A Generic, Adaptive Systems Engineering Information Model
Overview of Discussion (1 of 2)

- Definition of Systems Engineering
- Motivation for Generic, Adaptive Systems Engineering Information Model
- Literature Review
  - Systems Engineering History
  - Systems Engineering Current Practice
  - Systems Engineering Standards
  - Requirements Models
  - Process Models
  - Information Models
Overview of Discussion (2 of 2)

- Relational Database Selection
  - MySQL Database
  - HSQL Database
  - PostgreSQL Database
- Model Development
- Logical Data Model
- Discussion
- Summary
Definition of Systems Engineering

- Systems engineering is a structured technical management and control process used in the design, development, production and operation of large-scale complex systems.
Motivation for SE Information Model

- Integrate different types of systems engineering tools
- Reduce risk of supportability and interoperability problems
- Support wide range of computer-based SE tools
- Leverage open standards and applications
- Facilitate loosely coupled information constructs
- Support viable operation over total system lifecycle
- Provide flexible application development and deployment
- Enforce basic system rules and processes
- Support new models from users and development partners
- Encourage incremental development and deployment
- Enable activity and task pattern recognition
# Three Tier System Types

<table>
<thead>
<tr>
<th>Client Tier</th>
<th>Application Tier</th>
<th>Database Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Application</td>
<td>Client Application</td>
<td>Database 1a</td>
</tr>
<tr>
<td>Executable Model</td>
<td>Client Application</td>
<td>Database 1b</td>
</tr>
<tr>
<td>Application Server</td>
<td>Application Server</td>
<td>Database 2a</td>
</tr>
<tr>
<td>Enterprise Beans</td>
<td>File Server</td>
<td>Database 2b</td>
</tr>
<tr>
<td>Dynamic HTML Pages</td>
<td>Java Server Pages</td>
<td>Database 3a</td>
</tr>
<tr>
<td>Executable Model</td>
<td>Enterprise Beans</td>
<td>Database 3b</td>
</tr>
<tr>
<td>Client Application</td>
<td>Application Server</td>
<td>Database 4a</td>
</tr>
<tr>
<td>Executable Model</td>
<td>File Server</td>
<td>Database 4b</td>
</tr>
</tbody>
</table>

- **System 1**
- **System 2**
- **System 3**
- **System 4**
Systems Engineering History

- Large scale civil and military projects
- United States Military:
  - Department of Defense MIL-STD-499 1969
  - Army Field Manual 770-78 “Systems Engineering”
Current Systems Engineering Practice

- Large scale civil and military projects
- Different customer types and expectations
- Common Systems Engineering Standards
  - Electronics Industries Association (EIA) 632
  - Institute of Electrical and Electronic Engineers (IEEE) 1220
  - EIA 731 “Systems Engineering Capability Model”
System Software Development

- Modern military projects with high percentage of software
- DoD-STD-2167A, mandated requirements traceability
- Based on a document centric view of requirements management
- No mandated process, method or approach.
Requirements Traceability Studies

- Naval Postgraduate School Requirements Traceability studies
- Based on links between text documents.
- Four Models: (High End User Models)
  - Requirements Management Model
  - Design Allocation Model
  - Design/Implementation Decision Making Model
  - Compliance Verification Model
Design Allocation Model

- **Source**
  - Standards/Policies/Methods
    - constrain

- **Requirements**
  - drive
    - allocated to
      - System/Subsystem/Components
        - part of
          - is a
            - Resources
              - used by
                - System/Subsystem/Components
                  - is a
                    - is a
                      - Standards/Policies/Methods

- **Functions**
  - satisfy
    - is a
      - Stakeholders
        - responsible for
          - System/Subsystem/Components
            - part of
              - External Systems
                - depend on
                  - System/Subsystem/Components
                    - defined by
                      - Resources
                        - allocated to
                          - System/Subsystem/Components
                            - used by
                              - Stakeholders
                                - responsible for
                                  - System/Subsystem/Components
                                    - is a
                                      - Standards/Policies/Methods
                                        - constrain

