Advanced Virtual Engineering Centre

Systems-of-Systems Design Using Architecture Patterns

Demetrios Joannou
d.joannou@lboro.ac.uk

http://www.lboro.ac.uk/research/avrrc/research/currentprojects/modelingandsimulation/
Overview

• Introduction to Systems of Systems
  • Systems of Systems
  • Designing For Adaptability and Evolution in Systems of Systems (DANSE Project)

• Architecture Patterns for SoS Development
  • Architecture Patterns and SoS Design
  • Mining Architecture Patterns

• Online Architecture Patterns Repository
  • Architecture Patterns Classification
  • Architecture Pattern Anatomy
  • Using the Repository
  • Applying Architecture Patterns to an SoS Design Project

• Architecture Patterns Examples
Some of the complexity issues

• Possibly no conscious knowledge of other systems
• Operational independence,
• Geographically distributed,
• Constantly Evolving,
• Independent Governance, …
DANSE - Designing for Adaptability and evolution in System of systems Engineering

November 2011 – October 2014
DANSE aims at developing new approaches to the design and management of the operation of SoS based on advanced methodologies based on a new evolutionary:

- Adaptive and iterative SoS life-cycle model;
- Semantically sound models based on the notion of contracts;
- Innovative architectures that provide the infrastructure to allow the dynamic affiliation of components so that the behaviour of the ensemble is not disturbed;
- Novel supporting model based tools for analysis, simulation, and optimization;
- Organized in an integrated environment.

DANSE focuses on the development of a new methodology to support evolving, adaptive and iterative SoS life-cycle models based on a formal semantics for SoS inter-operations and supported by novel tools for analysis, simulation, and optimisation.
To achieve these challenges DANSE addresses the following innovation points:

• A new evolutionary, adaptive and iterative SoS life-cycle model, offering evolutionary simulation, analysis, and development of the SoS in real time.

• Development of new formal semantics for SoS modelling based on an architectural framework.

• A formal method for “correct by evolution” analysis which constantly guarantees that the SoS evolves towards its current goals.

• High-level behavioural simulation based on SoS abstraction using theoretical relationships rather than detailed event-level simulation.

• Methods and tools to allow optimisation at the global SoS level and at the local level of constituent systems, in face of possibly contradicting goals.
DANSE Technical Approach – Modelling and Simulation

- Design Exploration
  - C4I System – centralized w/ LTE
  - C4I System decentralized w/ TETRA

- Architecture Modelling
  - UPDM/ NAF etc

- Run Time analysis

- Component Data
  - GCSL
  - Graph Grammars
  - Architecture Patterns
  - Agent-based Sim
  - Toolnet-connections

- Enablers

- Operational Sequence/ Behaviour
  - Objectives Specification

- Co-Simulation

- FMI
  - SV-10c (e.g. SysML State-Machine)
  - Performance
  - TBD

- Emergent Behaviour Analysis
  - Human Aspects (Decisions)
  - Performance Analysis
  - Fire Emergency Response
  - Emergency (x, y, z, Day/Time)
  - Vehicle Average Velocity Distributions

- Statistical Models
  - CS native models
Patterns are not new – they have been in existence for centuries

Design patterns used extensively by software engineering

However, Architecture Patterns are new

Patterns encapsulate considerable knowledge from experienced practitioners

Patterns are templates or recipes to describe solutions to known problems (relate Context, Problem & Solution in a consistent manner)

Provide a generalised rule or guideline for realising certain architecture design characteristics.

Patterns allow specification of different architectural approaches

Support abstract representations that facilitate greater understanding of complex SoS architectures
SoS is modelled using an architecture framework

Iterative process until desired global characteristics implemented
Architects identify a series of patterns which offer desired qualities and attributes

Selected patterns implemented in architectural framework and analysed
Architecture Patterns for SoS Development

- Candidate architectures submitted to simulation environment
- Architecture evaluated against a set of SoS constraints and target functions
- Alternative patterns substituted if a particular solution does not converge towards the desired target
Candidate architectures submitted to simulation environment

Architecture evaluated against a set of SoS constraints and target functions

Alternative patterns substituted if a particular solution does not converge towards the desired target

SoS Design Space Exploration
Architecture Patterns for SoS Development

- Candidate architectures submitted to simulation environment
- Architecture evaluated against a set of SoS constraints and target functions
- Alternative patterns substituted if a particular solution does not converge towards the desired target

