A Three-Dimensional Model for System Design Evolution

Prof. Stephen R. Schach
Vanderbilt University, Nashville, TN
Dr. Amir Tomer
Rafael Ltd., Haifa, Israel

The 1st National Conference on Systems Engineering
INCOSE_IL 2002
Hertzeliya, Israel, 12 March, 2002

Overview

- The systems engineering process
- One-dimensional system development
- Two-dimensional system development
- Three-dimensional system development
- Controlling system design in the three-dimensional evolution space
- Conclusions
The Systems Engineering Process (SEP)*

- A generic problem-solving process, which provides the mechanisms for identifying and evolving the product and process definitions of a system.

- Applies throughout the system life cycle to all activities associated with product development, verification/test, manufacturing, training, operation, support, distribution, disposal, and human systems engineering.


System Life Cycle [IEEE-1220-1998]
### Artifacts

- **Tangible pieces of information**
  - Used or produced by the development process
    - Documents
    - Models
    - Code modules

### Simplified SEP and its application to Systems and Software

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activities</td>
<td>Artifacts</td>
</tr>
<tr>
<td>Requirements</td>
<td>Requirements analysis</td>
<td>Requirements baseline and specification</td>
</tr>
<tr>
<td>analysis (R)</td>
<td></td>
<td>Software requirements analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software requirements description and specification</td>
</tr>
<tr>
<td>Architecture (A)</td>
<td>Functional analysis</td>
<td>Functional architecture</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software architectural design</td>
</tr>
<tr>
<td>Design (D)</td>
<td>Synthesis</td>
<td>Physical architecture</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Software detailed design</td>
</tr>
<tr>
<td>Coding (C)</td>
<td>(None)</td>
<td>Coding and unit testing</td>
</tr>
<tr>
<td></td>
<td>(None)</td>
<td>Code</td>
</tr>
<tr>
<td></td>
<td>(Applies only to software, and on only the component level)</td>
<td></td>
</tr>
</tbody>
</table>
One-Dimensional System Development ("Once Through")

RA = Requirements Artifacts
AA = Architecture Artifacts
DA = Design Artifacts
CA = Code Artifacts

Problem: No support for changes

A case study (development scenario)

1. Requirements Analysis → RA completed
2. Architecture → AA completed
3. Design → DA in process
   - Requirements change requests
4. Requirements Analysis → RA modified
5. Architecture → AA modified
6. Design → DA completed
7. FAIT → product implementation in process
   - Security constraints violated
8. Design → DA modified
9. FAIT → product implementation completed
   - Product approved for production (PRR passed)
The Waterfall Model [Royce, 1970]

- Support for changes ("unforeseen difficulties")
- Changes may be introduced at any phase
- Feedback loops to previous phases.

Problem: Timing and order of feedback loops not modeled

Two-Dimensional System Development

The Evolution Tree of System Design

Configuration baselines:
RA₀-AA₀-DA₀
RA₁-AA₁-DA₁-CA₁
RA₂-AA₂-DA₂-CA₂

Schach & Tomer - INCOSE_IL 2002 3D System Design Evolution - 9
Product Lines and Reuse

- **Product line:**
  - "A set of [software-intensive] products... sharing a common, managed set of features"
    - Clements & Northrop – A framework for software product line practice

- **Reuse**
  - Black-box reuse
  - Glass-box reuse
  - COTS

- **Reusable design artifacts**
  - Architectures
  - Mechanical mechanisms
  - Chemical processes
  - Electronic design
  - Algorithms
  - ...

Each product in the product-line has its own evolution tree

Reusable artifacts transfer between products

An asset repository may be utilized to manage all reusable artifacts

Three-Dimensional System Development

- Mining an existing asset
- Reuse
- Reengineering
- Acquiring a core asset
- Core assets
- Engineering
- progress
- diversity
- effort

Schach & Tomer - INCOSE_IL 2002
Artifact Location in the 3D Evolution Space

Each individual artifact is located in the 3D system evolution space. Its "coordinate" is denoted by a triple (T,V,P), as follows:

- **T = Type**
  - The location along the engineering axis
  - Types should be ordered
  - Other type examples:
    - Inception/elaboration/construction/transition (RUP)
    - Use-cases/sequence-diagrams/code-modules (UML)
- **V = Version**
  - The location along the reengineering axis
  - Usually a decimal number
  - Specifies the time ordering of artifacts of the same type
- **P = Product**
  - The location along the reuse axis
  - Specifies the identification of the specific product library in which the artifact resides

Example: ("functional architecture", 3.7,"RS for navy")

The Three Basic Operations of System Design

- **B = modify(A)**
  - Artifact B is obtained by making modifications to artifact A
  - A "move" along the reengineering axis.
    - \((t',v,p') = modify(t,v,p)\)
- **C = copy(A)**
  - Artifact C is an exact replica of artifact A
  - A "move" along the reengineering axis
    - \((t,v,p') = copy(t,v,p)\)
- **D = generate(A)**
  - An artifact D is generated from another artifact A.
  - Example: An architecture is generated from the requirements document
  - A "move" along the engineering axis
    - \((t',v,p) = generate(t,v,p)\)
Conclusions

- System design is an evolutionary process
- The system design evolution may be described in a three-dimensional space
- Every artifact (either actual or historical) is "located" in the 3D evolution space
- The location of an artifact specifies it uniquely
- The 3D design evolution model provides a means for controlling design artifacts throughout entire life cycle

Contacts

Dr. Amir Tomer
Director of Systems & Software Engineering Processes
RAFAEL
P.O.Box 2250/1P
Haifa, 31021
ISRAEL
(+972)-4-879-5989
tomera@rafael.co.il