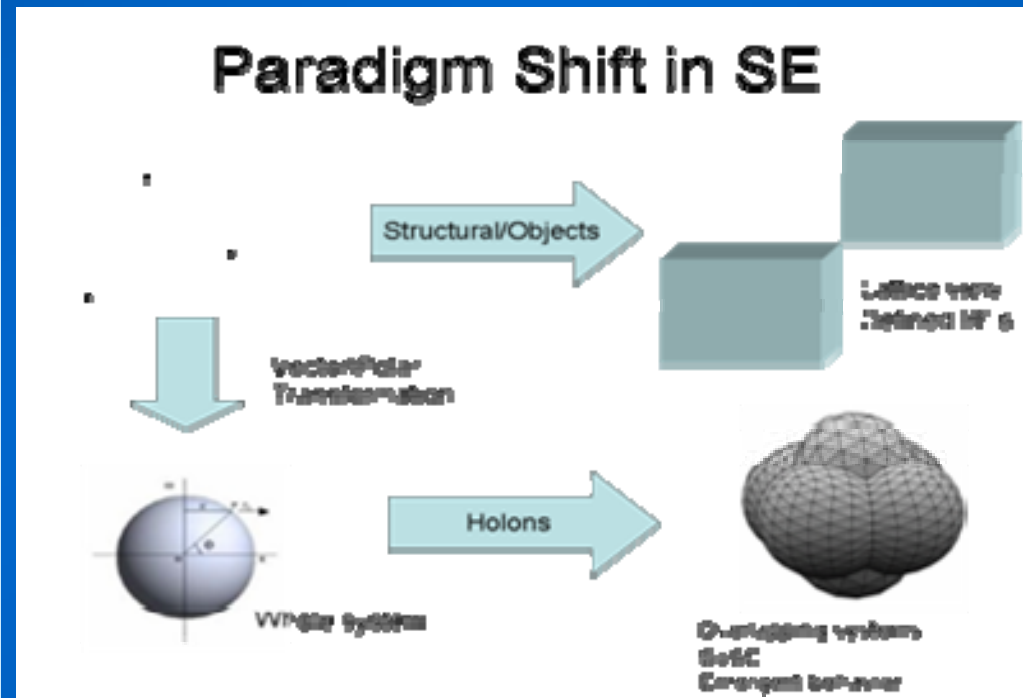
Motivation...

- Thus there is an identifiable need for an executable and integrative approach to Model-Based Systems Engineering (MBSE) for Whole-System Modeling (WSM).
- Operational Evaluation Modeling for Context-Sensitive Systems (OpEMCSS), as developed by Dr. John R. Clymer, is a modeling (MBSE) and simulation [Simulation-Based Systems Engineering (SBSE)] tool that allows an explicit understanding of complex system interactions and complex adaptive system behavior amongst system components, subsystems, and processes [4].



Paradigm Shift

- Indeed, a paradigm shift is occurring within systems engineering, where the level of complexity of System of Systems projects can only accurately and precisely be modeled by an executable modeling construct.
- This further requires a paradigm shift in the treatment of subsystems and agents as holons.

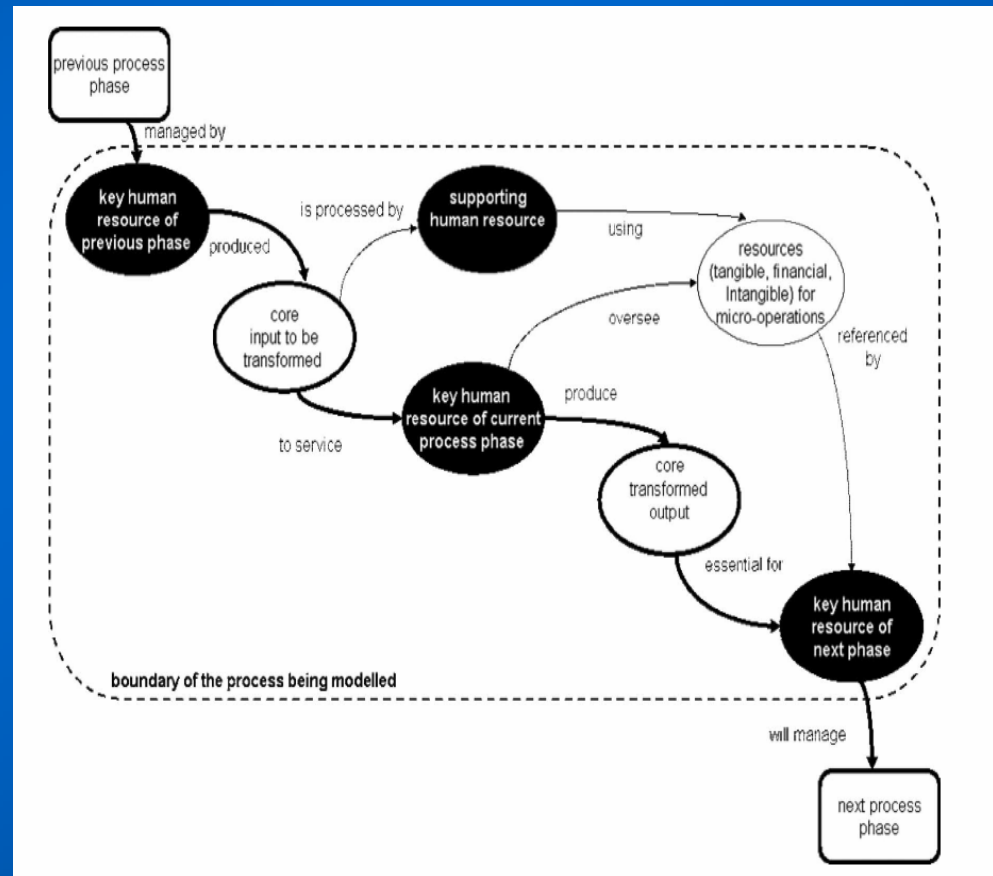


Holonic Systems Engineering

- A new modeling and requirement construct is needed to better understand agents and their multiple process instantiations that are part of an executable and integrative WSM application such as OpEMCSS.
- This conceptual application is the use of holons. A holon is a term coined by Arthur Koestler in his 1967 book *The Ghost in the Machine* [20].
- In this study, the term holon describes agents, subsystems, and entities that are both autonomous and cooperative.
- They are characterized by multiple levels of granularity, in a self-similar structure [3]. A holon is both a whole and part of another whole, simultaneously.
- Holons are system agents that are autonomous and cooperate, and lend themselves well to understanding the interdependent and emergent properties of agent-based systems
- A holonic requirement, as defined in this study, encapsulates a requirement within a process or agent framework. It is considered one holon, which is both part and whole to other holons. Thus, holons (a set of holons is termed holarchy) facilitate a non-reductionist and aggregational set of requirements

