

Concept Maps: Front End or Replacement for SysML/MBSE?

INCOSE Cleveland Chapter Meeting
February 2016

Brian Moon



Carl Dister





Worker

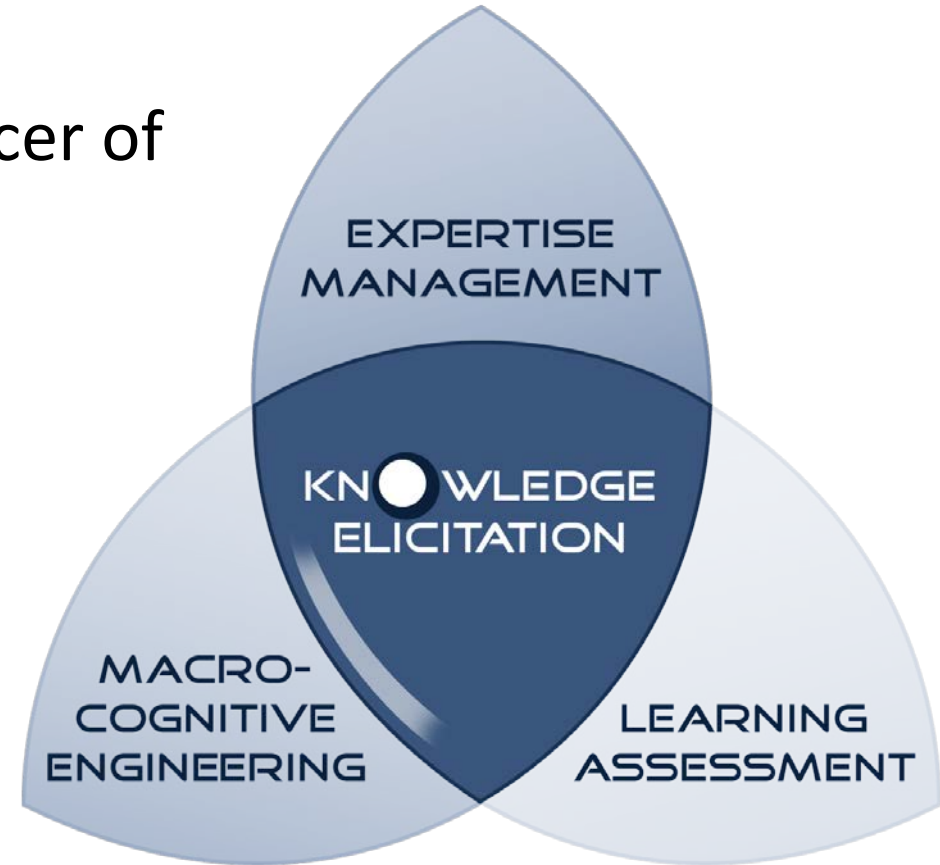
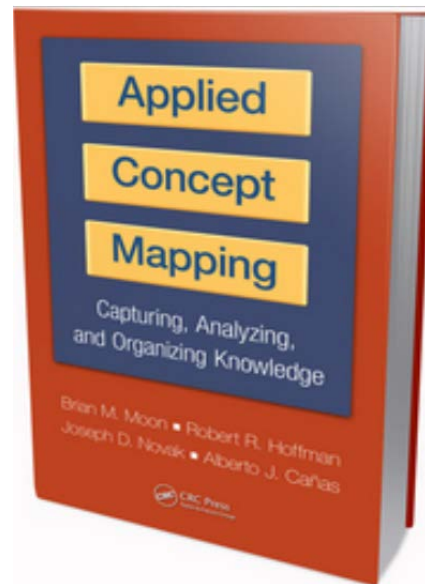


Relaxed
Worker

Speaker Bio

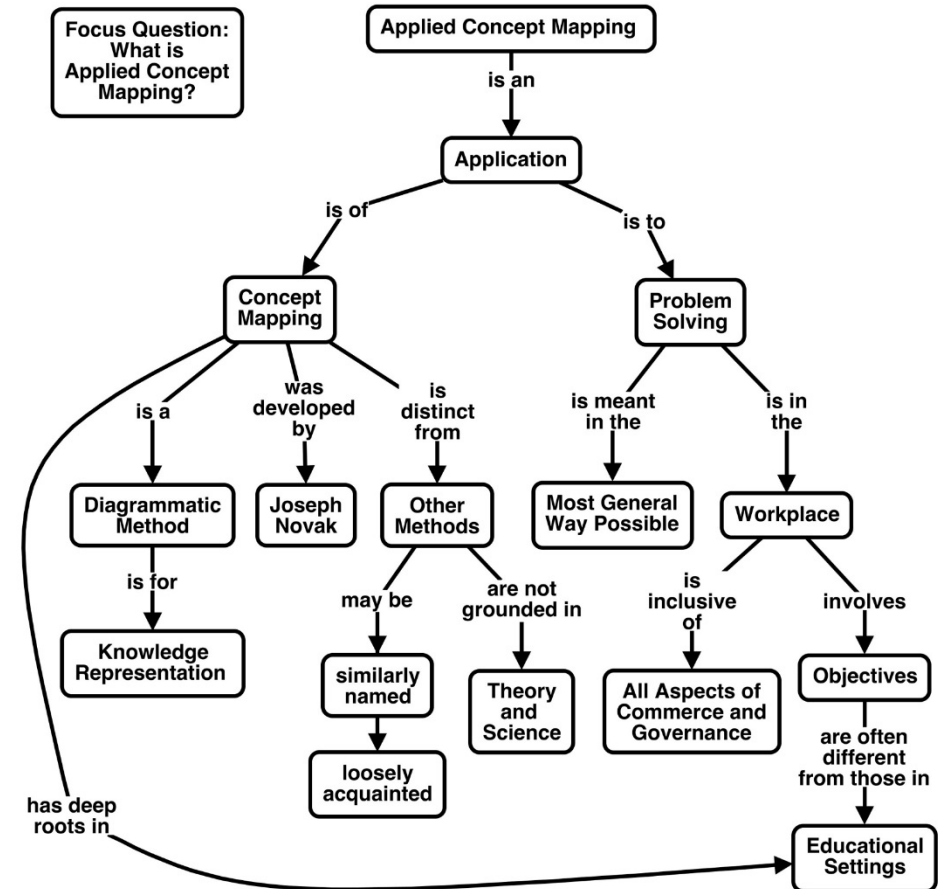
Co-Founder and Chief Technology Officer of
Perigean Technologies LLC

