10TH INTERNATIONAL DESIGN STRUCTURE MATRIX CONFERENCE, DSM'08
11 – 12 NOVEMBER 2008, STOCKHOLM, SWEDEN

RE-ENGINEERING LEGACY KNOWLEDGE BASED ENGINEERING SYSTEMS USING DSM

Sreeram Bhaskara
Boeing

Keywords: Knowledge Based Engineering (KBE), DSM, Re-Engineering, Architecture Management, Lean, Lattix

1 INTRODUCTION

Knowledge Based Engineering (KBE) Design Systems capture all engineering knowledge as well as intent behind a particular design process in the form of a product knowledge model, which is then used to generate and evaluate variant designs or in design reuse. Complexity of a KBE system lies in the explicit representation of the ontology of design along with design data, rules, core logic, and interfaces as well as in the design software architecture itself. This paper presents a methodology to understand, analyze, reengineer and migrate legacy knowledge based engineering design systems using Design Structure Matrix (DSM).

2 COMPLEXITY OF KNOWLEDGE BASED ENGINEERING SYSTEMS

Software complexity arises from large number of dependencies between software modules [1]. Applying DSM methodology to complex software systems has been described by Lattix DSM technology. A KBE system is a combination of engineering design process and software development process. This results in a complex system that not only captures the engineering process but also the software architecture. System complexity increases as it evolves with time and with new requirements. Many KBE developers and domain experts are involved at various stages in the evolution of these systems. This in turn multiplies the complexity and makes it very difficult to understand the functional as well as the implementation architecture of these systems. DSM representation and analysis techniques are applied to reduce complexity and make complex KBE systems transparent.

KBE is one of the many legacy systems that are being used within Boeing business processes. We propose to use DSM for re-engineering legacy systems based on refined DSM architecture map and analyze refined architecture to migrate to new tools. And once we understand re-engineering process, we plan to apply DSM techniques to forward engineer new KBE systems by defining and generating new architectures using Lattix DSM tool. A decomposition structure of these DSMs is then fed to detailed KBE development activity. The methodology proposed in this paper has been primarily applied in re-engineering and/or refining legacy knowledge architectures and for migrating existing knowledge models to new design tools.

3 RE-ENGINEERING KBE SYSTEM: A DSM APPROACH

The re-engineering process is comprised of the following steps:

1) Basic understanding of legacy KBE system structure
2) Generating DSMs for sample legacy KBE systems using Lattix DSM tool
3) Analyzing, understanding and documenting KBE system architectures using DSM.
4) Applying DSM to refactor/re-structure/re-engineer these legacy systems to reduce complexity
5) Extracting domain specific knowledge i.e. design data and rules from KBE DSMs
6) Identifying and defining architectural and design rules, constraints as needed
7) Developing migration strategies for legacy systems using DSM

Initial focus of this activity has been in the analysis and migration of KBE systems used in Aircraft Fuselage Design. These systems are based on ICAD [2] a Knowledge Based Engineering Design System that has been commercially around for over two decades. In ICAD system a product model is developed that contains all engineering knowledge and intent behind a particular design process. Variant designs can then be generated by modifying initial design and configuration data. ICAD system provides an object oriented framework to represent all the design data, rules and knowledge used in a design process. The types of knowledge [3] that can be represented using a typical KBE System such as ICAD are:

Class (or Part) and attributes, design procedures (or methods) relationships
Part, Sub part and Super part (or Mixin) relationships
Product configuration logic
Geometric specifications and modelling methodology
Design rules, Mathematical expressions
Modular Systems structure
Interface with external systems (CAD Systems, PDM Systems, Databases etc.)

For ICAD based legacy systems, an ICAD-DSM parser has been written to automatically generate DSMs. This parser is used to reverse engineer and represent the architecture of sample KBE legacy systems as a Lattix DSM. To capture the DSM, dependencies and problem decomposition are recorded as a hierarchical structure of KBE System, sub-systems, files, parts, sub-parts, design parameters (inputs, attributes etc.), design rules and design procedures (methods, functions etc.). Figure 1 shows our proposed use of different types DSMs within KBE lifecycle processes. Current work is primarily concentrated in the use Component Based DSM [4] and Parameter Based DSM [5]. We have explored the analysis of large-scale KBE systems from the perspective of difficulty to reengineer/migrate these systems to new tools and technologies.