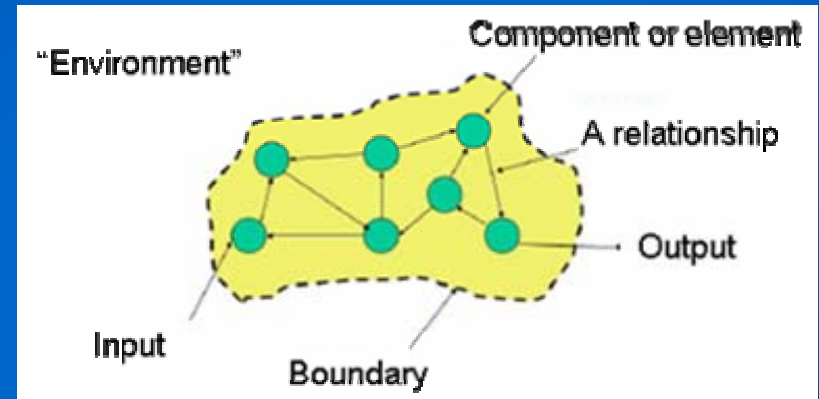
Defining Holonic Systems Engineering

- Clegg [22] describes a template for a “Process-Oriented Holonic” (PrOH) model for business processes



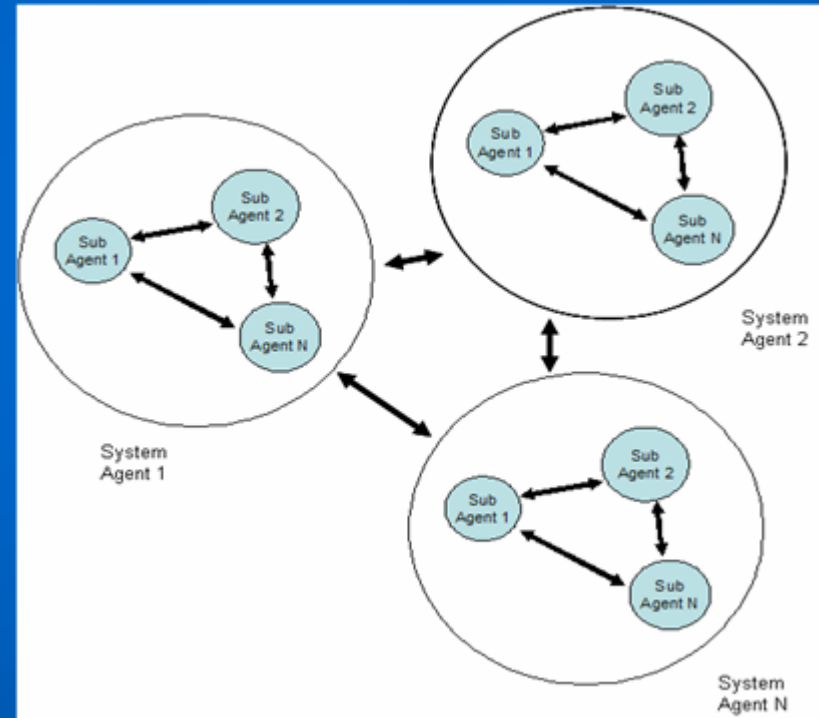
Systems Engineering Process

- A formal definition of the systems engineering process must begin with a thorough understanding of what constitutes a “system” and what is meant by a “process”.
- According to Buede [8], a system is a “set of components acting together to achieve a set of common objectives via ... a set of tasks.”
- A process is a series of operations or steps used to instantiate a desired result, such as in the engineering design of a system.
- Forsberg and Mooz Systems Engineering Vee model of the systems engineering process, as highlighted in Buede



Shift to System of Systems Engineering

- The integration of multiple complex systems (System of Systems) has given rise to a new challenging set of problems facing systems engineers and practitioners [16].
- A System of Systems is a system abstracted and constructed of an aggregate of individual systems, which interoperate and interface with each other, and are characterized by emergent properties
- Network-centric operations enable System of Systems operations by facilitating message passing.



Shift to System of Systems Engineering

- The aforementioned paradigm shift to the System of Systems understanding of system problems, and their engineering via SoSE, is an evolutionary value-adding process that presents a variety of challenges that go beyond complexity, such as:
 - (1) ambiguity, (2) human social dynamics, (3) sustainability, and (4) methodology [Ring,18].
- These statements substantiate the need for a paradigm shift in applying holons to the systems engineering process, thus enabling the validation of the hypothesis.

SoS: Emergent Behavior

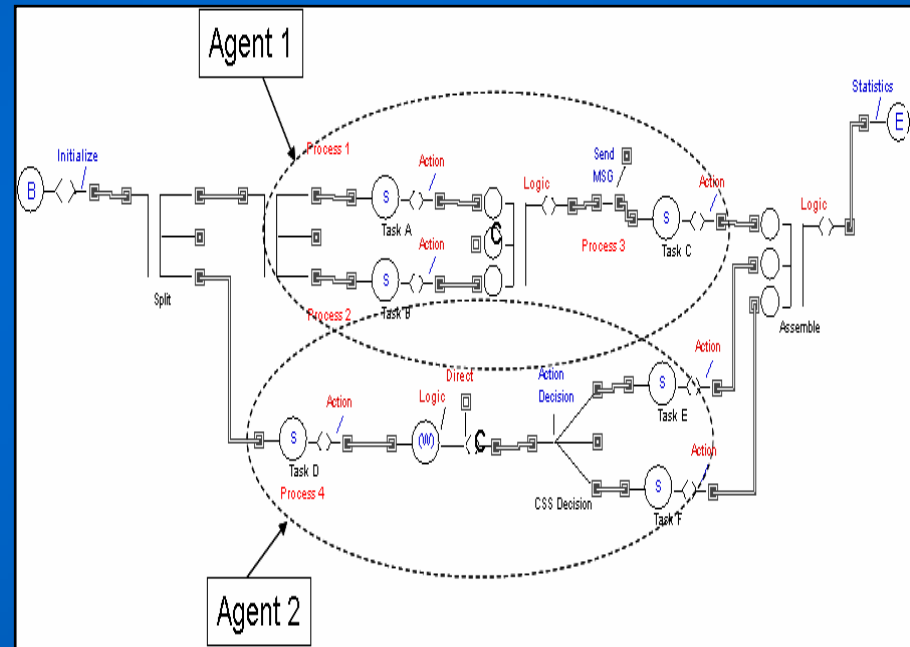
- System of Systems are characterized by emergent behavior. Emergent behaviors are properties of a system that emerge at a higher level of complexity that do not occur at a lower level [6]. Understanding emergent behavior is critical to properly developing executable models. There are four principles of emergence as discussed in Hsu [19]:
 1. The “condition of emergence” only occurs for an aggregate of subsystems or entities that are part of the System of Systems.
 2. “Emergent behavior is inversely proportional to the degree of bondage between systems.” The greater the degree of coupling amongst subsystems, the less likelihood emergent behavior will emerge. Loosely coupled systems exhibit a greater degree of emergent behavior.
 3. “Emergent behavior is non-linear.”
 4. “Emergent behavior is self-organized.”