Co-Editor

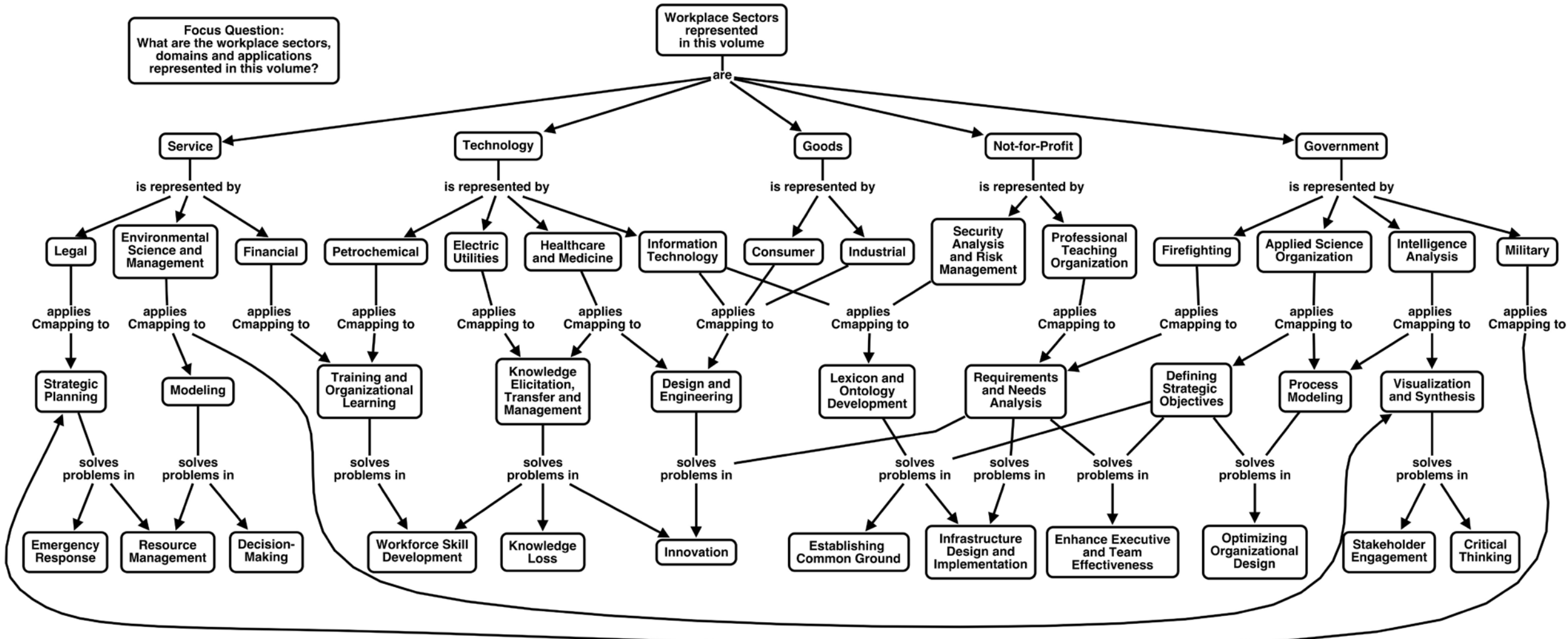


Concept Maps

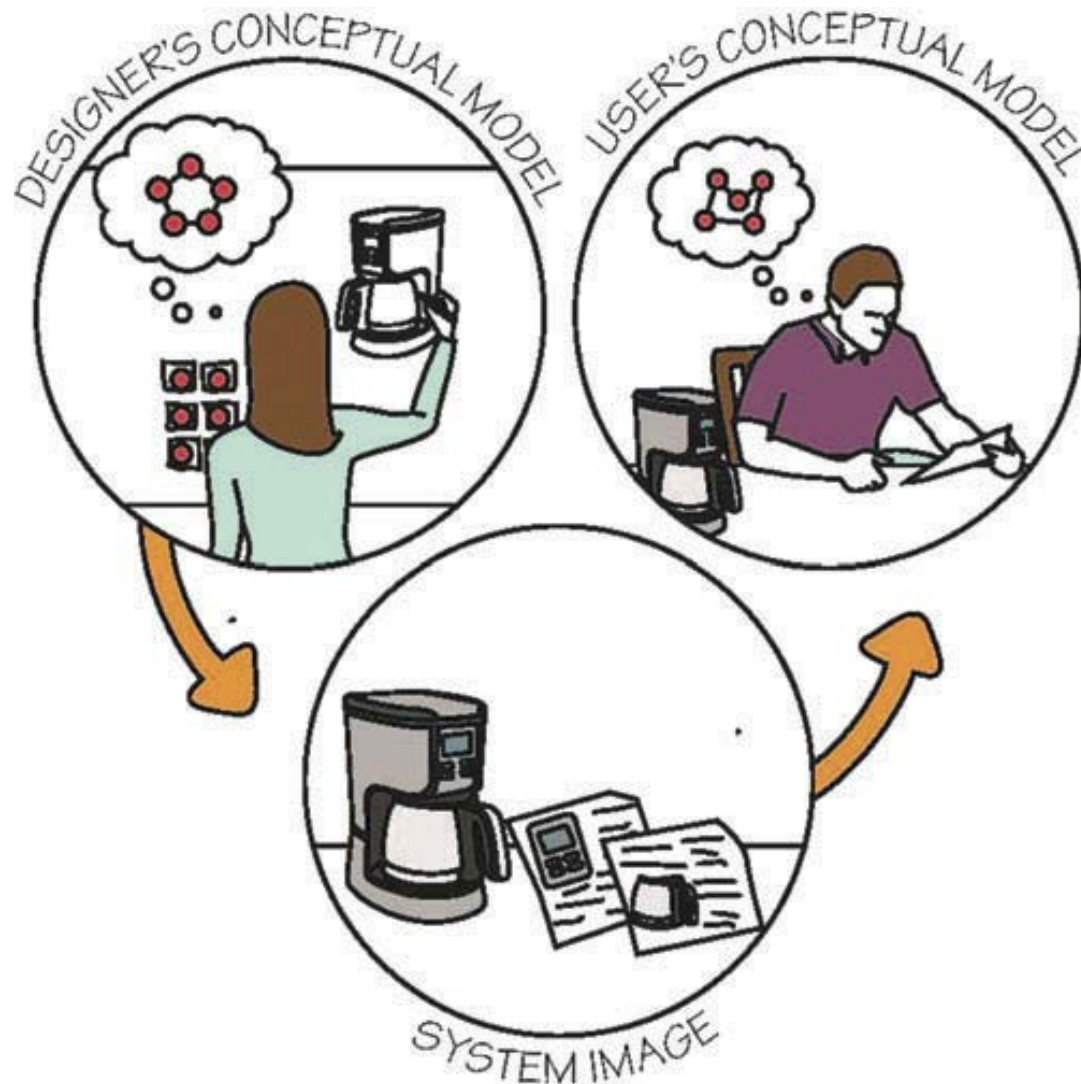
- A diagrammatic knowledge representation technique
- Distinguishing features
 - Propositions (Concept – Link – Concept)
 - Concept = perceived regularity in events and objects
 - Links = expressed relationships
 - Propositional coherence = all propositions can be read independent of the concept map
 - Contextual meaning = all propositions are understood in the context of the concept map
 - Shape (Semi-hierarchy)
 - Allows for cross-links
 - Proposition & shape interaction