SoS Design Space Exploration Iterations

SoS Design Space Exploration

SoS Constraints and Target Functions

Capability Goals

Analysis & simulation

Performance of alternatives

Joint simulation

Emergent behaviour

Architecture Expression

SoS Candidate Architecture (UPDM)

Patterns Repository

Selected Patterns
Mining Architecture Patterns

- Mining patterns is an iterative process
  - There is an art to mining patterns
  - Patterns have structure
  - Remember patterns are small reusable elements that are generally repeatable in structure
- Think simple - an elegant pattern is characterized by its simplicity

- Architecture Patterns are abstractions
  - Don't go into specific implementation details
  - This is too deep
  - Note some patterns may look similar but can actually be radically different
- New patterns can be created or evolve from pre-existing patterns
Mining Architecture Patterns

1. Consult SME(s) (if possible)

2. Create Operational Viewpoint
   - Obtain CONOPS Documentation (if available)
   - Identify Global and Local Goals
   - Construct Initial Operational/Structural View (Using UPDM or Mind mapping)

3. Identify Systems Features in Scope
   - Identify Fundamental Functional Architectural Elements (e.g., Command & Control, Comms, Interfaces etc.)
   - Analyse/Identify Constituent Systems
   - Identify Fundamental Behavioural Architectural Elements

4. Undertake Architecture Pattern Elicitation
   - Extract Potential Architecture Pattern(s)
   - Complete Patterns Template
   - Compare Architecture Elements against Online Patterns Library
   - Existing Pattern Available

5. Create Architecture in Modelling Environment
   - SoS Architecture Patterns Library
   - Create New Pattern and Update Patterns Library
   - Non Existing Pattern

SoS/Constituent System Under Consideration
Mining Architecture Patterns

Consult SME(s) (if possible)

Create Operational Viewpoint
- SoS/Constituent System Under Consideration
  - Obtain CONOPS Documentation (if available)
  - Identify Global and Local Goals
  - Construct Initial Operational/Structural View (Using UPDM or Mind mapping)

Identify Systems Features in Scope
- Identify Fundamental Functional Architectural Elements (e.g., Command & Control, Comms, Interfaces etc.)
  - Analyse/Identify Constituent Systems
  - Identify Fundamental Behavioural Architectural Elements

Undertake Architecture Pattern Elicitation
- Extract Potential Architecture Pattern(s)
  - Complete Patterns Template
  - Compare Architecture Elements against Online Patterns Library
  - Existing Pattern Available
  - Non Existing Pattern
  - SoS Architecture Patterns Library

Create Architecture in Modelling Environment
- Construct SoS Architecture
Mining Architecture Patterns

Consult SME(s) (if possible)

Create Operational Viewpoint
- Obtain CONOPS Documentation (if available)
- Identify Global and Local Goals
- Construct Initial Operational/Structural View (Using UPDM or Mind mapping)

Identify Systems Features in Scope
- Identify Fundamental Functional Architectural Elements (e.g., Command & Control, Comms, Interfaces etc.)
- Analyse/Identify Constituent Systems
- Identify Fundamental Behavioural Architectural Elements

SoS/Constituent System Under Consideration

Undertake Architecture Pattern Elicitation
- Extract Potential Architecture Pattern(s)
- Complete Patterns Template
- Compare Architecture Elements against Online Patterns Library
- Existing Pattern Available
- SoS Architecture Patterns Library
- Non Existing Pattern
- Create New Pattern and Update Patterns Library

Create Architecture In Modelling Environment
- Construct SoS Architecture
Operational, Functional and Systems Patterns Use within Architectural Framework

- OV-1 High Level Operational Concept
- OV-5b Operational Activity Model
- SV-4b Systems Functionality Description
- SV-1 Systems Interface Description
- SV-1 Systems Interface Description BDD
- SV-5a Operational Activity to Systems Function Traceability Matrix

Operational Patterns

Functional Patterns

Systems Patterns

AV-1 Overview and Summary Information

SV-10b Systems State Transition Description
Process

Step 1 – Produce or pull up SoS Architecture (UPDM) of ER SoS

Step 2 – Abstract CS of interest: Look at the Command System of the ER SoS

Step 3 – Set criteria for which new pattern must satisfy

Step 4 – Search pattern repository/database for Suitable SoS command patterns

Step 5 – Produce alternative architecture models

Step 6 – Plug in to Architecture

Original Command Pattern

UPDM Pattern Repository
Architecture Patterns Repository

Architecture Patterns repository includes larger catalog of patterns e.g. UPDM, SysML, Test Cases etc.