- **Design/Implementation**
  - see figure 3
Design and Implementation Decision Making Model

- Assumptions
- Rationale
- CSF
- Decisions
- Stakeholder

- Design/Implementation
- Issues/Conflicts
- Alternatives

- Based on
- Supported by
- Influenced by
- Made by
- Refine
- Resolve
- Evaluate

(from figure 2)
Compliance Verification Model

Requirements → Standard

System/Subsystem/Components → Requirements

verify

developed for

Compliance Verification Procedures

identify
allocated to

Changes

Test

is a

based on

Inspection

is a

Simulation

Prototype

Resources

comply with

satisfy

verify

(c) 2005 Joseph J. Simpson
# SE Process Implementation

<table>
<thead>
<tr>
<th>Steps to Apply Scientific Method to Problem Solving</th>
<th>Early Man Developing Cultural Patterns</th>
<th>Basic Research</th>
<th>Operations Research</th>
<th>System Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Recognize problem</td>
<td>Unsatisfied Physiological Need</td>
<td>Identify gap in body of scientific knowledge</td>
<td>Identify operational objective to be achieved</td>
<td>Describe mission or use requirements</td>
</tr>
<tr>
<td>2 Describe problem</td>
<td>Discover alternative ways to increase satisfaction of need</td>
<td>Develop theory of probable cause and effect relationships</td>
<td>Define situation &amp; resources which can be used to attain objectives</td>
<td>Define req'd operation and logistic functions to attain use objectives</td>
</tr>
<tr>
<td>3 Select hypothesis for solving problem</td>
<td>Select favored way of satisfying need</td>
<td>Select hypothesis for investigation</td>
<td>Describe tailorable variables to achieve desired objectives</td>
<td>Specify the system performance / design requirements</td>
</tr>
<tr>
<td>4 Develop model for testing hypothesis</td>
<td>Devise implements &amp; techniques to practice favored way</td>
<td>Describe experimental model to test hypothesis</td>
<td>Construct statistical model to interrelate variable conditions</td>
<td>Accomplish detail design &amp; qualification testing of components</td>
</tr>
<tr>
<td>5 Conduct tests under controlled conditions</td>
<td>Use selected techniques for some period of time</td>
<td>Conduct controlled lab/field investigation to obtain data</td>
<td>Perform computation to obtain statistical values</td>
<td>Build, assemble, test complete prototype system</td>
</tr>
<tr>
<td>6 Analyze and evaluate test data</td>
<td>Decide if techniques result in tolerable satisfaction of need</td>
<td>Analyze and evaluate collected data</td>
<td>Analyze and evaluate summary statistical data</td>
<td>Analyze and evaluate test data</td>
</tr>
<tr>
<td>7 Derive conclusions to confirm, deny, modify hypothesis</td>
<td>Transmit techniques to others &amp; establish cultural pattern</td>
<td>Derive conclusions to confirm, deny, modify hypothesis</td>
<td>Recommend actions to achieve desired objectives</td>
<td>Recommend modifications for production system</td>
</tr>
</tbody>
</table>

SE Information Views

1.0 Assess Available Information

2.0 Define Measures of Effectiveness

3.0 Create What Model

4.0 Create How Model

5.0 Perform Trade-off Analysis

6.0 Iterate to Find Feasible Solution

7.0 Create Sequential Build Plan

Core Information Models

- Seven Core Information Models
  - System Behavior
  - System Input and Output
  - System Structure and Behavior
  - System Requirements
  - Effectiveness Measure Creation
  - Text Requirements, Behavior and Content
  - Build and Test Plan
SE Information Views

- Joint Technical Architecture
- Department of Defense Architecture Framework
- Function, Requirement, Architecture, and Test
- Environment, Informational, Functional, Behavioral, and Implementation
- Logical, Process, Physical, Development, and Scenario
SE Requirements Models

- Historically Text Based
- Support SE Process Model Steps
- History of Semantic Confusion
- Executable Requirements Models Are Needed
- RDD-100, IDEF0 Tools
- UML 2.0, SYSML
- Custom Built Requirements Tools
SE Process Models