Concept Maps



Concept Mapping in Software Development



The user of any system, of any artifact, necessarily develops a mental model of how that system works and of how he or she should interact with it to carry out some task

The designer/developer(s) of an artifact has some mental model of how the system/product would work and that that mental model, the designer/developer model, is de facto instantiated as the system/product itself.

Norman (2014)

Concept Mapping in Software Development

- *Conceptual Mapping as a First Step in Data Modeling*

- Hector Gómez-Gauchía and Ron McFadyen
- (Chapter 14 of *Applied Concept Mapping*)



- *Using Concept Maps in Product Development: Preparing to Redesign java.sun.com*

- Hugh Dubberly
 1. Set Goals
 2. Identify Terms
 3. Prioritize Terms
 4. Define Terms
 5. Organize Terms
 6. Test "Armature" (primary sentence or two)
 7. Add Terms
 8. Review and Revise
 9. Subdivide Large Maps
 10. Refine the Typography
 11. Check Again

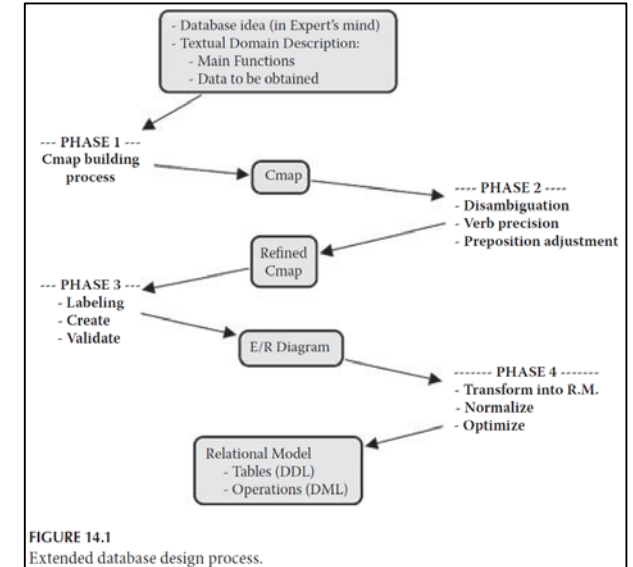
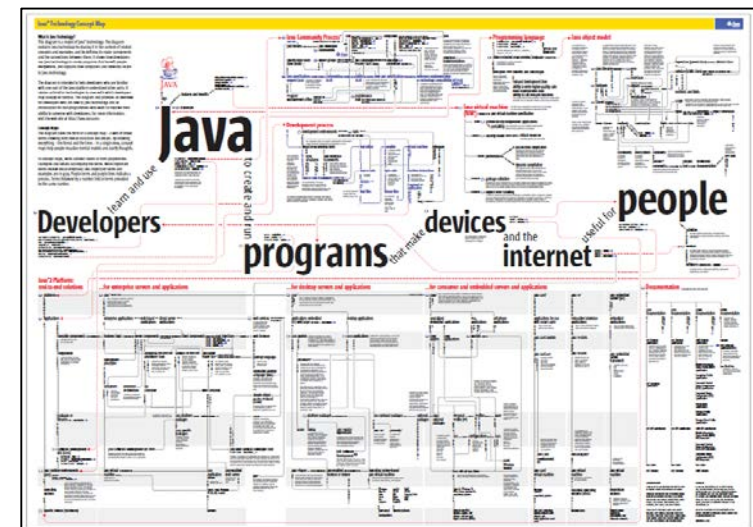


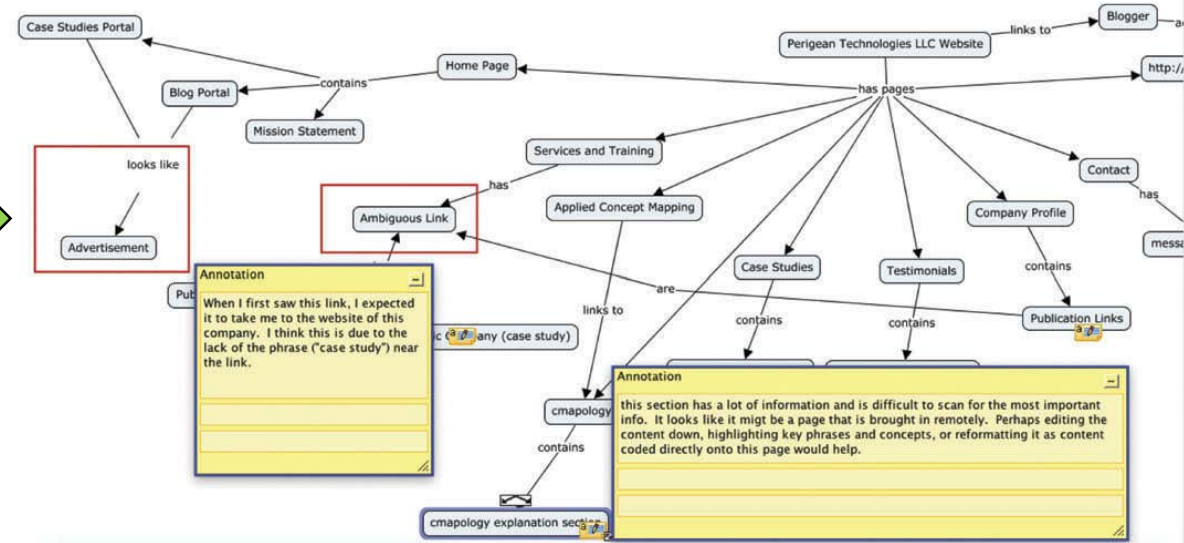
FIGURE 14.1
Extended database design process.