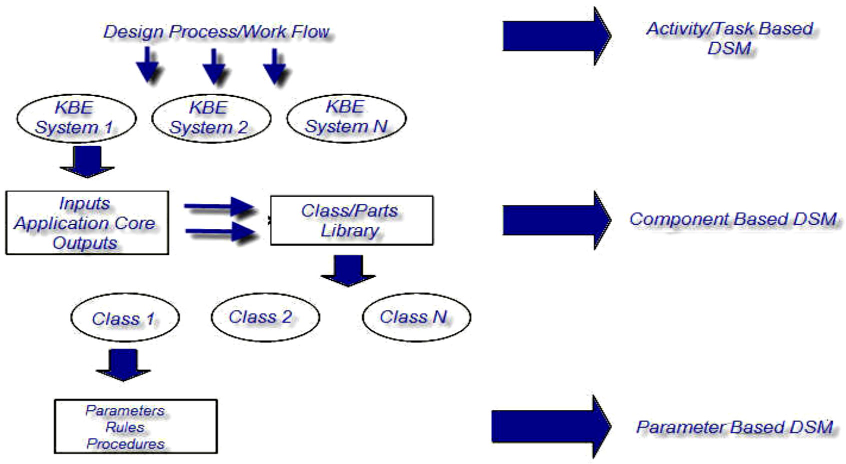


Figure 1: DSMs for Knowledge Based Engineering Software Systems

DSM partitioning has been applied to create a Layered Systems Architecture, which is then used to identify high level and low level sub systems and eventually used to identify external libraries, core logic, utility sub systems as well as top level functional layer of the overall application. Systems with flat structure have been identified and restructured to create sensible hierarchies using clustering. DSM Partitioning and Clustering analysis algorithms allowed for easy, efficient and data-driven planning of architecture refinement and migration.

4 CONCLUSION

In using DSM we were able to get good insights into both Engineering process as well as Software architecture perspectives of KBE systems. Following are some of the advantages of applying DSM to KBE processes:

1) Effective problem/product decomposition and System integration
2) Supports Lean principles for Software Engineering by eliminating waste, by identifying and removing redundant and unused entities (i.e. systems, files, parts etc.)
3) Enables refactoring and efficient recycling of KBE systems
4) Enables modularization and collaboration within and among various KBE Systems
5) Enables knowledge sharing across team members
6) Enables Layering of Knowledge [Figure 3], making a transparent knowledge model.
7) Supports defect analysis and regression testing processes
8) Documents System Architecture evolution process

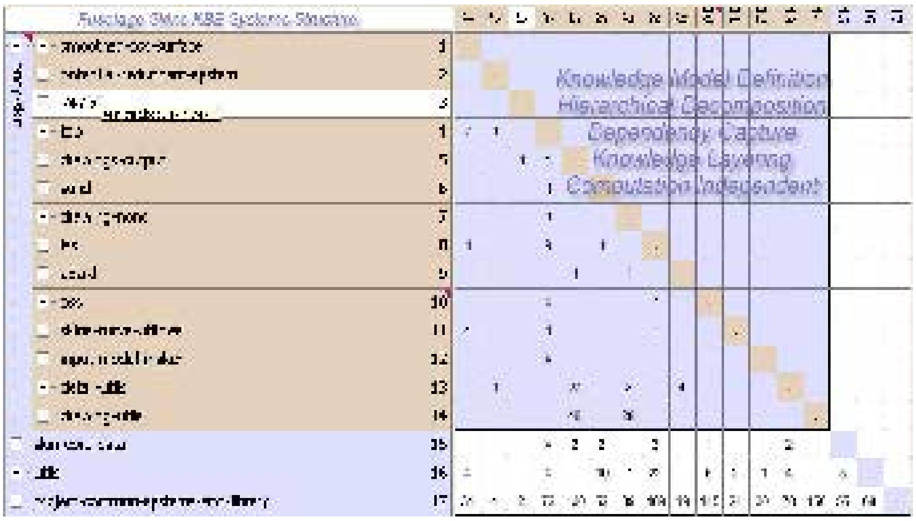


Figure 3: Knowledge Layering using DSM

Re-engineering and migration of legacy KBE systems were the focus areas of this paper. Though DSM provided us a good understanding of top down perspective of a KBE system, it is still a challenging task to understand complexity of implementation details captured by Parameter Based DSM. This still requires a domain expert to manually go through the KBE system to make the legacy architecture transparent. Future work will explore DSM as a forward engineering tool to capture the design requirements and feed a formal design model to downstream KBE system development activity.

