CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Information Modeling for Continuous Revolution – Modeling Knowledge and Coordination in Large-Scale Agile Development

Jennifer Horkoff + many colleagues

jenho@chalmers.se
jennifer.Horkoff@gu.se

MoDRE Keynote, RE’24

Inspiration

Information Systems 23(3-4), June 1998.

Information Modeling in the Time of the Revolution[1]

John Mylopoulos[2]
University of Toronto

Abstract

Information modeling is concerned with the construction of computer-based symbol structures which capture the meaning of information and organize it in ways that make it understandable and useful to people. Given that information is becoming an ubiquitous, abundant and precious resource, its modeling is serving as a core technology for information systems engineering.

We present a brief history of information modeling techniques in Computer Science and briefly survey such techniques developed within Knowledge Representation (Artificial Intelligence), Data Modeling, (Databases), and Requirements Analysis (Software Engineering and Information Systems). We then offer a characterization of information modeling techniques which classifies them according to their *ontologies*, i.e., the type of application for which they are intended, the set of *abstraction mechanisms* (or, *structuring principles*) they support, as well as the *tools* they provide for building, analyzing, and managing application models. The final component of the paper uses the proposed characterization to assess particular information modeling techniques and draw conclusions about the advances that have been achieved in the field.

Keywords: Conceptual model, semantic data model, requirements model, knowledge representation language, ontology, abstraction mechanism, classification, generalization, aggregation, contextualization, materialization, normalization, parameterization, semantic network.

1. Introduction

> *"...The entity-relationship model adopts ... the natural view that the real world consists of entities and relationships... (The entity-relationship model) incorporates some of the important semantic information about the real world..."*
>
> Peter Chen [43]

We live through the Age of the Information Revolution. Thanks to advances in telecommunications, computer hardware and software, we are flooded with ever-growing amounts of information. The tremendous impact of the revolution to individuals and organizations alike is a daily news topic. One important

[1] This paper is based on a keynote address presented at the Ninth Conference on Advanced Information Systems Engineering (CAiSE'97) in Barcelona, Catalunya on June 17, 1997; an earlier version of the paper, titled "Characterizing Information Modeling Techniques for Information Systems Engineering", is included in Bernus, P., Mertins, K., and Schmidt, G., (eds.) *Handbook on Architectures of Information Systems*, Springer-Verlag, 1998 (to appear).

[2] Author's address: Department of Computer Science, University of Toronto, 6 King's College Road, Toronto, Canada M5S 3H5; voice: 416-978-5180, fax: 416-978-1455, email: jm@cs.toronto.edu.

Information Modeling

"Information modeling is concerned with the construction of computer-based symbol structures which capture the meaning of information and organize it in ways that make it understandable and useful to people." Mylopoulos, 1998

Revolution

In this case, the information revolution.

Information Modeling for Continuous Revolution – Modeling Knowledge and Coordination in Large-Scale Agile Development

Information Model – Same

Revolution – Agile revolution and fall (de-(re)volution?)

Continuous – Does the revolution ever stop?

Research Context

Chalmers Software Center Project #27 Engineering Knowledge Flows in Large-Scale Agile System Development

- Formerly: RE for Large-Scale Agile System Development
- Current members: Eric Knauss, Jennifer Horkoff, Hans-Martin Heyn
- Past members/collaborators: Rebekka Wohlrab, Rashidah Kasauli, Salome Maro, Jan-Philipp Steghöfer, Jörg Holtmann, Grischa Liebel, Francisco Gomes
- Many master's and bachelor's students

FAMER - FAMER - Facilitating Multi-Party Engineering of Requirements

- September 2023 - August 2026, Vinnova, https://www.vinnova.se/en/p/famer---facilitating-multi-party-engineering-of-requirements/
- University of Gothenburg, Kognic, RISE, Volvo Cars, and Zenseact

Software Center Project Background (Kasauli)

ELSEVIER
Journal of Systems and Software
Volume 172, February 2021, 110851

Requirements engineering challenges and practices in large-scale agile system development

Rashidah Kasauli [a], Eric Knauss [a], Jennifer Horkoff [a], Grischa Liebel [b], Francisco Gomes de Oliveira Neto [a]

Show more

Add to Mendeley Share Cite

https://doi.org/10.1016/j.jss.2020.110851 Get rights and content

Highlights

- Large-scale agile systems engineering implies requirements engineering challenges (RE).
- Challenges are not sufficiently covered by scaled-agile frameworks or traditional RE.
- Challenges apply during transition to agile as well as to full agile system development.
- A comprehensive overview allows to identify holistic solutions.