Concept Mapping in Software Development

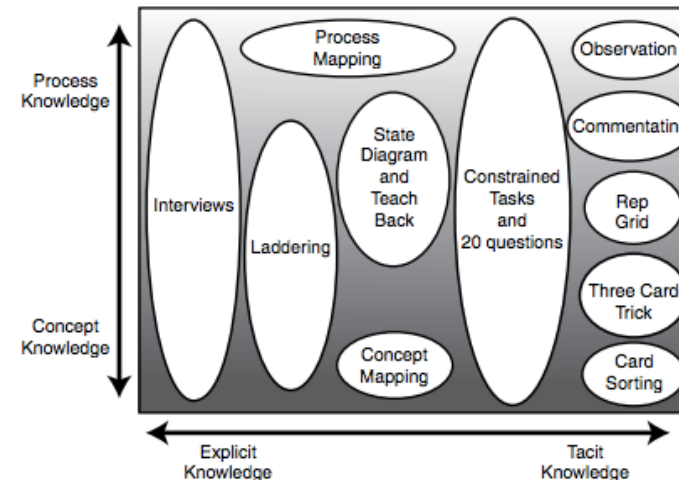
- *Concept Mapping Usability Evaluation*

- Bias, Moon and Hoffman



- *UML for Developing Knowledge Management Systems*

- Anthony Rhem



Concept Mapping to SysML/MBSE

The categories of linking phrases in Concept Mapping overlap nicely with the various SysML diagram types:

Causal → State Diagram

Classificational → Use Case Diagram

Nominal → Package Diagram

Property → Block Definition Diagram

Explanatory → Requirements Diagram

Procedure/Method → Sequence Diagram

Event → Activity Diagram (State Diagram)

Uncertainty → Parametric Diagram

Compare to Aristotle's Universals

1. Substance
2. Quantity
3. Quality
4. Relation
5. Place
6. Time
7. Position
8. State
9. Action
10. Affection

Use Case Diagram

"involves;is a type of;types of which are;includes;categories;examples;is a kind of"

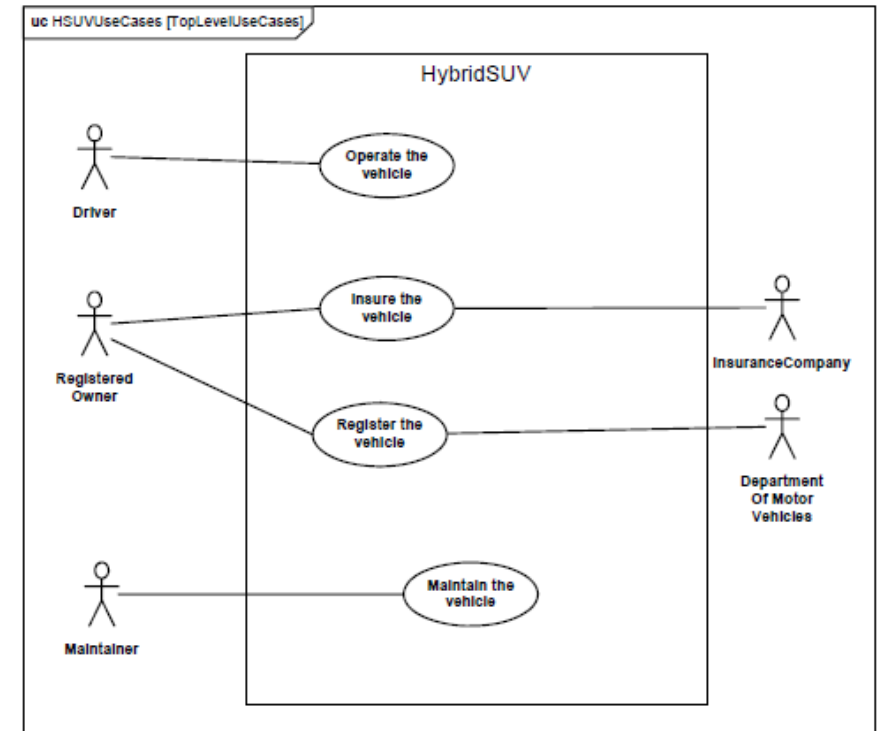
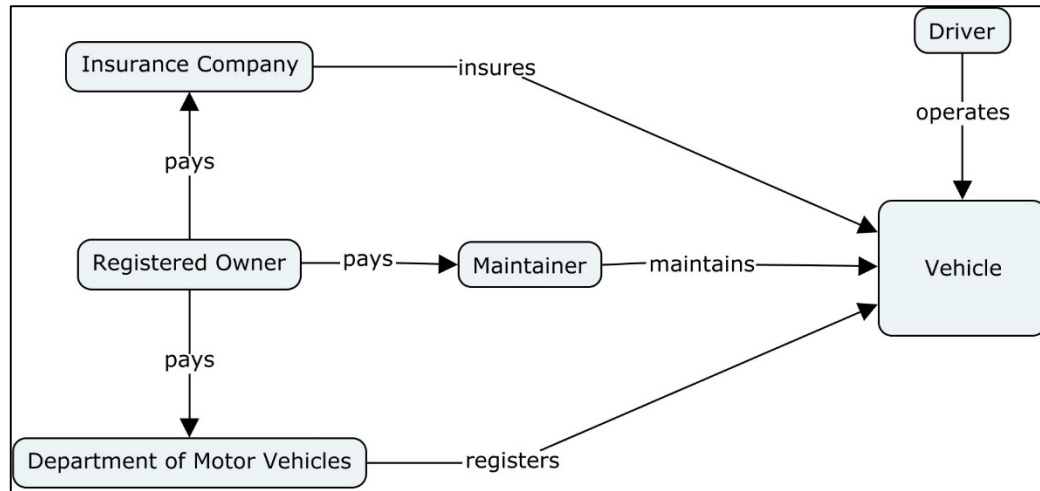