REFERENCES

[1] Lattix, Inc., *DSM for Managing Software Architecture: The Lattix Approach*, 2004.
[2] Wagner, M.R., *Understanding the ICAD System,* ICAD Inc, 1990.
[3] Kneebone, S. and Blount, G.N., *A Short Term Strategy for Product Knowledge Sharing and Re-Use. International Conference on Engineering Design, ICED 1997*, pp203-210.
[4] Pimmler, T. U. and S. D. Eppinger, *Integration Analysis of Product Decompositions. Minneapolis, MN. ASME Conference on Design Theory and Methodology*, 1994, pp343-351, http://web.mit.edu/eppinger/www/publications.html
[5] Rask, I. and Sunnersjo, S., *Design Structure Matrices for the Planning of Rule-Based Engineering Systems. Goteborg, Sweden. Proc. European Conference on Integration in Manufacturing,* 1998.

Contact: Sreeram Bhaskara
The Boeing Company
22833 SE Black Nugget Rd
Issaquah, WA, USA 98029
USA
Phone: (001) 425-373-8502 (Work) (001) 425-281-5431 (Mobile)
e-mail: sreeram.bhaskara@boeing.com
http://www.boeing.com

10TH INTERNATIONAL DSM CONFERENCE

Re-engineering Legacy Knowledge Based Systems (KBE) Using DSM

Sreeram Bhaskara

Technische Universität München

Knowledge Based Engineering (KBE)

- Engineering design using advanced software development techniques
- Captures process and product knowledge
- Aides in design automation & design reuse
- Implemented using process centric and/or cad-centric approach
- Uses design objects, design parameters and design rules
- Reduces design time and aides in innovation
- Complexity increases as system evolves over time

Technische Universität München

KBE at Boeing

- Boeing has been one of the pioneers of applying KBE technology
- Significant number of KBE systems developed over a span of two decades
- Large repository of legacy systems in Aircraft Structures (i.e. fuselage design, wing design etc.) and Systems (i.e. Electrical, Hydraulics etc.)
- Business need to migrate, modernize these systems and make the legacy knowledge transparent

Problem Statement

- Large amount of Intellectual capital is embedded inside legacy KBE systems
- Systems become black boxes, as they evolve over long periods of time
- Difficulty re-engineering these systems
- Need to analyze, understand and maintain existing systems
- Propose improved solutions for current systems
- Migrate critical systems to new design tools and technologies
- Difficulty finding experienced resources with domain expertise
- Difficulty training new resources

Proposed Solution

- Explore the application DSM for re-engineering Knowledge Based Engineering Systems
- Generate as-is DSMs for legacy systems
- Reduce system complexity by applying DSM techniques to refactor and re-structure
- Use DSM to reverse engineer and re-architect legacy processes, systems and enterprise architectures for migration and modernization
- Extract domain specific knowledge i.e. design data and rules etc.
- Use DSM representation as a transparent view of legacy system and for generating system documentation

FOLLOW THE FLOW OF INFORMATION!
MANAGE COMPLEX SYSTEMS

Legacy System Analysis

- Understand system functionality and use
- DSM tool used is Lattix LDM
- Generate Preliminary DSM based on System Structure
- Classify dependencies
- Analyze and Re-engineer System Architecture by restructuring DSMs
- Identify and define architectural and design rules and constraints as needed
- Integrate with Project Management, Forward Engineering and Model Based Development activities