(C1) Build and maintain shared understanding of customer value
a) Bridge gap to customer
   i. *Make team understand customer value*
   ii. *Unable to express value in user stories*
   iii. *Feedback and clarification*
b) Building long-lasting customer knowledge

(C2) Support Change and Evolution
a) Managing experimental requirements
b) Synchronization of development
c) Avoid re-specifying, encourage re-use
d) Updating requirements

(C3) Build and maintain shared understanding about system
a) Documentation to complement tests and stories
b) System vs. component thinking
c) Creating and maintaining traces
d) Learning and long-term knowledge
e) Backward compatibility

(C4) Representation of reqts knowledge
a) Manage levels vs. decomposition
b) Quality reqts as thresholds
c) Tooling not fit for purpose
d) Accommodate different representations
e) Consistent reqts quality

(C5) Process aspects
a) Prioritization of distributed functionality
b) Manage completeness
c) Consistent requirements processes
d) Quality vs. time-to-market

(C6) Organizational aspects
a) Bridge plan-driven and agile
b) Plan V&V based on reqts
c) Time for invention and planning
d) Impact on infrastructure

Fig. 4. Challenging Areas of RE for Large-Scale Agile System Development.

All of our case companies have agile software development teams that operate within the context of a larger system engineering process, …
“It feels like agile islands in a waterfall.” — FG 2

Software Center Project Background (Wohlrab)

The Journal of Systems and Software 162 (2020) 110516

Contents lists available at ScienceDirect

ELSEVIER

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Why and how to balance alignment and diversity of requirements engineering practices in automotive

Rebekka Wohlrab[a,b,*], Eric Knauss[a], Patrizio Pelliccione[a,c]

[a] Chalmers | University of Gothenburg, Gothenburg, Sweden
[b] Systemite AB, Gothenburg, Sweden
[c] University of L'Aquila, L'Aquila, Italy

ARTICLE INFO

Article history:
Received 1 July 2019
Revised 4 November 2019
Accepted 23 December 2019
Available online 26 December 2019

Keywords:
Requirements information models
Aligning software engineering practices
Automotive software engineering
Large-scale software development
Mixed methods research

ABSTRACT

In large-scale automotive companies, various requirements engineering (RE) practices are used across teams. RE practices manifest in Requirements Information Models (RIM) that define what concepts and information should be captured for requirements. Collaboration of practitioners from different parts of an organization is required to define a suitable RIM that balances support for diverse practices in individual teams with the alignment needed for a shared view and team support on system level. There exists no guidance for this challenging task. This paper presents a mixed methods study to examine the role of RIMs in balancing alignment and diversity of RE practices in four automotive companies. Our analysis is based on data from systems engineering tools, 11 semi-structured interviews, and a survey to validate findings and suggestions. We found that balancing alignment and diversity of RE practices is important to consider when defining RIMs. We further investigated enablers for this balance and actions that practitioners take to achieve it. From these factors, we derived and evaluated recommendations for managing RIMs in practice that take into account the lifecycle of requirements and allow for diverse practices across sub-disciplines in early development, while enforcing alignment of requirements that are close to release.