Figure C.5 - Establishing Top Level Use Cases for the Hybrid SUV (Use Case Diagram)

Sequence Diagram

"is followed by;produces;done by;is a way to do;results in;demands;prompts"

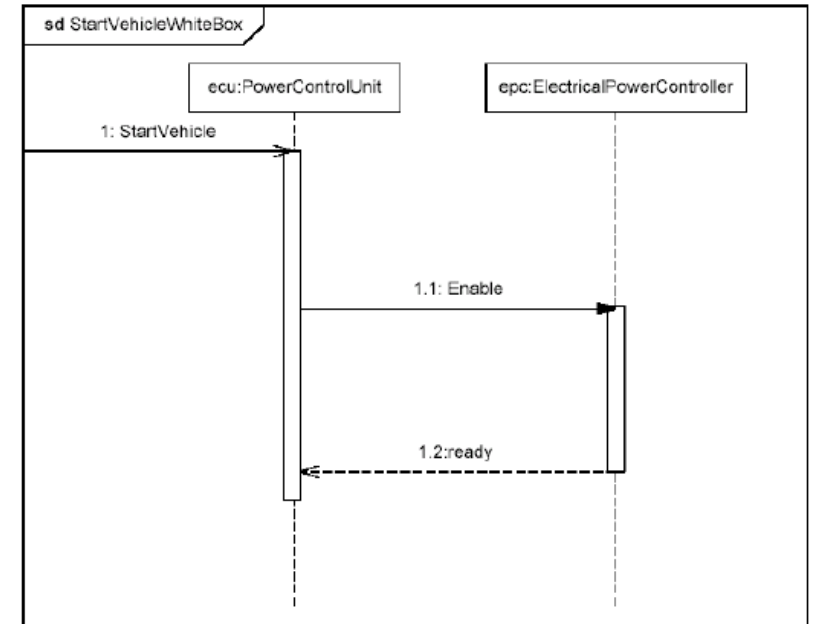
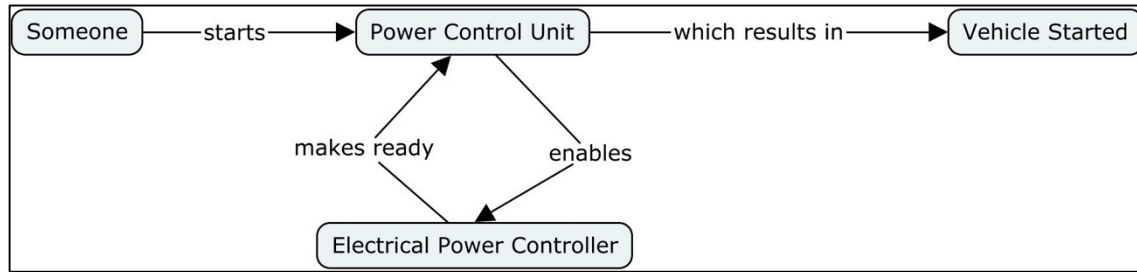


Figure C.10 - White Box Interaction for "StartVehicle" (Sequence Diagram)

State Machine Diagram

"becomes; evolves to"

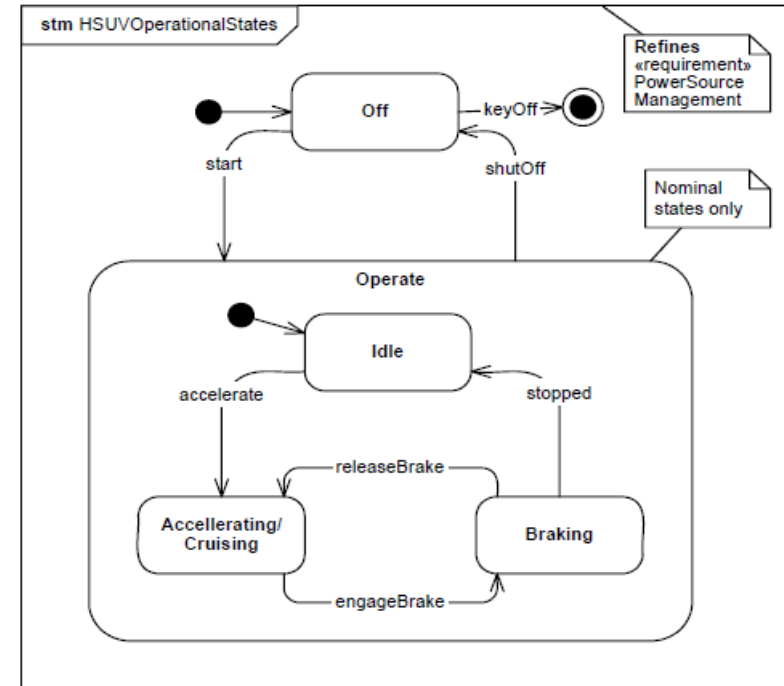
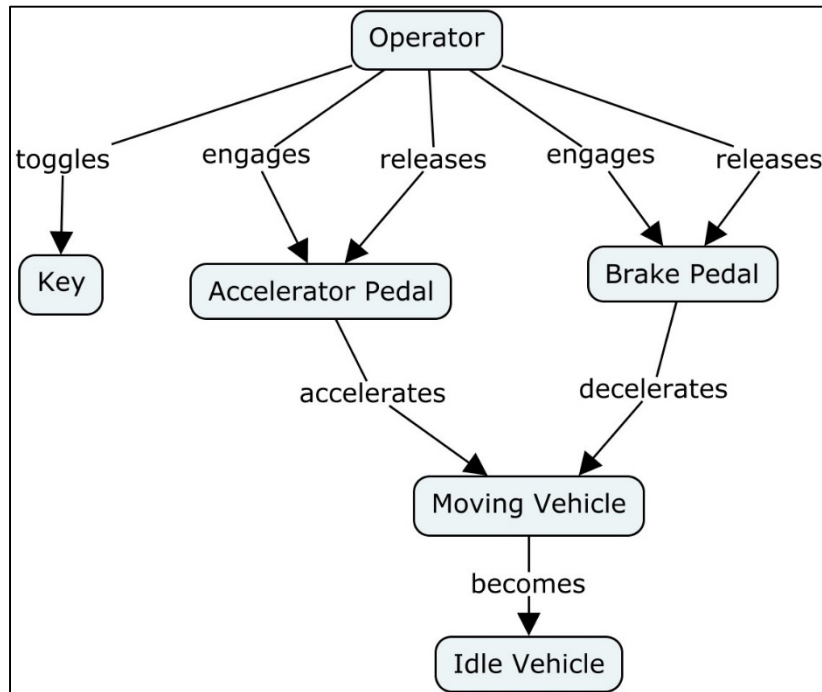