Sophisticated online repository for architecture patterns with powerful search capabilities, option to store new patterns.

The repository exists itself in three forms;

- a document-based repository,
- a repository of IBM Rhapsody profiles, and
- an online searchable repository with the option to download IBM Rhapsody SysML/UPDM profiles for inclusion in DANSE Tool-net.

Accessed via:

- Conventional web browser (all popular browsers supported),
- Apple iPad running the free FileMaker App – FileMaker Go.
- User run-time version of FileMaker
Summary of Online Repository

**Summary:**
This is an online repository for architecture patterns. Patterns are available to view with full details. Patterns have been classified into Root architecture patterns and Specific architecture patterns. Patterns have been further classified according to the domain of use and the users can access them according to their requirements. For each pattern a section giving its background information and a section describing how to use that pattern in the design and analysis of a SoS has been given. Patterns have been supplemented with examples demonstrating how the subtle differences in their characteristics can influence the decision-making process of the architect. The library can be used to search for patterns, to understand where to use them effectively and to download SysML profiles. In due course, the library will also allow the users to add new patterns and modify or delete existing patterns. This repository is subject to frequent improvement.

**Catalogues:**

**Root Architecture Patterns**
- Command, Control and Execution Architecture Patterns
- Communication Architecture Patterns
- Interface Patterns
- Security and Authorisation Patterns
- Resilience Patterns
- Contractual Specification Patterns
- Service Oriented Architecture Patterns
- User Interaction Patterns

**Specific Architecture Patterns**
- Command, Control and Execution Architecture Patterns
- Communication Architecture Patterns
- Interface Patterns
- Security and Authorisation Patterns
- Resilience Patterns
- Contractual Specification Patterns
- Service Oriented Architecture Patterns
- User Interaction Patterns
- Design Space Exploration Patterns

**Domain Specific Architecture Patterns**
- Water Supply and Distribution Patterns
- Air Transport System Patterns
- Ground Transport System Patterns
- Operational Patterns (DANSE - Concept Alignment Example)

**User Guides:**
- Guide for Web Browser Users
- Guide for iPad Users

**Contact**
For any query, please feel free to contact any of us:
- Roy S. Kalawsky
  r.s.kalawsky@lboro.ac.uk
- Demetrios Ioannou
  d.ioannou@lboro.ac.uk
- Antara Bhatt
  a.bhatt@lboro.ac.uk
Any key words that may appear in the pattern that will be useful when looking up the pattern in a repository.

The Author of the Pattern

This refers to the problem and why you would use the pattern to address the issue.

Also known as.

Statement of why the pattern would be utilised to address the design problem or situation. It will help understand the structure and consequences later in the pattern.

Name of Pattern

Centralised Command and Control Pattern

Keywords

Command, Centralised, Control

Author

Loughborough Uni

Text

Exercise of authority (invention, advice, opinion, influence, or command) and direction by a control system over assigned resources to achieve accomplishment of the specified mission. The Central Command/Control System governs and exercises full authority over resources.

Also Known

CCC, C3

Motivation: Goals

The pattern allows for a single command centre, which has unquestionable overall authority.

Motivation: Capabilities

1. Many points of intelligence access the Centralised Command System, allowing for a more informed decision making process.

2. Centralised command facilitates all knowledge to be in one central location, resulting in more accurate decision making.

Using the Pattern

Structure

Applicability

Root Architecture Patterns

Models

Model (SysML)

Model (SysML+Concise)

Model (UPDM)

Model (UPDM+Concise)

Model (Other)

Rhapsody Models Available for Download
Here, not only are the participants being shown that make up pattern, but also how, and with which other elements are interacting, describing briefly the relationship between elements which facilitate the tasks they need to conduct.