Received: 30 September 2018 | Revised: 14 January 2019 | Accepted: 17 February 2019
DOI: 10.1002/smr.2166

WILEY Journal of Software: Evolution and Process

SPECIAL ISSUE PAPER

Boundary objects and their use in agile systems engineering

Rebekka Wohlrab[1,2] | Patrizio Pelliccione[1,3] | Eric Knauss[1] | Mats Larsson[2]

[1]Department of Computer Science and Engineering, Chalmers and University of Gothenburg, Gothenburg, Sweden
[2]Systemite AB, Gothenburg, Sweden
[3]University of L'Aquila, L'Aquila, Italy

Correspondence
Rebekka Wohlrab, Department of Computer Science and Engineering, Chalmers and University of Gothenburg, Gothenburg, Sweden.
Email: wohlrab@chalmers.se

Funding information
VINNOVA FFI, Grant/Award Number: 2014-05599 and 2015-04881; Knut and Alice Wallenberg Foundation

Summary

Agile methods are increasingly introduced in automotive companies in the attempt to become more efficient and flexible in the system development. The adoption of agile practices influences communication between stakeholders and makes companies rethink the management of artifacts and documentation like requirements, safety compliance documents, and architecture models. Practitioners aim to reduce irrelevant documentation but face a lack of guidance to determine what artifacts are needed and how they should be managed. This paper presents artifacts, challenges, guidelines, and practices for the continuous management of systems engineering artifacts in automotive based on a theoretical and empirical understanding of the topic. In collaboration with 53 practitioners from six automotive companies, we conducted a design-science study involving interviews, a questionnaire, focus groups, and practical data analysis of a systems engineering tool. The guidelines suggest the distinction between artifacts that are shared among different actors in a company (boundary objects) and those that are used within a team (locally relevant artifacts). We propose an analysis approach to identify boundary objects and three practices to manage systems engineering artifacts in industry.

KEYWORDS
agile systems engineering, boundary objects, design science, documentation

BOMI (Boundary Object and Methodological Islands) First Steps

2020 IEEE/ACM International Conference on Software and System Processes (ICSSP)

Charting Coordination Needs in Large-Scale Agile Organisations with Boundary Objects and Methodological Islands

Rashidah Kasauli
Chalmers | University of Gothenburg
Gothenburg, Sweden
Makerere University, Uganda
rashida@chalmers.se

Rebekka Wohlrab
Chalmers | University of Gothenburg
Systemite AB
Gothenburg, Sweden
wohlrab@chalmers.se

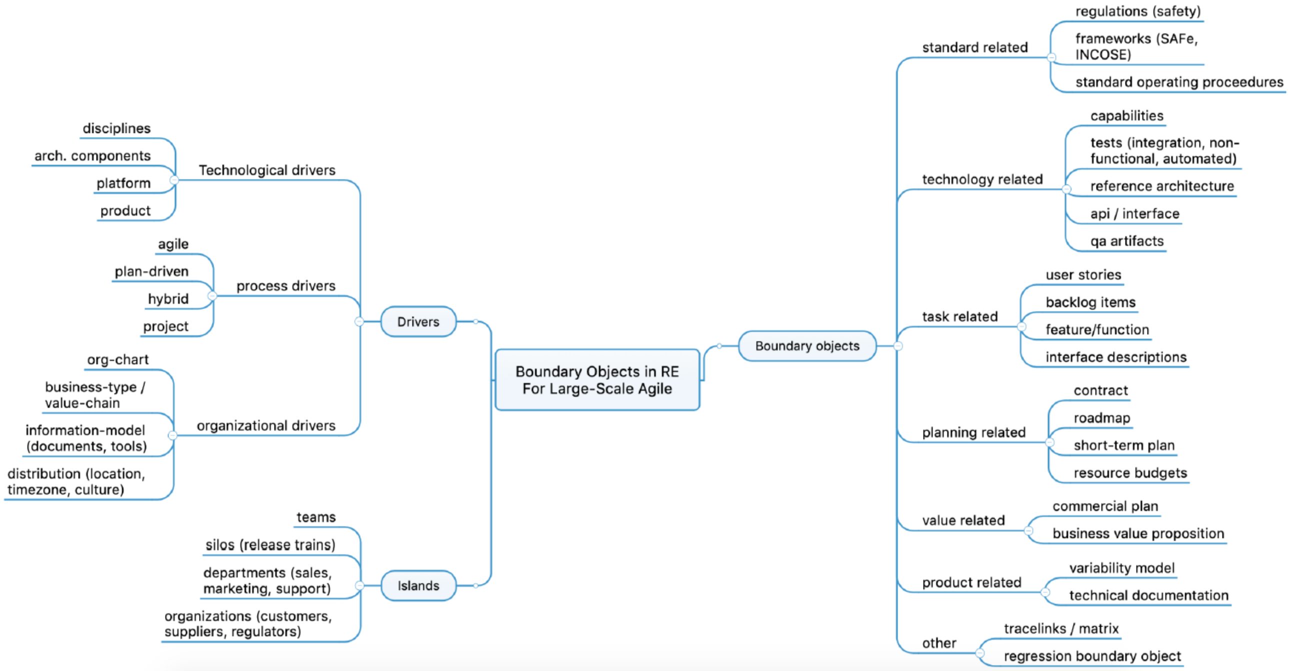
Eric Knauss
Chalmers | University of Gothenburg
Gothenburg, Sweden
eric.knauss@cse.gu.se