Complex Adaptive Systems (CAS)

- A system that has emergent behavior due to component interactions
- Complex Adaptive Systems (CAS) have emergent behavior due to Context Sensitive System (CSS) interactions.
- Three things required in solving CAS problems:
 - Methodology – Context-Sensitive Systems (CSS) approach to systems engineering
 - Model – Operational Evaluation Modeling (OpEM) graphical language
 - Tool - Operational Evaluation Modeling (OpEM) for Context-Sensitive Systems (CSS) or OpEMCSS

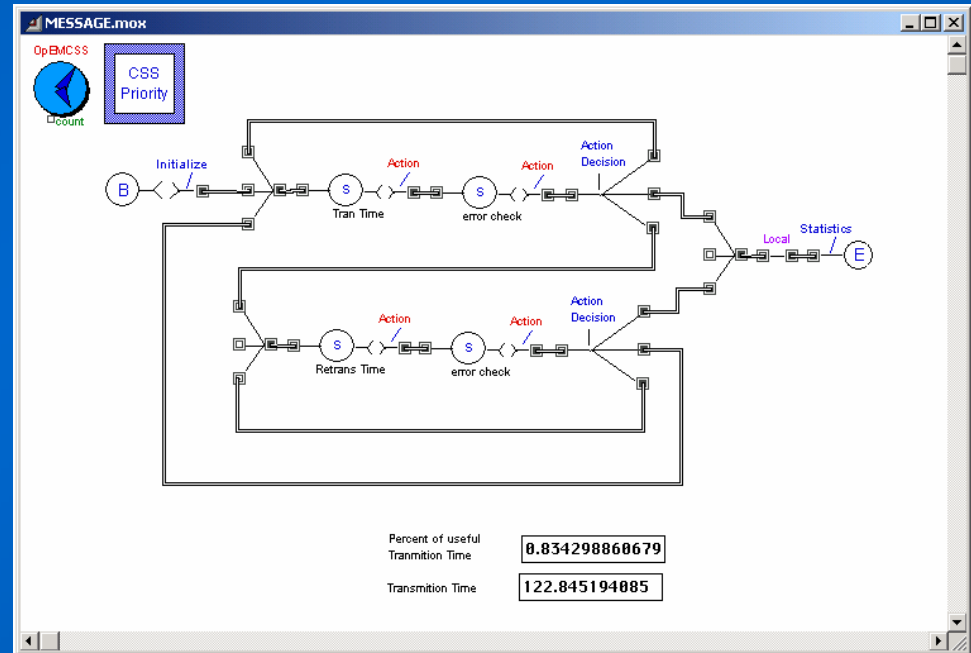
Clymer Definition of Agent-based system architecture

- An agent-based system architecture is a network of intelligent agents that share facts with other agents and adapt their behavior in response to these shared facts; indeed intelligent agents apply knowledge in the form of rules to transform input to output facts and to make decisions to adapt their behavior.
- Through such collaboration, an agent-based system achieves overall system goals, and it often exhibits emergent behavioral patterns for the overall system [4].
- It is a subsystem or component of a larger system that interacts with other agents and or subsystems within the system, which results in emergent behavior. Agents can be hardware-based, software-based, process-based, people-based, or any multitude of entities that exhibit complex adaptive behavior.



Implementation of MBSE: OpEMCSS

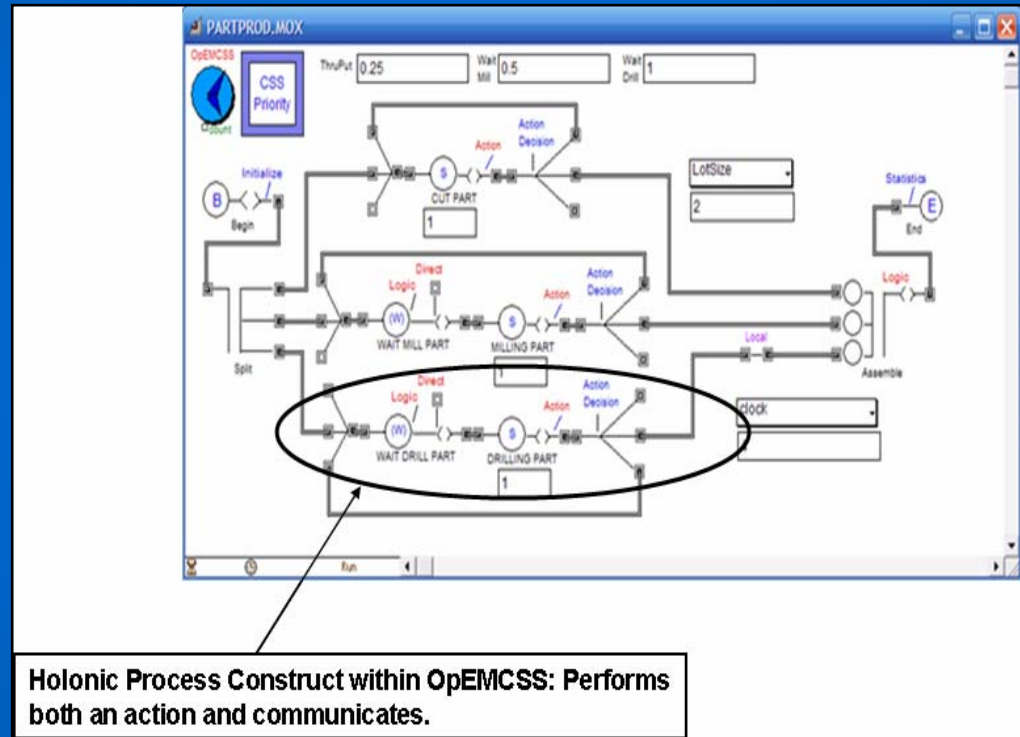
- **Operational Evaluation Modeling (OpEM) Language** : (Extend s/w by ImagineThat, Inc.)
 - Parallel processing language concepts
 - Modeling concurrent process interactions
 - Multiple instantiation of parallel threads.
- Visually shows important concepts such as process state-transitions, synchronization, resource contention, communication and adaptation.
- Allows variable instantiations of process threads using same modeling diagrams
- Provides data structures to represent process instances, state variables, and shared global memory
- An executable and integrative approach to Model-Based Systems Engineering (MBSE) for Whole-System Modeling (WSM) via OpEMCSS, allows both an expansionist and reductionist approach