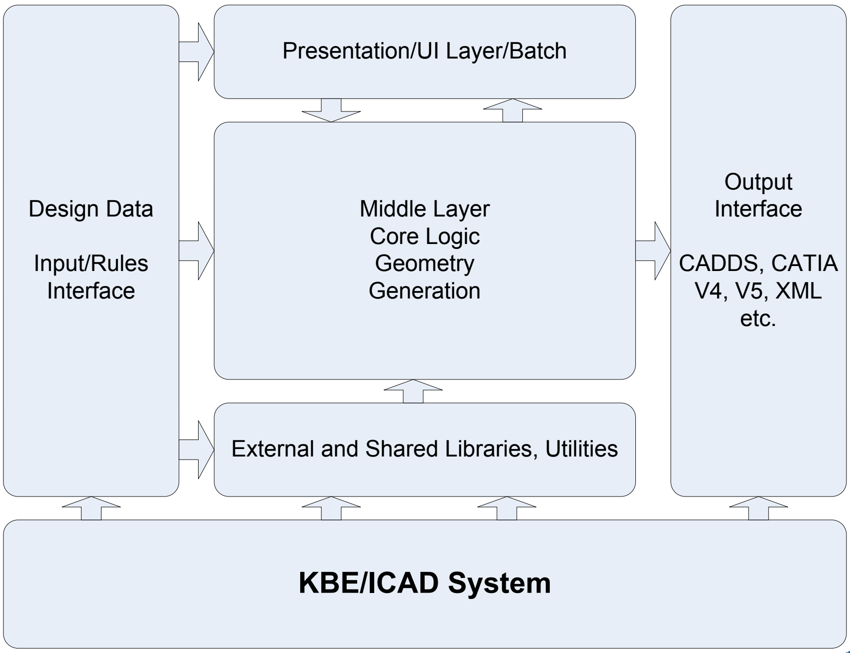
Legacy System Dependencies

- Legacy Applications analyzed are based on ICAD System
- ICAD is a commercial KBE System and has been around for over two decades
- Product model as well as design process can be captured using ICAD
- Component Based and Parameter Based DSMs are generated based on existing System Structure
- Dependencies Captured:
  - System dependencies
  - Design Part (defpart) dependencies
  - Part attributes and method dependencies
  - Other dependencies (functions, macros, etc.)

Dependency Classification

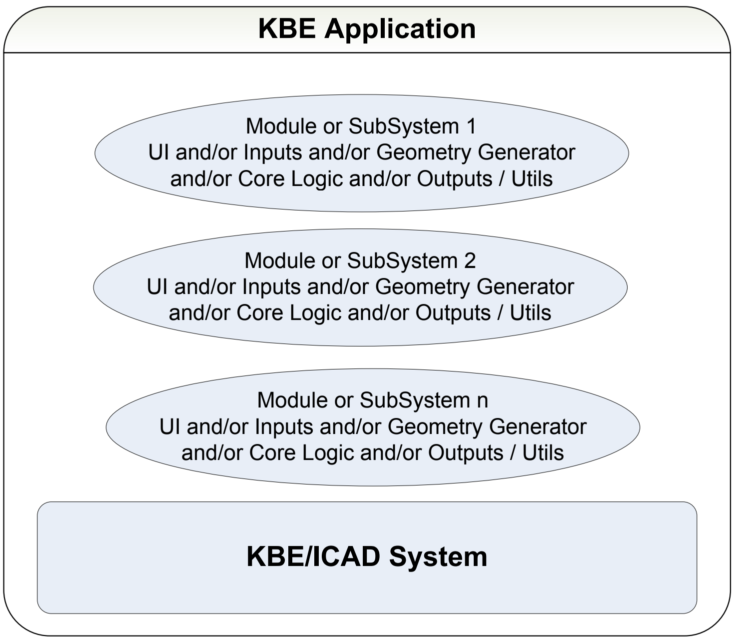
- Legacy System dependencies are classified as follows:
  - Type 1 dependencies
    - ICAD System, Sub-System structure
    - System and Files structure
    - File and DefParts (i.e. parts)
  - Type 2 dependencies
    - Defpart, mixins, sub-parts, attributes and methods structure
  - Type 3 dependencies
    - Attributes, methods, sub-part dependencies
  - Type 4 dependencies
    - Other dependencies (functions, macros, global design variables etc.)