Jan-Philipp Steghöfer
Chalmers | University of Gothenburg
Gothenburg, Sweden
jan-philipp.steghofer@cse.gu.se

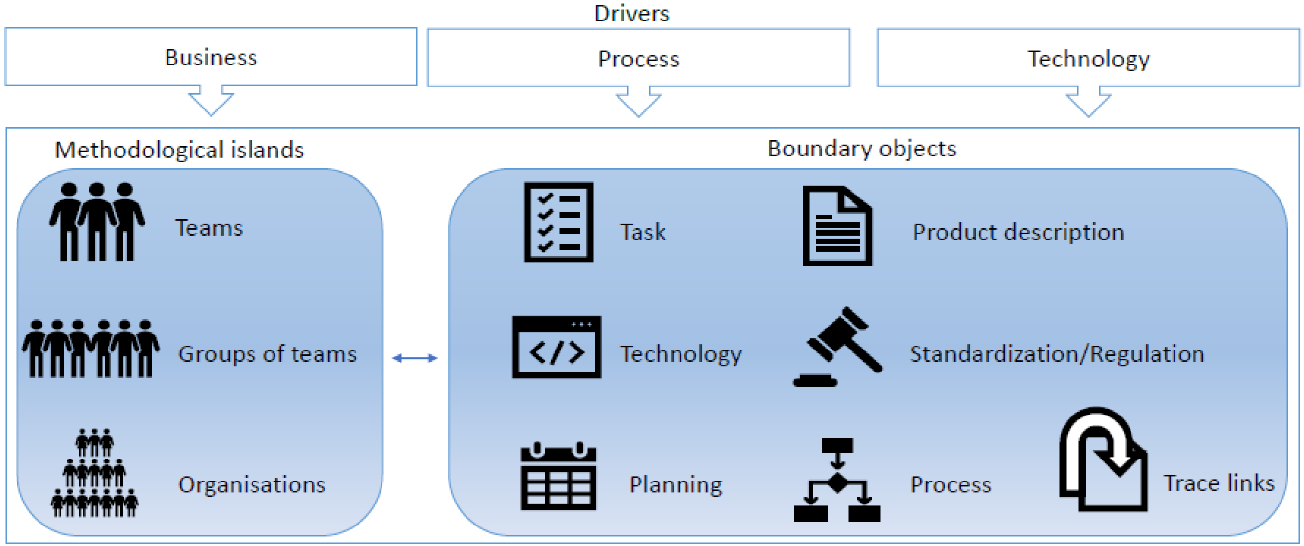
Jennifer Horkoff
Chalmers | University of Gothenburg
Gothenburg, Sweden
jenho@chalmers.se

Salome Maro
Chalmers | University of Gothenburg
Gothenburg, Sweden
salome.maro@cse.gu.se

BOMI (Boundary Object and Methodological Islands) First Steps



BOMI (Boundary Object and Methodological Islands) First Steps



BOMI Modeling

Modeling and Analysis of Boundary Objects and Methodological Islands in Large-Scale Systems Development