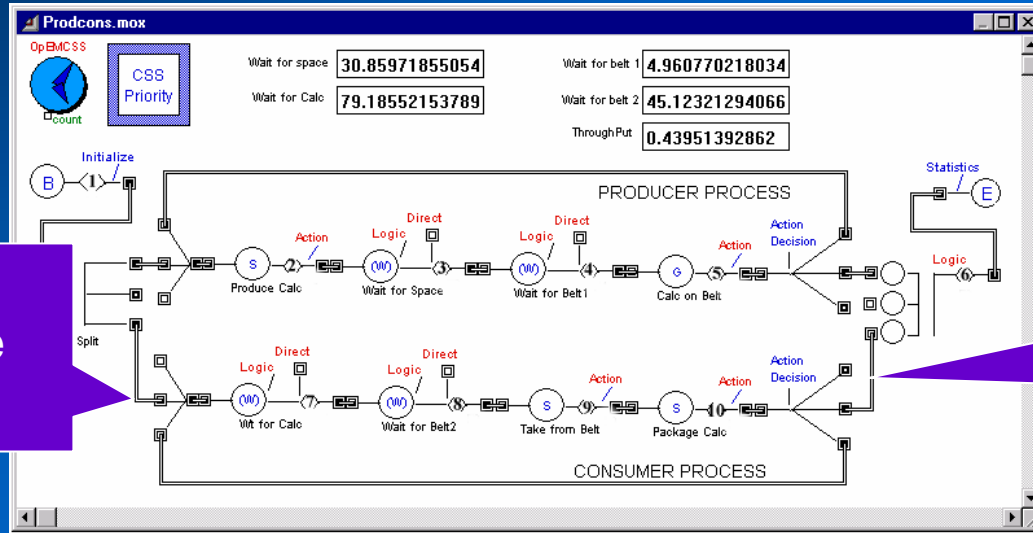
- Dynamic and Executable modeling

Holons in OpEMCSS

- A holon can be easily adapted in OpEMCSS for executable Whole-System Modeling of a multitude of system implementations, such as businesses, enterprises and manufacturing operations [24].
- OpEMCSS allows for the creation of special auxiliary blocks that capture transformation and action between various processes and subsystems.



Action Blocks

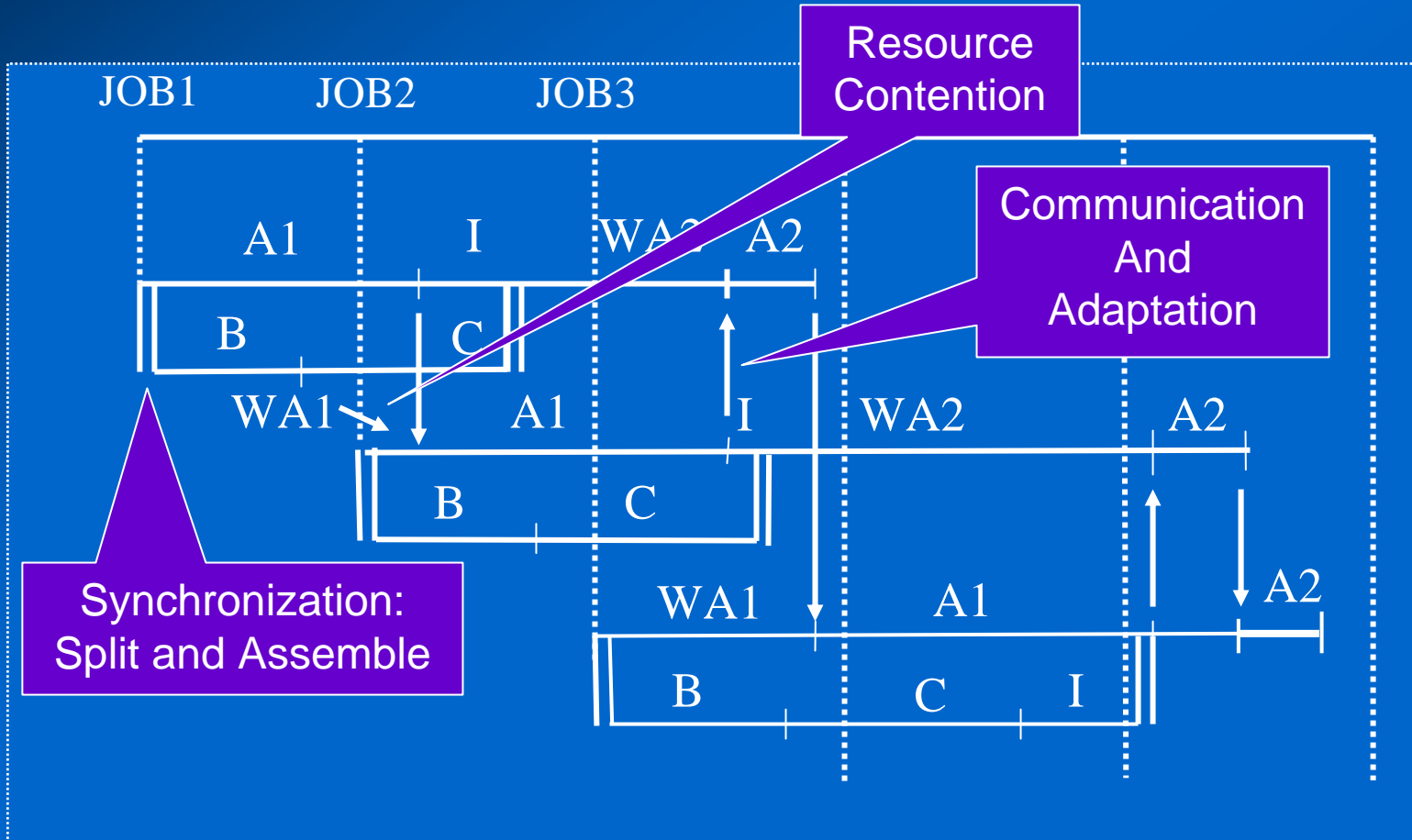


Event Occurrence

Alternate Action

- Event Occurrence lets us merge multiple inputs together
- Alternate Action lets us select one of three paths