### Motivation: Capabilities

| 1. Many points of intelligence access the Centralised Command System, allowing for a more informed decision making process. |
| 2. Centralised command facilitates for all knowledge to be in one central location, resulting in more accurate decision making. |
| 3. Excellent planning and tasking potential. |
| 4. Command System able to rank requests in order of highest priority and able to act accordingly. |

### Motivation: Limitations

| 1. For Command/Control to be effective, the command and control information network must be “interoperable, sustainable, and survivable”. |
| 2. A single command/control system is subject to serious common mode failure and is completely dependent on the Command/Control System being available. |
| 3. A single command/control system is extremely cost effective in terms of interfaces but due to point 2 above it may be necessary to consider alternative more resilient architectures variants. |

### Participants

- Central Command System, Command Systems

### Collaborators

- Central Command System – Command System: Centralised Command System passes orders on to the appropriate command system to achieve a specific result, mission or goal. Command Systems may act as a point of intelligence to the Central Command system, collecting “on-scene” information or data to help co-ordinate tasks and to reallocate resources for example if need be.

### Performance Metrics

- Associated metrics e.g. bandwidth, response time, cost, redundancy level, etc.

### Implementation

- On implementation, consideration should be taken on the following issues:
  1. Commands to other command systems need to be unambiguous but also considering how intelligent a command should be, for security concerns.
  2. The capacity of command systems a central command system can cope with in a high demanding situation.
  3. Whether or not the centralised command pattern will facilitate for a fast or slow response time depending on SoS goals.

### Emergent Properties

- Information will be provided after simulation has been run.

### Example(s)

- Emergency Response agencies (fire departments, police departments and medical services) are sometimes controlled by a single overarching control centre with authority over all three. Knowledge and intelligence is collected and manipulated by the single command center. Decisions are made as to which resources should be deployed.

### Advice/Guidance on the Usage of the Pattern

- Provides some considerations to be undertaken when the pattern is applied.
The consequences refer to differing variables that may influence the usage of the pattern. What aspect of the pattern structure does it allow you vary in order to fit your specific application?

URL References to related materials

An example where the pattern is in use.

If the pattern has stemmed down from an original pattern, or patterns. Which are these?

Any patterns that may have been form from the pattern.

Catalogue which the pattern belongs to or can be found

Where the pattern has known to be used in real-life scenarios and in which domains. E.g. Military, Emergency Services.
## SOS Architecture Patterns

<table>
<thead>
<tr>
<th>Pattern Name</th>
<th>Centralised Command/Control HQ and Communications (TETRA/LTE) Network Pattern [CAE Optimisation Pattern]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keywords</td>
<td>Command, Communications, Telecommunications, Emergency Response,</td>
</tr>
<tr>
<td>Author</td>
<td>Loughborough Uni</td>
</tr>
</tbody>
</table>

### Pattern Information

**Intent**
This pattern is to provide the structure of constituent systems in an emergency response SoS in which Concise Modeling stereotypes can be added to perform optimisation activities. Optimisation will consider both how the constituent systems of the emergency response work together.

**Also Known As**
Not applicable

**Motivation: Goals**
To provide two options of communications networks – Terrestrial Trunked Radio (TETRA) and Long Term Evolution (LTE) to support message sending and data transfer between the agencies making up an Emergency Response SoS.

**Motivation: Capabilities**
1. Provides the option between two communications networks.
2. The pattern can be used for optimisation activities in Concise Modelling.
3. The pattern allows for communication coverage for all agencies at all times if optimised effectively.

### Using the Pattern

#### Structure

- AntennaTE
- AntennaTetra
- FireStation
- PoliceStation
- Hospital
- Controller:LTE
- HQ
- Controller:Generic
- Area
- AntennaCoverage:TetraPre
- AntennaCoverage:LTEPre
- PrevAntenna:TetraInstall

#### Applicability
Specific Architecture Patterns

- Model (SysML)
- Model (SysML+Concise)
- Model (UPDM)
- Model (UPDM+Concise)
- Model [v]

The implementation of this pattern is likely to be multiple phase process as existing legacy systems may already be in place, for example, command centres, base station and...
Easy Deployment from Repository to Rhapsody
To import patterns profile into Rhapsody, the following steps are followed:

- Step 1: Download the pattern from the patterns repository to the local machine.
- Step 2: In Rhapsody project, select "File" --> "Add to Model".
• Step 3: In **Add to Model** dialogue, select the .sbs file of the pattern that needs to be imported and choose "As Unit" at right hand side instead of “As Reference”, then click the "Open" button.