Rebekka Wohlrab[1,2](✉), Jennifer Horkoff[1], Rashidah Kasauli[1], Salome Maro[1], Jan-Philipp Steghöfer[1], and Eric Knauss[1]

[1] Chalmers|University of Gothenburg, Gothenburg, Sweden
{wohlrab,jenho,rashida}@chalmers.se,
{salome.maro,jan-philipp.steghofer,eric.knauss}@cse.gu.se
[2] Systemite AB, Gothenburg, Sweden

Abstract. Large-scale systems development commonly faces the challenge of managing relevant knowledge between different organizational groups, particularly in increasingly agile contexts. In previous studies, we found the importance of analyzing methodological islands (i.e., groups using different development methods than the surrounding organization) and boundary objects between them. In this paper, we propose a metamodel to better capture and analyze coordination and knowledge management in practice. Such a metamodel can allow practitioners to describe current practices, analyze issues, and design better-suited coordination mechanisms. We evaluated the conceptual model together with four large-scale companies developing complex systems. In particular, we derived an initial list of bad smells that can be leveraged to detect issues and devise suitable improvement strategies for inter-team coordination in large-scale development. We present the model, smells, and our evaluation results.

Keywords: Boundary objects · Agile development · Empirical studies

- Metamodel
- Smells
- Guidelines

- Tooling
- Views

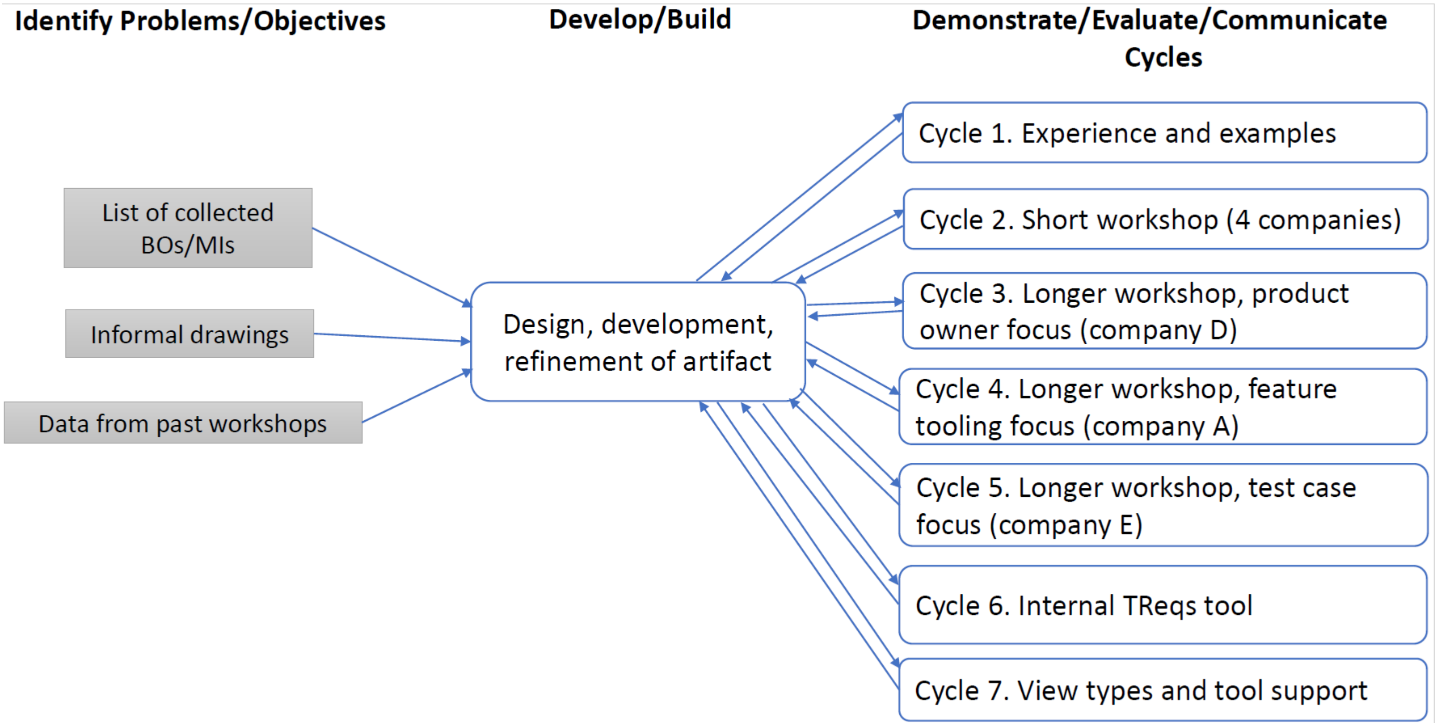
Holtmann, Jörg, Jennifer Horkoff, Rebekka Wohlrab, Victoria Vu, Rashidah Kasauli, Salome Maro, Jan-Philipp Steghöfer, and Eric Knauss. 2024. “Using Boundary Objects and Methodological Island (BOMI) Modeling in Large-Scale Agile Systems Development.” *Software and Systems Modeling (SoSyM).*