Figure C.8 - Finite State Machine Associated with "Drive the Vehicle" (State Machine Diagram)

Activity Diagram

"becomes; evolves to"

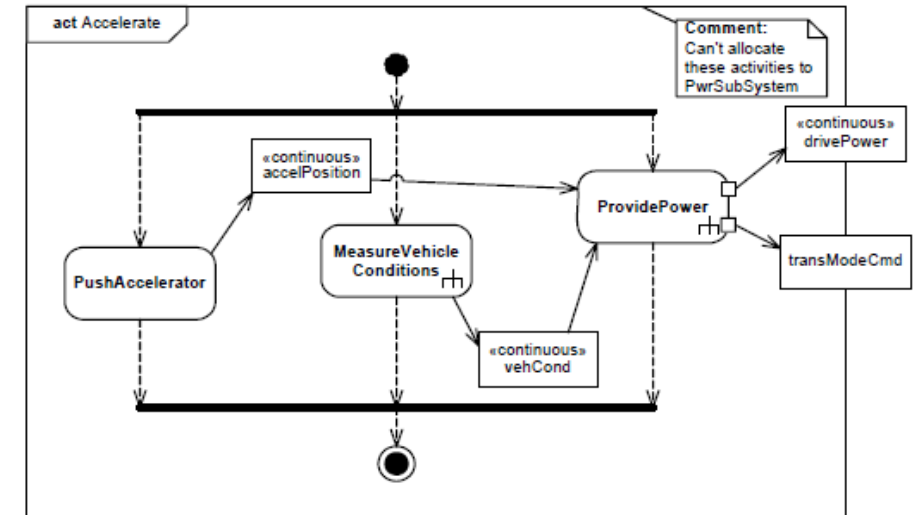
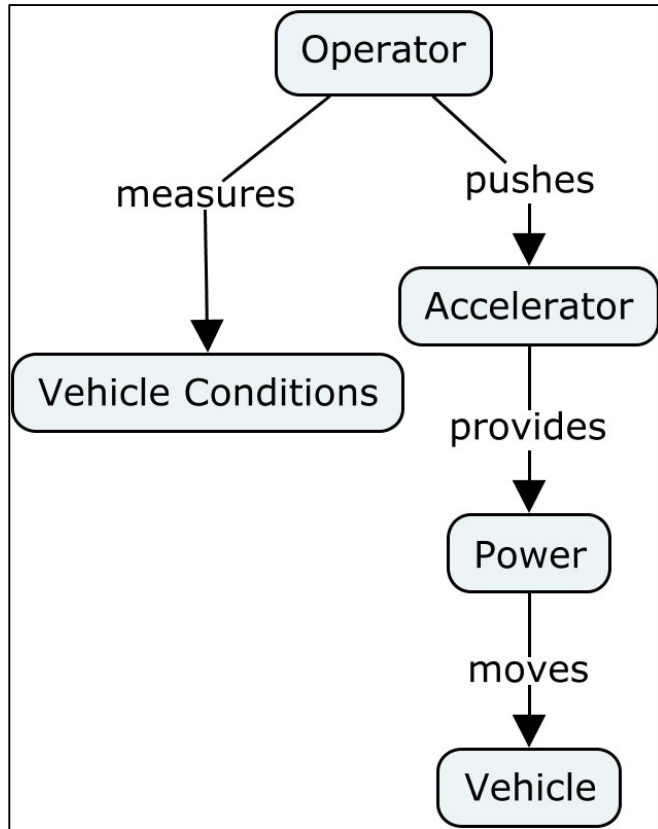


Figure C.33 - Behavior Model for "Accelerate" Function (Activity Diagram)

Requirements Diagram

"explains; reasons for; requires"

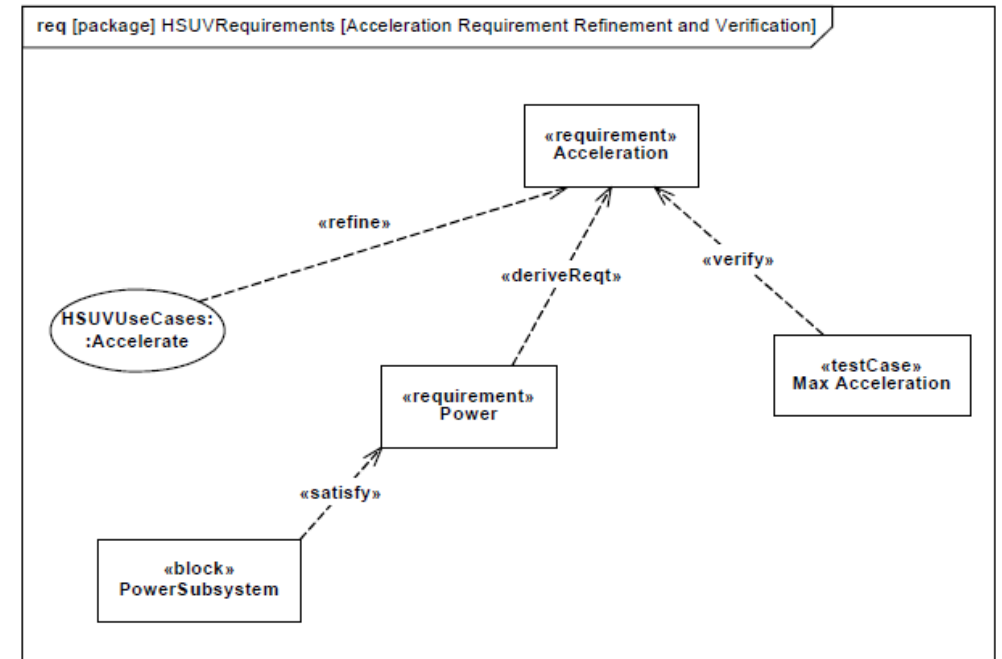
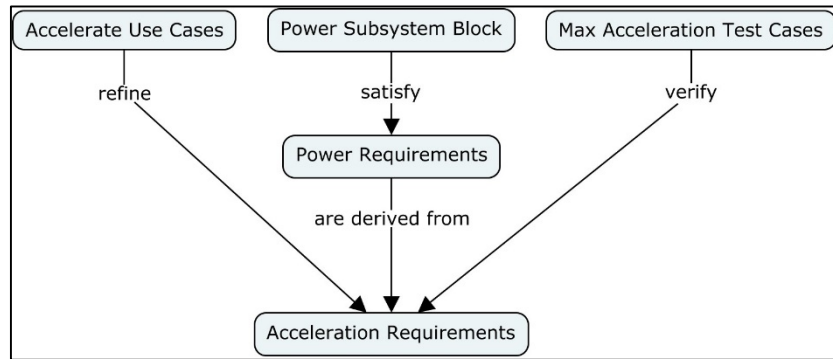