• Step 4: Under profiles folder of the Rhapsody project, a package of the same name as the pattern is added. The patterns profile is ready to use now.
• SoSArchitecturePatterns.sbs has been downloaded from the repository and imported into the CAE UPDM model using ‘Add to model’.

• SoSArchitecturePatterns package is added under ‘profiles’.

• The structure of CentralisedCommandControlHQ AndTETRA/LTECommunicationNetwork pattern is represented here
Main structure diagram in CentralisedCommandControlHQAndTETRA/LTECommunicationNetwork pattern
Architecture Patterns for SoS Development

Candidate architectures submitted to simulation environment
Architecture evaluated against a set of SoS constraints and target functions
Alternative patterns substituted if a particular solution does not converge towards the desired target
Architecture Pattern Examples
Water Supply Strategic Grid Architecture Patterns

Pattern Information

Pattern Name: Water Grid with Borehole - Demand management between borehole and transmission mains supply

Keywords: Borehole, water grid

Author: Loughborough Uni

Intent: Transfer of water to a demand centre through water supply grid that carries purified water. Water is also supplied through a borehole located near the demand point. The pump could be used for water supply grids for transferring water from one region to the other.

Also Known As: Not applicable.

Supply and distribution of water to a demand center through a water supply grid. The water treatment works supplies water directly to the water supply grid which is distributed to different demand points. The demand center also receives supply from a borehole.

Motivation: Goals

- The grid carries purified water.
- Supply can be managed between the borehole and the transmission main depending on the demand.

Using the Pattern

Structure:

Water Treatment Works

Demand Centre

Borehole Supply

Applicability: Domain Specific Architecture Patterns

Model (SysML):

Model (OMG-Complex):

Model (UPDM):
Emergency Response Communication Patterns

ARCHITECTURE PATTERNS REPOSITORY

Pattern Name: Generic Tetra Communication Infrastructure for Emergency Response

Keywords: Tetra, Communications, Network, Emergency Response

Author: Loughborough Uni

Pattern Information:
The pattern provides the communications medium for an entire emergency response "SOS". This two-way transceiver specification allows for information exchange between constituent systems of the complete SOS during a major incident. The specification is

Intent:

Also Know:

Motivation: Goals:

1. The pattern allows for information exchange between the constituents of a single agency and also inter-agency communication.

2. Talk-Groups can be set up to include a large number of participants simultaneously.

Using the Pattern:

Structure:

Applicability:

Specific Architecture Patterns:

Model (SyMIL):

Model (SyMIL+Concise):

Model (IIP/OM):

Model (IPDM+Concise):

Using the Pattern:

Specific Architecture Patterns:

Upload Export

Upload Export

Upload Export

Upload Export
Emergency Response Communication Patterns
Air Traffic Control Patterns

ARCHITECTURE PATTERNS REPOSITORY

Pattern Name: Air Traffic Management (ATM) Organisational Structure

Keywords: Air Traffic Control, Centrally controlled

Author: Loughborough University

Pattern Information

Intent:
This is a generic architectural pattern of an air traffic management organisation structure for monitoring the overall air traffic flow in the National Airspace System.

Also Known As:
This pattern enables management, organisation and coordination between various control authorities in the air traffic management structure in order to provide traffic, separation services to air traffic participants. The constituent parts and their control hierarchy are as shown below:

Motivation: Gains:
- The hierarchy enables air traffic management personnel (ATM) to analyse demand in the system and implement initiatives that are then relayed to the air traffic controllers.

Applicability:
Domain Specific: Architecture Patterns

Using the Pattern

Air Traffic Control during various flight phases (Image: Courtesy www.kao.pt)
Concluding Remarks

- Architecture patterns are an excellent resource for the SoS architect
- Provide key building elements for SoS architectures
- Integrated with modelling and simulation environments
- Facilitate abstraction of complex systems
- Support SoS design space exploration
- Built from verified and validated designs
On going Research

• Continued development of ontological database for wider integration with other modelling and simulation tools

• Expansion of architecture patterns repository to include further patterns
  • Root patterns
  • Application specific patterns

• Inclusion of additional examples and performance metrics within patterns

• Release of patterns repository (post DANSE project)
Thank you for your attention