Typical KBE Application Architecture

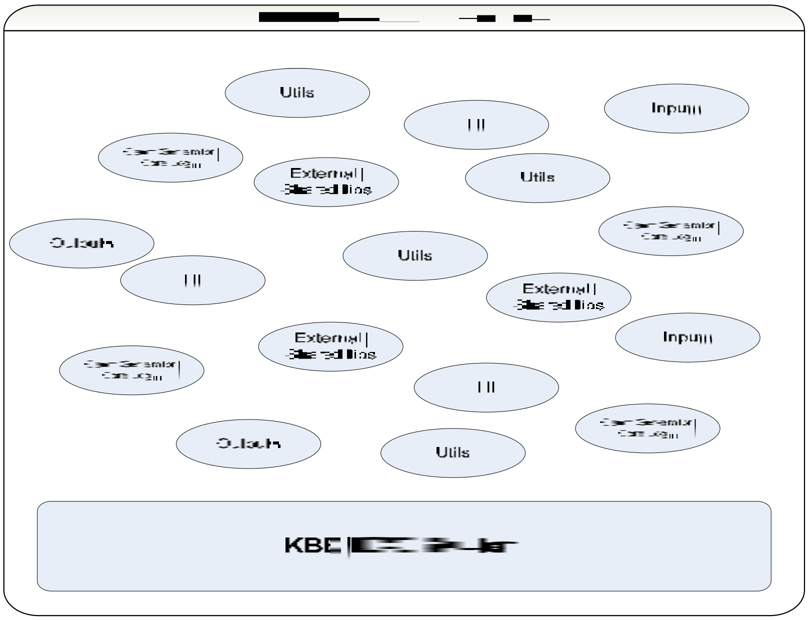


FOLLOW THE FLOW OF INFORMATION!
MANAGE COMPLEX SYSTEMS

ICAD KBE System Structure

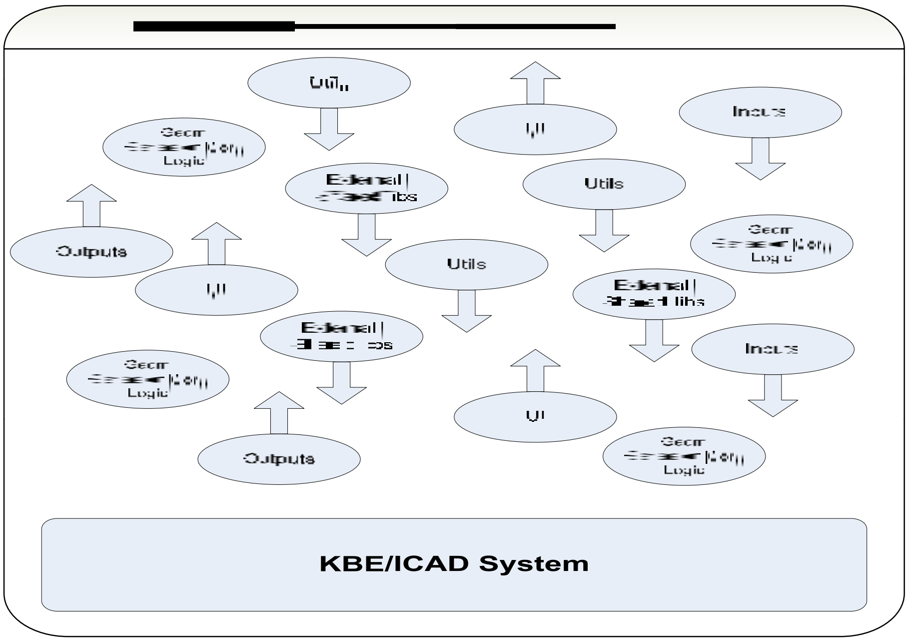


Typical KBE Architecture Components – Minimal Architecture

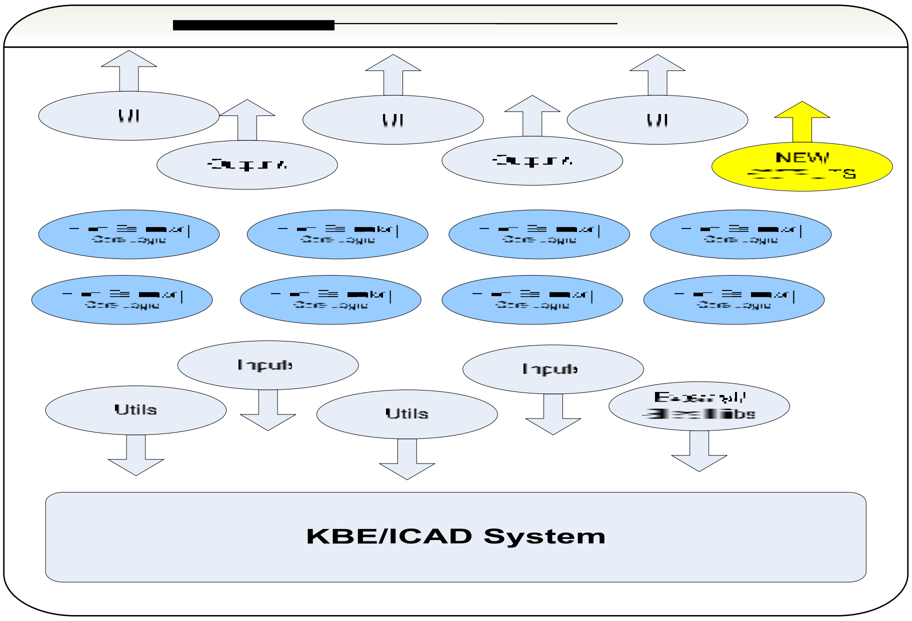


FOLLOW THE FLOW OF INFORMATION!
MANAGE COMPLEX SYSTEMS

Layered Architecture: Systems With Orientation



Layered Systems Architecture Applying DSM Techniques



Analyze and Re-Engineer Architecture

- Identify high level and low level sub systems (***Type 1 dependencies***)
  - Identify sub systems that are external libraries
  - Use DSM partitioning to create a Layered Systems Architecture

- Eliminate and streamline systems and architecture
  - Type 1 dependencies
    - Remove unused sub systems, files etc.
  - Type 2 dependencies
    - Remove unused defparts, mixins etc.
  - Type 3 dependencies
    - Remove unused attributes, methods, sub parts etc.
  - Type 4 dependencies
    - Remove unused functions, macros etc.

- Identify/reorganize systems with flat structure (***Type 1 dependencies***)
  - Identify sub systems with many children (i.e. files, defparts)
  - Re-organize and create new sub systems as needed
  - Create a sensible hierarchy in sub systems
  - Capture dependency relationship among new sub systems

Analyze and Re-Engineer Architecture (cont'd)

- Identify source files with multiple parts (***Type 1 dependencies***)
  - Re-organize and move multiple defparts into new source files
- Identify key parts of the system (***Type 2 dependencies***)
  - Sort defparts on usage weight
  - Identify top level defparts with functional inputs
  - Identify defparts that interface with external libraries
- Identify circularities in defpart, attribute and method dependencies (***Type 2 & 3 dependencies***)
  - Highlight potential circularity errors (circular references) in the design code

Architecture Management

- Re-Engineering
  - Modify existing legacy system structure based on refined architecture map