Research Methodology (Exemplar)

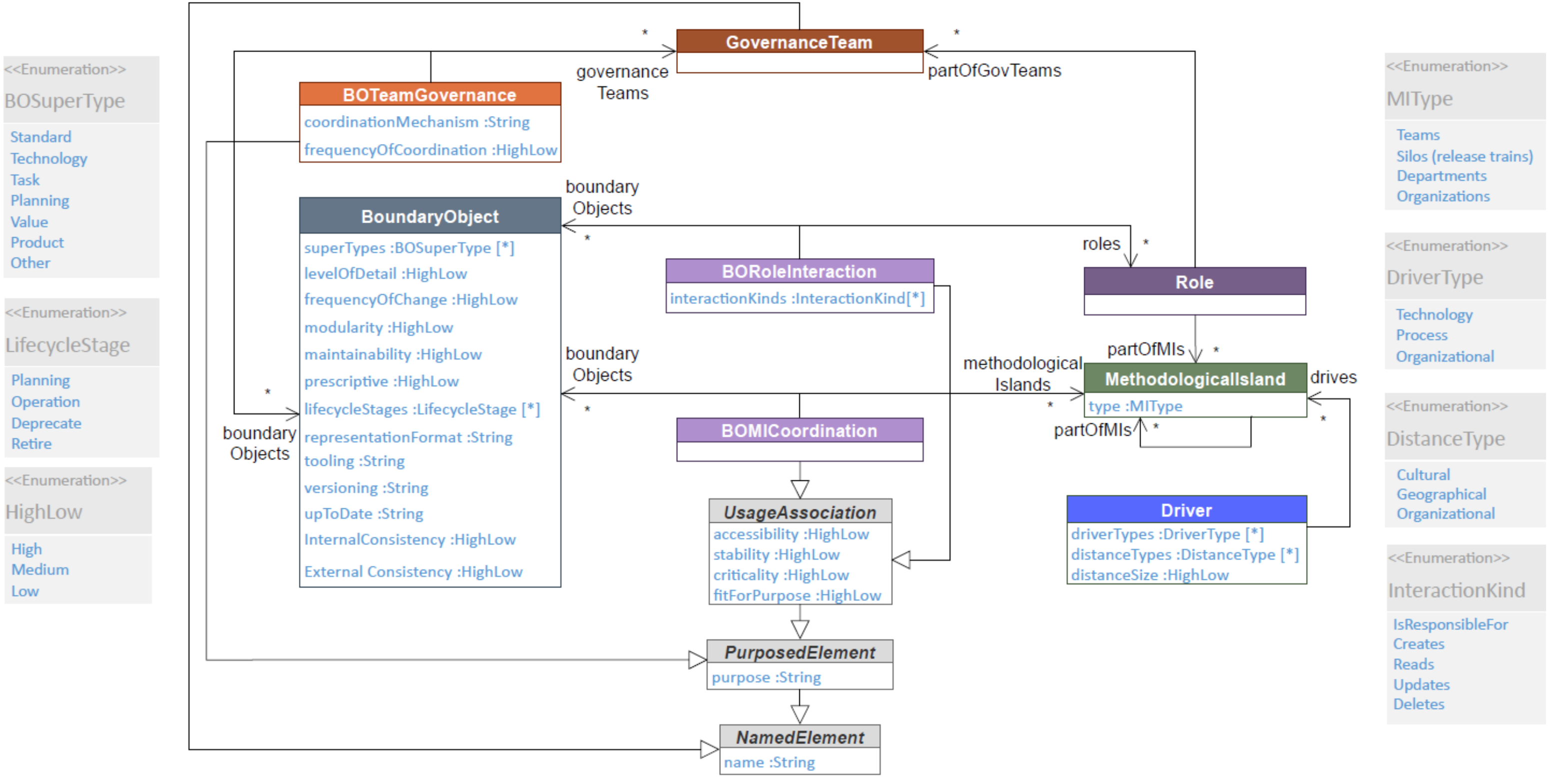
Table 3 Descriptions of participating companies.