Figure C.13 - Acceleration Requirement Relationships (Requirements Diagram)

Block Definition Diagram

"consists of;has;has feature;has defining feature;has property;have"

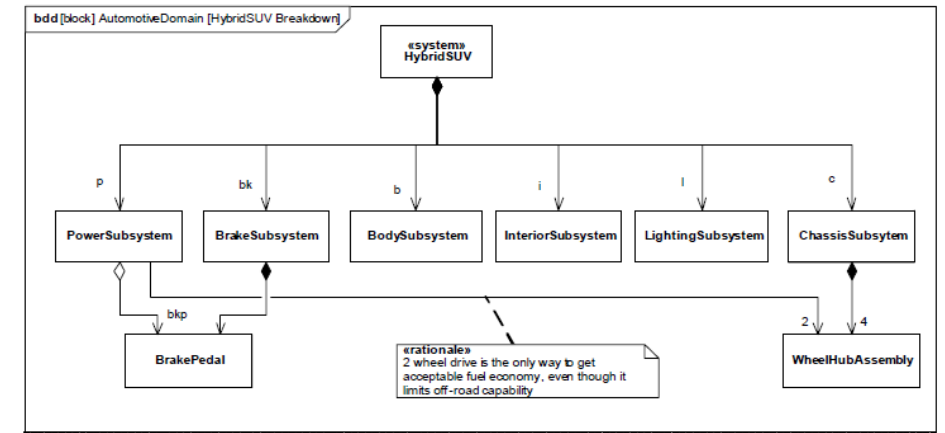
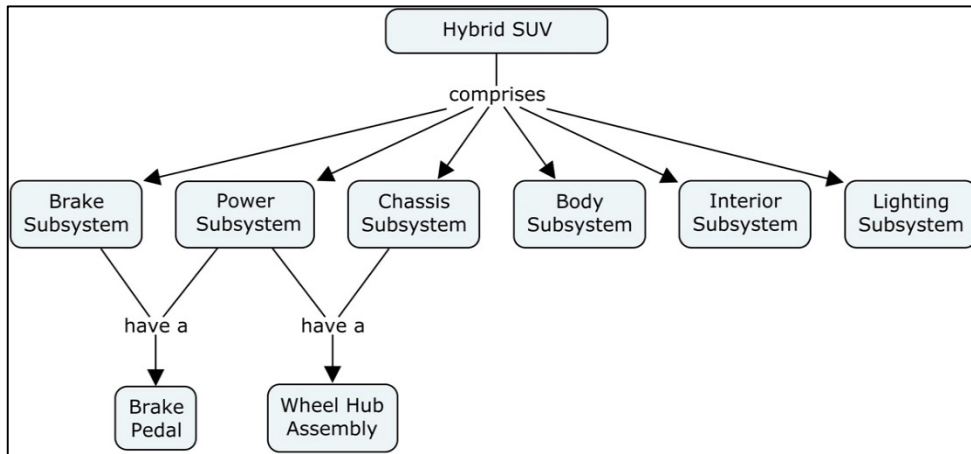


Figure C.16 - Defining Structure of the Hybrid SUV System (Block Definition Diagram)

Parametric Diagram

"always; may or may not; sometimes is; is more likely; can be; often is; usually is; rarely is"

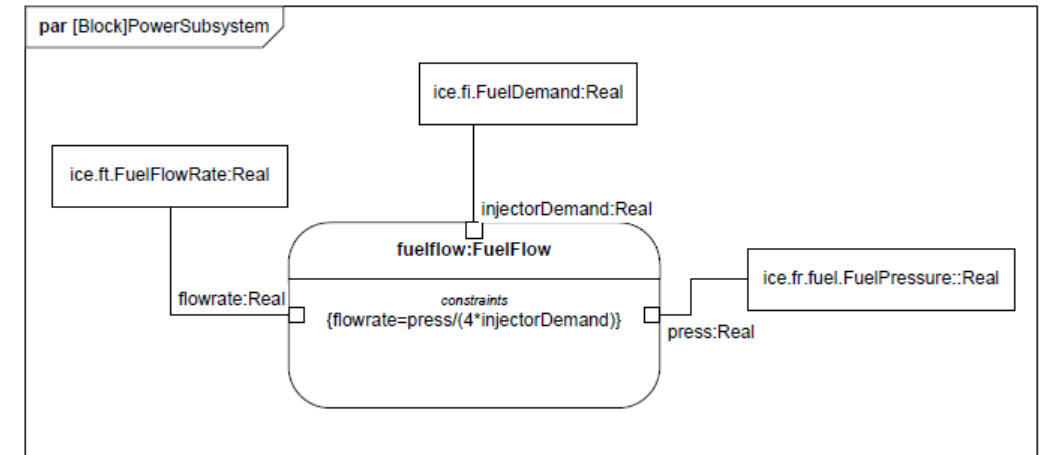
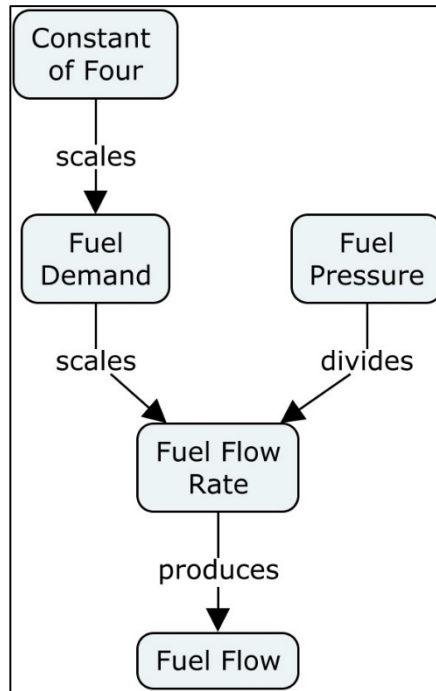