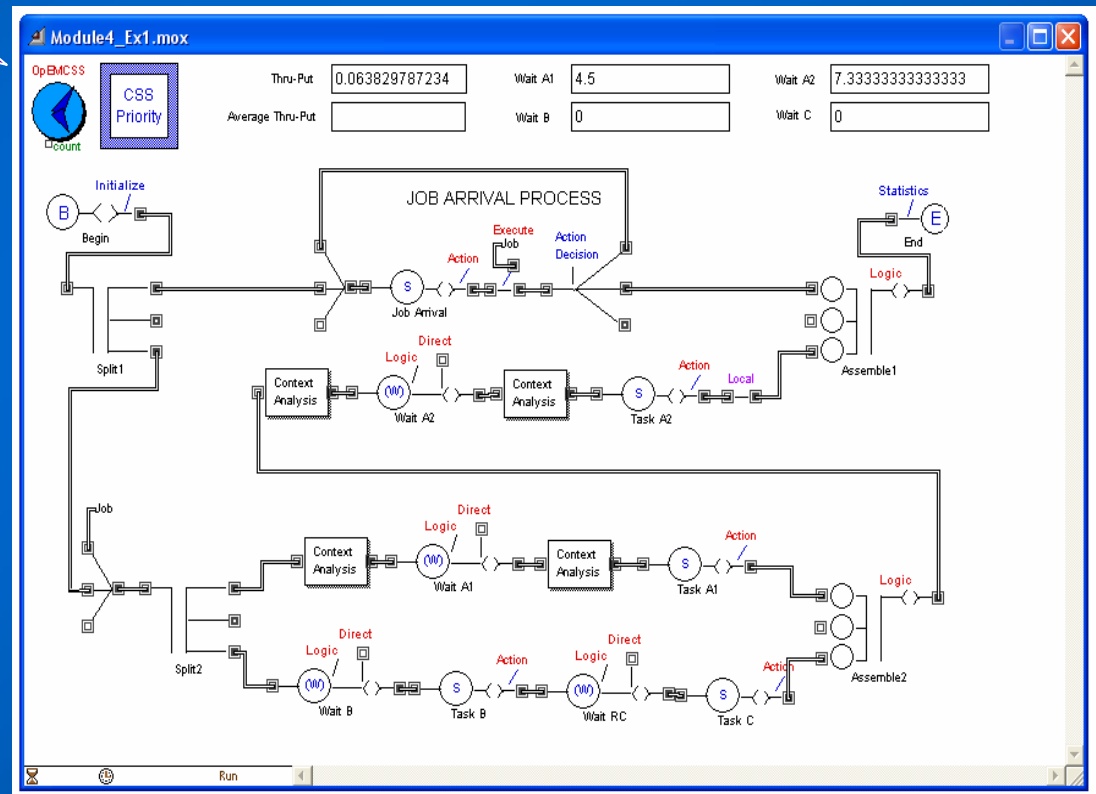
Timeline: Complex Interactions



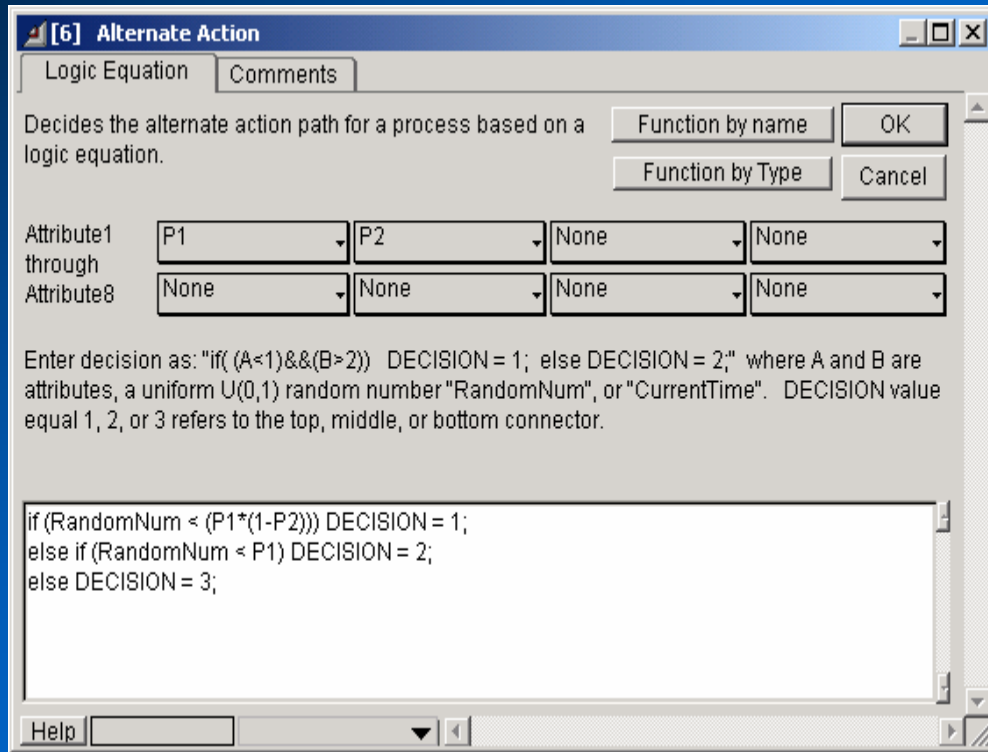
OpEMCSS: Sensitivity Analysis

Model
Scoreboard
Contains
MOEs and
MOPs

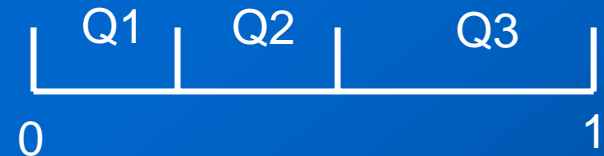
- Measures of Effectiveness (MOE) and Measures of Performance (MOP) are the quantities by which we determine if a system meets its requirements
- Key Performance Parameters (KPP) are the quantifiable factors we identify as best able to influence achieving the MOEs & MOPs



Probabilistic Decision



How to simulate a discrete probability distribution



1. Draw a random number R
2. If $R < Q1$ Do action 1 else we eliminate Q1 region
3. If $R < (Q1 + Q2)$ Do action 2 else we eliminate Q1 and Q2 region and Do action 3

- First decision is IF ($Rnd < (P1 * (1 - P2))$) DECISION = 1;
- Second decision is IF ($Rnd < (P1 * (1 - P2)) + P1 * P2$) DECISION = 2; which is same as IF ($Rnd < P1$) DECISION = 2;