Company A	Develops telecommunications products. Separate organizational units exist for sales, product management, and other purposes.
Company B	Develops mechanical products, both for consumer markets and for industrial development and manufacturing. The systems are decomposed into several elements, which is also reflected in the organizational structure.
Company C	Is an automotive Original Equipment Manufacturer (OEM). Traditionally, the company has been structured according to vehicle parts (e.g., powertrain, chassis, ...), but has undergone restructuring into agile teams.
Company D	Develops high-tech solutions for vehicular systems. Software development teams are largely independent of hardware development.
Company E	Develops hardware and software products for consumers and industries. Follows a test-driven, scaled agile methodology.

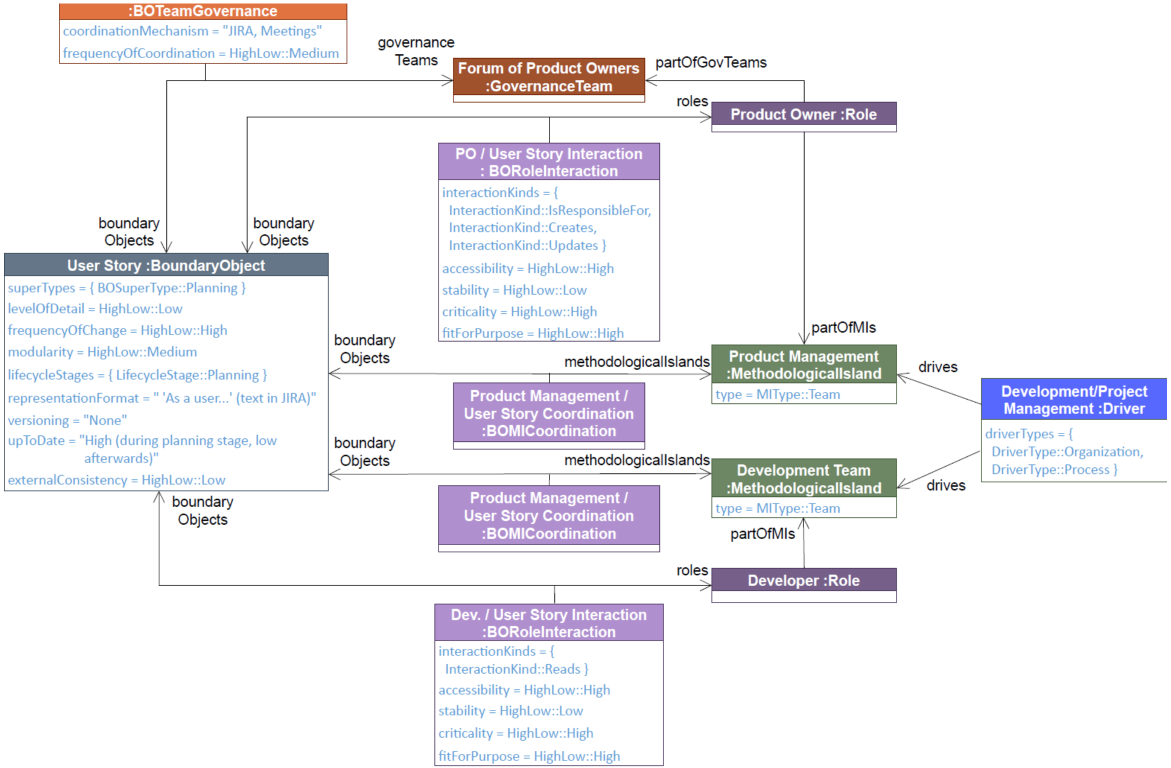
Research Methodology (Exemplar)