Figure C.24 - Defining Fuel Flow Constraints (Parametric Diagram)

Package Diagram

"which are/is; is a; is; i.e.; for example; e.g.; referred to as; such as"

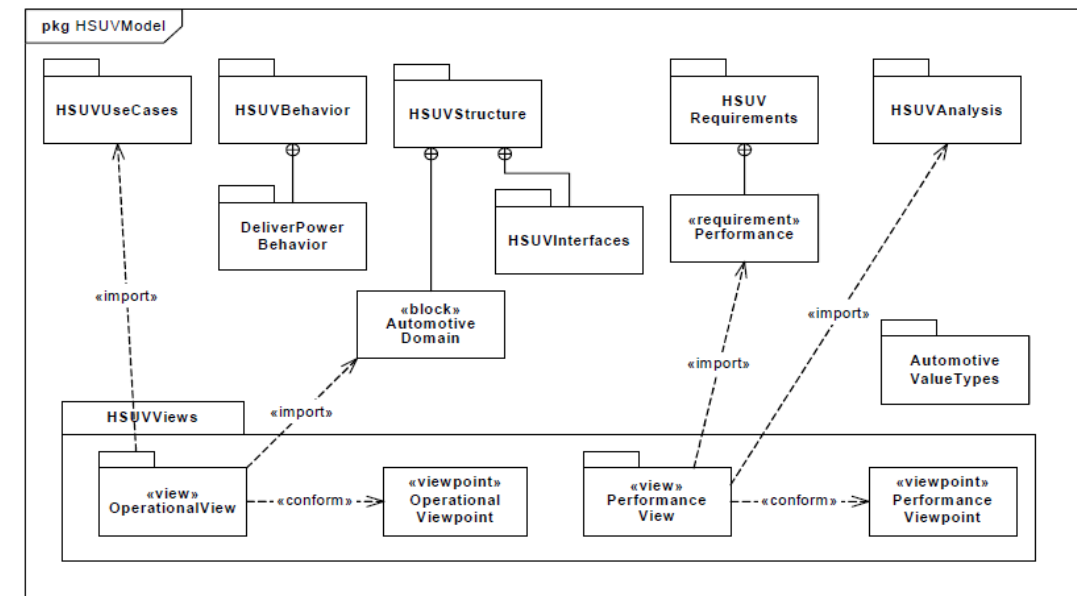
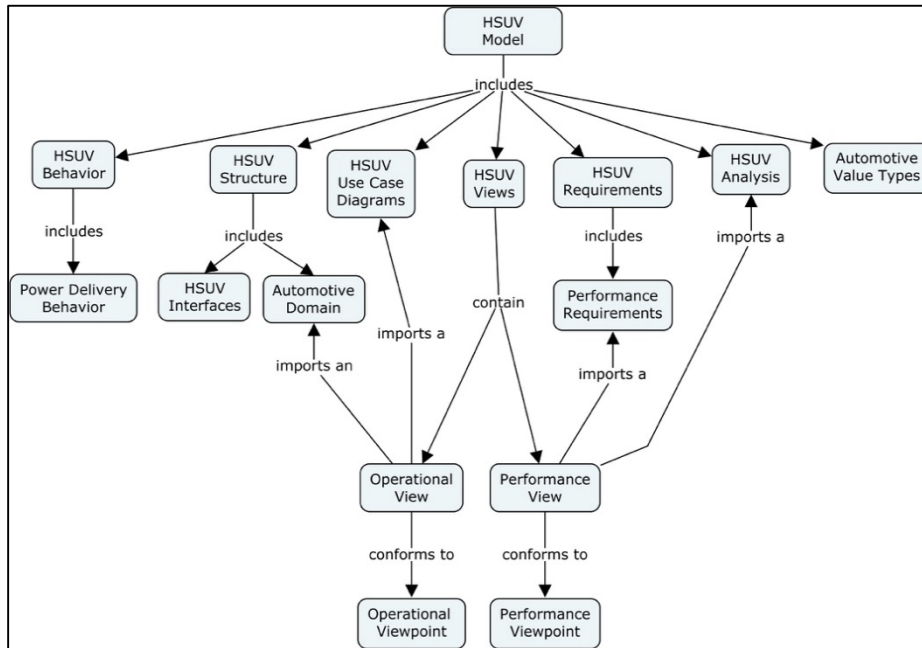


Figure C.3 - Establishing Structure of the User Model using Packages and Views (Package Diagram)

When to Apply Concept Mapping during the Systems Lifecycle

- **NEW SYSTEMS AT KICKOFF:** Consider starting project using Concept Mapping (e.g. Stakeholder Elicitation of Requirements)
- **COMPLETE LIFECYCLE:** Low Budget and have time to manual convert Concept Maps database to requirements documents.
- **SYSTEM MODIFICATIONS:** For modifications to existing systems, consider Concept Mapping to capture expertise on current system, then modify
- **SPECIAL CASES:** Consider strongly when migration to SysML is really required:
 - Special automatic code generation tools being used
 - High Complexity requiring symbolic properties difficult to express in words

Words of Potential

- Extremely flexible
- No special vocabulary, symbology required
- Quick uptake

Words of Caution

- May not integrate readily with other tools
- No inherent symbology
- Limited drawing capabilities
- Big difference between making maps and making 'good' maps

Concept Mapping Skills Development

- Concept Map features (propositional thinking)
- Concept Mapping software capabilities (merging, exporting, auto-arranging)
- Quickly drawing while interviewing stakeholders
- Reorganizing maps
- Interviewing skills

CmapTools



- Free!
- Florida Institute of Human and Machine Cognition
- cmap.ihmc.us
- Client – desktop, laptop; platform independent
 - Create, edit
- Server
 - Share, publish
- iPad



INSTITUTE FOR HUMAN & MACHINE COGNITION

Thank you!

Brian Moon

brian@perigeantechnologies.com

540.429.8126



Carl Dister

Carl.Dister@rfirst.org

216-503-0675