Whole-Systems Modeling

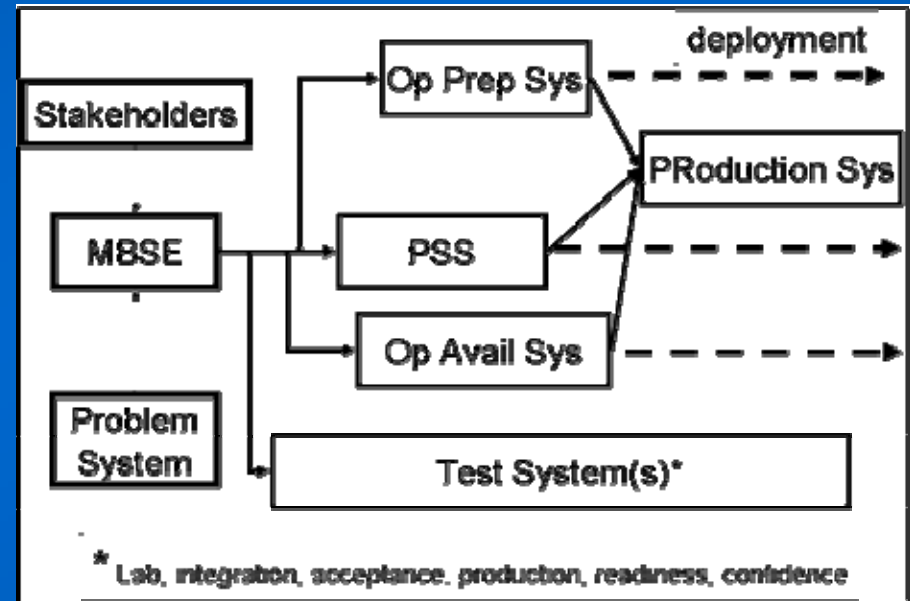
- The paradigm shift of modeling systems, from simple flow diagrams to whole-systems modeling is evolving out of the necessity to comprehend complex systems.
- A complex system is a system composed of a subset of interconnected parts that exhibit collective properties and behaviors as a whole.
- Whole-System Modeling includes both model-based and simulation-based system engineering as an aid in high-level abstraction of a system design.
- Model-Based Systems Engineering is the practice and discipline within the field of systems engineering that models system interactions and interoperability in order to better engineer or develop an intended system design.
- Simulation is a computer-based or mathematical-based analysis of a complex system which measures the MOE's of a system concept constrained by a model scenario of the operational environment [14].
- Simulation-Based Systems Engineering is the process and discipline of using simulation to evaluate candidate system designs within an operational scenario.

“There is no whole that isn’t also simultaneously a part of some other whole, indefinitely, unendingly. Time goes on, and today’s wholes are tomorrow’s parts ... ”

–Ken Wilber

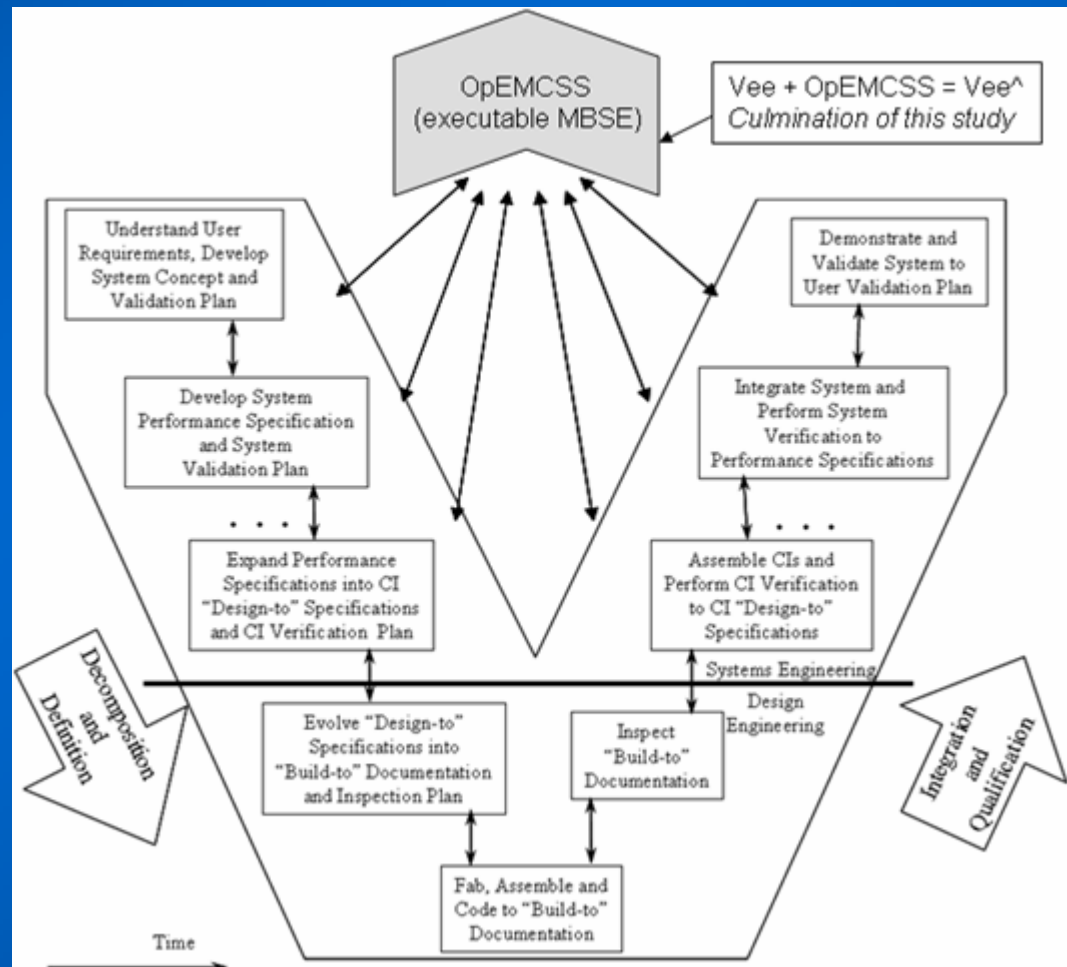
Taking a Whole-System Scope

- According to Ring [5], Whole-System Modeling is a “candidate response” for dealing with the persistent under-conceptualization inherent in traditional systems engineering.
- Most system problems experienced today are “N-body” system problems that co-learn, are agile, and require a model-based method of practice