- Different types, military and civilian
- Most SE process models are flexible
- Two main areas of application:
  - System under design
  - Design support systems
- Model driven design process
- System architecture
- Detailed phase and support process models
SE Information Models

- Relational and object-oriented databases are most often used on SE projects.
- Some basic design approaches:
  - Model based on SE process
  - Model based on:
    - Product architecture
    - Process architecture
  - Two groups of relational tables:
    - Management activities
    - System architecture evolution
Relational Database Management Systems

- Three Open Source Relational Database Management Systems:
  - MySQL Database
  - HSQL Database Engine
  - PostgreSQL Object Relational Database System
    - ACID Transactions
    - SQL 92
    - SQL 99
Global Model Development

- Generic SE Information Model Design Criteria
- Conceptual Information Model Design
- Base Systems Conceptual Models
  - System Context View Data Model
  - System Concept View Data Model
  - System Functional View Data Model
  - System Requirement View Data Model
  - System Architecture View Data Model
  - System Test View Data Model
Logical Model Development

- Logical Models for the “three basic systems”
  - Environmental System
    - Includes all other systems
  - Product System
    - The system under design
    - What the customer wants
  - Process System
    - The system that produces the product system
    - Includes people, equipment and processes.
Basic System Model Relationships

1.0

1.1
1.2
1.3
1.4

2.0

A.a

(c) 2005 Joseph J. Simpson
Basic System Model Relationships
Logical Model Development

- Logical data models were developed for the following entities:
  - System
  - Context
  - Concept
  - Function
  - Requirement
  - Architecture
  - Test
Logical data models were developed for the following entities:

- Decision
- Type
- Document
- Model
- Database model
- Fr_link
- Fa_link
- At_link
Context ER Model

document -- 1 --> context

context -- N --> concept

context -- 1 --> model

context -- 1 --> db_model

type -- 1 --> context

system -- 1 --> context

model -- N --> context

db_model -- 1 --> context
Concept ER Model

- **concept**
  - **type**
  - **model**
  - **db_model**
  - **function**
  - **requirement**
  - **architecture**
  - **test**
  - **decision**
  - **document**
  - **context**
Function ER Model

![ER Diagram]

- **model**: 1 to 1
- **db_model**: 1
- **function**: 1
- **type**: 1
- **concept**: 1
- **decision**: N
- **document**: N
- **fr_link**: N
- **fa_link**: N

(c) 2005 Joseph J. Simpson
Requirements ER Model

- **requirement**
  - **type**: 1
  - **fr_link**: N
  - **model**: N
  - **db_model**: 1
  - **decision**: N
  - **document**: N
  - **concept**: 1
Architecture ER Model

[Diagram of an Architecture ER Model]

- type
- concept
- model
- db_model
- decision
- document
- architecture
- at_link
- fa_link

- 1:1 relationship between type and concept
- 1:N relationship between architecture and decision
- 1:N relationship between architecture and document
- 1:1 relationship between model and architecture
- 1:1 relationship between db_model and architecture
- 1:1 relationship between at_link and architecture
- 1:1 relationship between fa_link and architecture
Test ER Model

![ER Diagram]

- **type**
  - 1
  - 1
- **at_link**
  - N
  - 1
- **model**
  - N
  - 1
  - 1
- **db_model**
  - 1
  - 1
  - 1
- **decision**
  - N
  - 1
- **test**
  - 1
  - 1
  - 1
  - 1
- **document**
  - N
  - 1
- **concept**
  - 1
Summary

- Flexible conceptual data model
- Can be applied and adapted to any type of system project
- Provides a connection to standard networked information system applications.
- Provides basis of automation of SE tasks and activities
Next Steps

- Select components for the application server tier and client tier
- Design an incremental, spiral approach to the development of the application logic and data connection to the application server tier
- Incrementally develop and test application components.
Conclusions

- Standards-based SE tools, utilizing standard computer languages, reduce the risk of unsupported, unusable systems information data stores.

- The systems engineering information models developed in this work provide a foundation for a wide range of standards-based SE tools and data stores.
Questions?