  - Explore methods to automate this process
- Migration
  - Analyze refined architecture maps to migrate and output to new tools
- Forward Engineering
  - For new applications, define and generate new architectures using Lattix DSM tool and feed the DSM as a decomposition structure to KBE development environments

Re-Engineered Aircraft Fuselage KBE System after Lattix DSM Analysis

Fuselage Skins KBE Systems Structure

body-skin

test-dist.body-skin.747-x

Fuselage Skins KBE Systems Structure	#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
[illegible]	1	.																
potential-redundant-system	2		.															
747-x	3			.														
top	4	2	1		.													
drawings-output	5			1	4	.												
solid	6				1		.											
drawing-nonc	7				1			.										
iss	8	1			9		1		.									
apard	9					1		1		.								
oss	10				4				1		.							
skins-curve-utilities	11	4			1				1			.						
input-model-maker	12				4								.					
detail-utils	13		1			27		2		4				.				
drawing-utils	14					48		28							.			
skin-core-data	15				4	2	2		3		1			2		.		
utils	16	4			4		10	1	22		6	2	1	4		3	.	
project-common-systems-and-library	17	34	4	2	72	130	52	39	189	19	145	24	30	70	150	55	68	.

Knowledge Model Definition
Hierarchical Decomposition
Dependency Capture
Knowledge Layering
Computation Independent

Summary

- DSM has been used to Understand and Restructure legacy systems
- Lattix DSM Layered Architecture approach has been used to separate out knowledge systems into different layers
- Based on the business need, identified systems for migration
- DSM enabled refactoring, re-engineering and efficient recycling of KBE systems
- DSM enabled knowledge sharing across team members and generation of system documentation
- DSM enabled modularization and collaboration within and among various KBE Systems
- DSM Enabled KBE system decomposition and integration at various levels of abstraction
- Supported lean principles of software engineering design