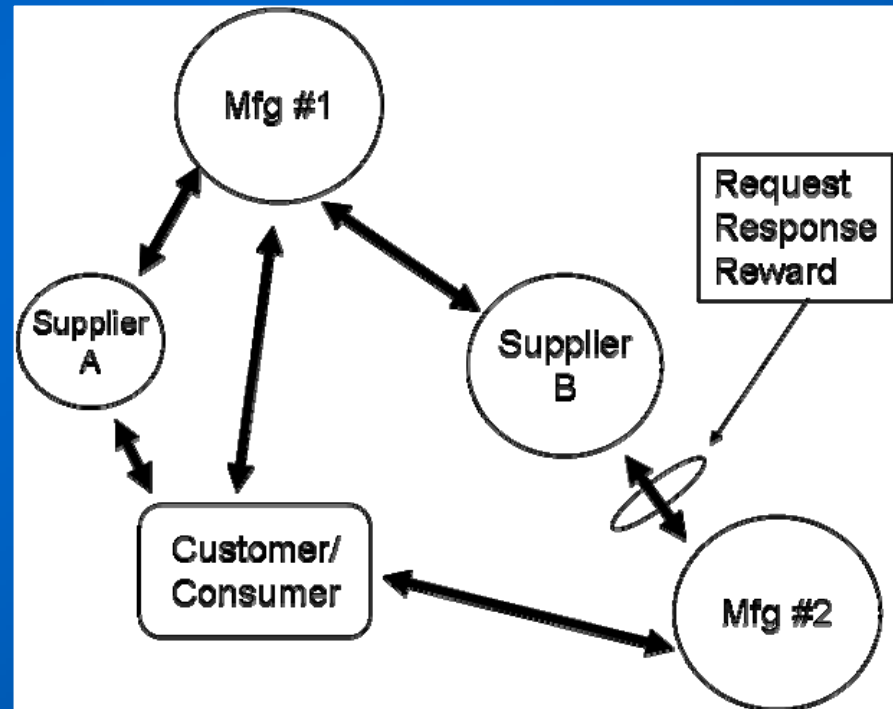
Applying OpEMCSS for WSM

- The Vee[^] of an Executable and Integrated SE Process for WSM



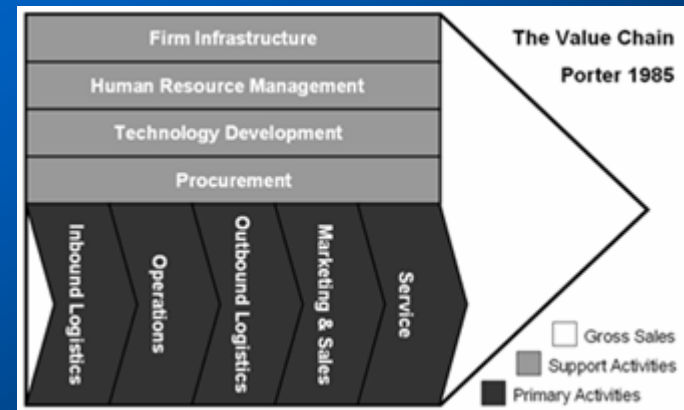
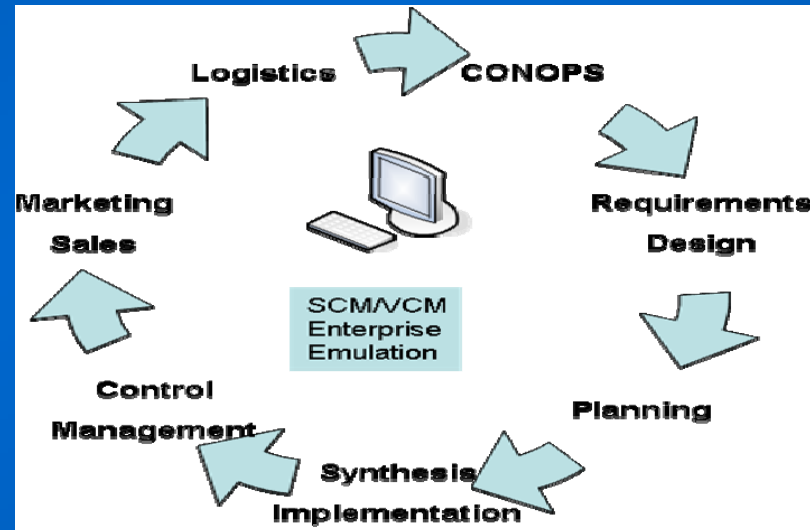
Supply Chains as systems of systems

- Chain consists of multiple agents
- Approach as a system of systems engineering problem
- Adaptation and intelligent behavior are key characteristics in understanding the project
- Agent Based Systems are Intelligent and Adaptive
- MBSE facilitates modeling and simulation of the system for better communication, control, planning and management



MBSE for SCM and VCM

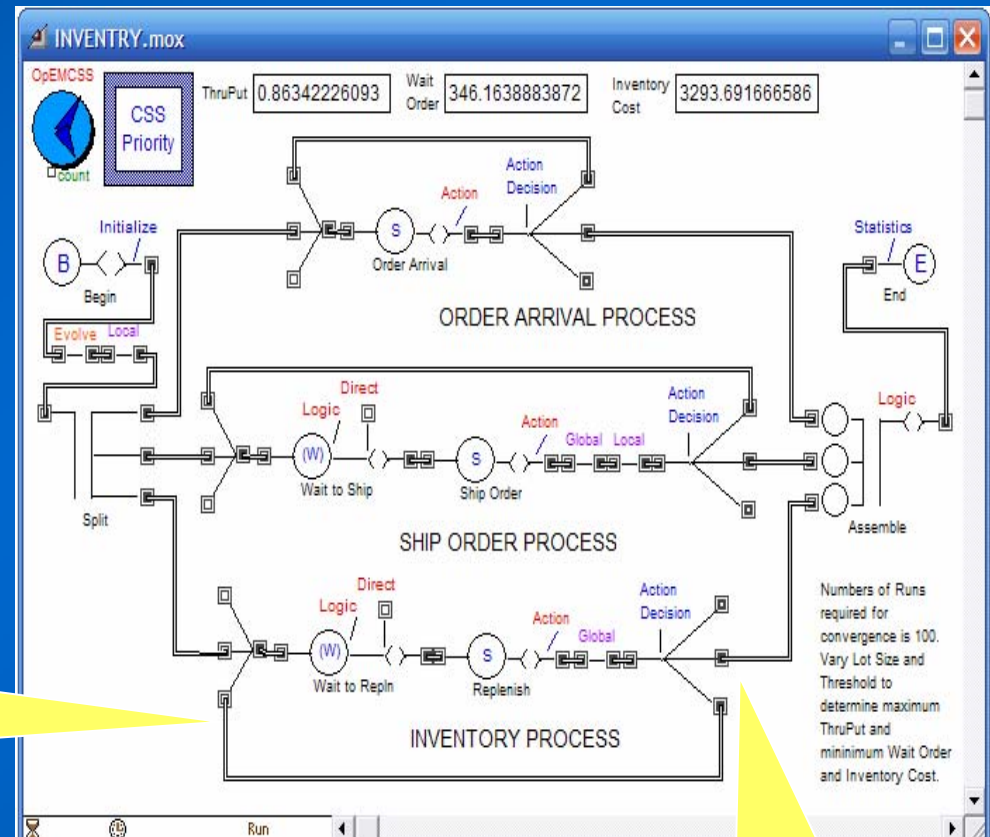
- Suppliers & buyers act like intelligent agents, assisting or disrupting a complex supply and value chain.
- Supply chain and value chain managements (SCM & VCM) are system of systems engineering problems
- **Solution:**
 - Take a model-based systems engineering approach
 - Model the agents within the SCM and VCM as intelligent enterprise
 - Their operation and management can be simulated and emulated
- **Our Goal:**
 - Ensure that the process of moving material, products, and knowledge, from source to destination is an intelligent and evolvable process that adds value at every step and learns for a continuous and incremental process of improvement.



Inventory Control Example

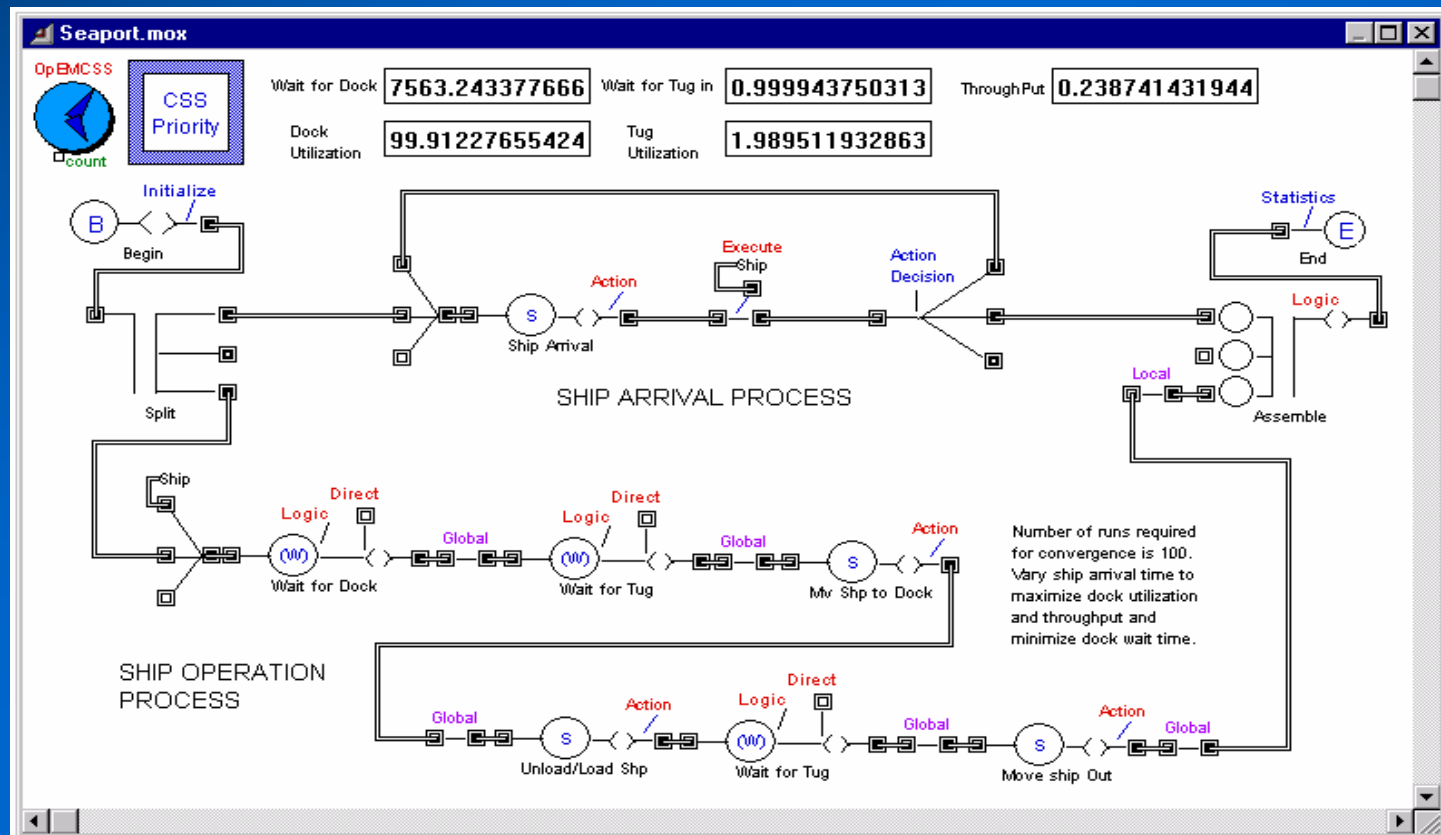
- This operational model includes an evolutionary algorithm block for optimization, that includes three concurrent processes
- Event Occurrence lets us merge multiple inputs together
- Alternate Action lets us select one of three paths

Event Occurrence



Alternate Action

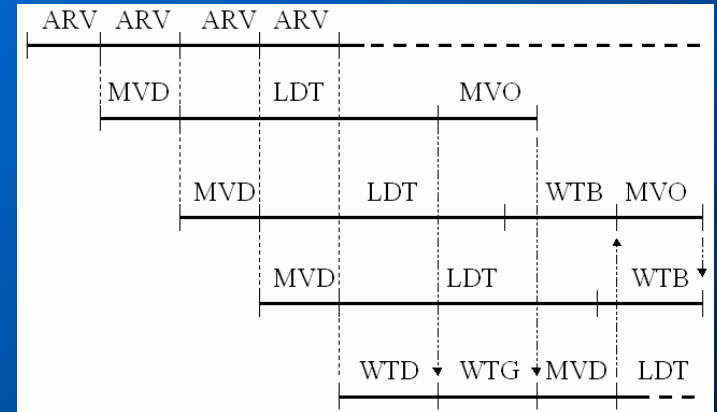
Transportation Example: A Sea Port



Time-lines of concurrent states

- Within the same process, group sequential tasks.
 - Ship Arrival Process:
 - ARV
 - Ship Operation Process:
 - WTD, WTG, MVD, LDT, WTB, MVO

- ARV: inter-arrival time for ships to enter system
- WTD: wait for dock to become available
- WTG: wait for tug boat to move ship into dock
- MVD: tug move ship to dock
- LDT: unload and load cargo
- WTB: wait for tug boat to move ship away from dock
- MVO: move ship out of harbor



Summary

- The purpose of this study was to provide and define an executable and integrative approach to Whole-System Modeling. This was accomplished by validating the hypothesis in the following three ways:
 - Developing a Holonic Systems Engineering process via holonic requirements.
 - Applying OpEMCSS, as a feed-back and feed-forward element to the Vee systems engineering process.
 - Highlighting how executable and integrative modeling, as applied by OpEMCSS enables a Whole-System Modeling construct.
- It is the recommendation of this study that OpEMCSS be adapted into the systems engineering process, and adopted as the Whole-System Modeling construct that provides executable and integrative modeling, for a holonic systems engineering process.