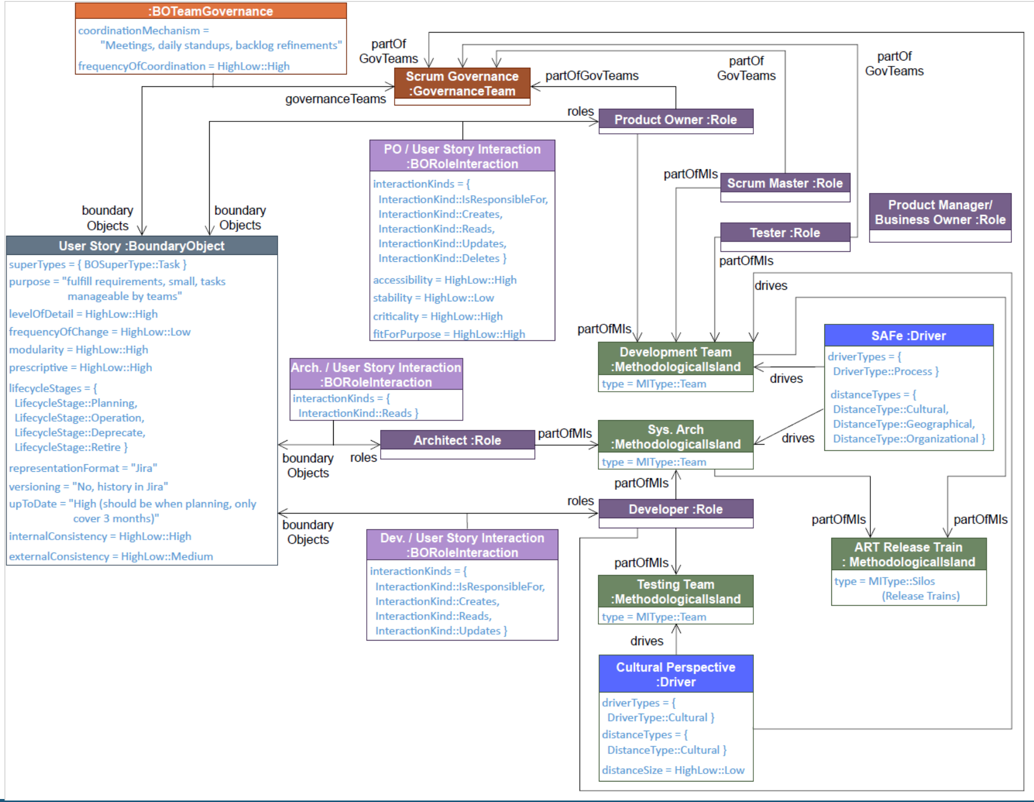
BOMI Metamodel



Example Instance Models from Workshop (1/2)



Example Instance Models from Workshop (2/2)



Elicitation Guide

Overview

- Which BO would you like to focus on
  - Pick one that's problematic in some way
- What roles interact with the BO?
- Which islands do the roles belong in?
- Is the BO governed, by whom?

About the BO

- What type is it? Sub-type?
- What is the: level of detail, frequency of change, modularity, etc..

About the Roles

- Do the roles CRUD with the BO?
- How about Accessibility, stability, criticality, etc. for that role

About the Islands

- What type are they?
- What is the distance, driver between each island

About Governance

- Which roles are part of the governance team?
- What frequency of governance, formality of governance

BOMI Smells (early examples)

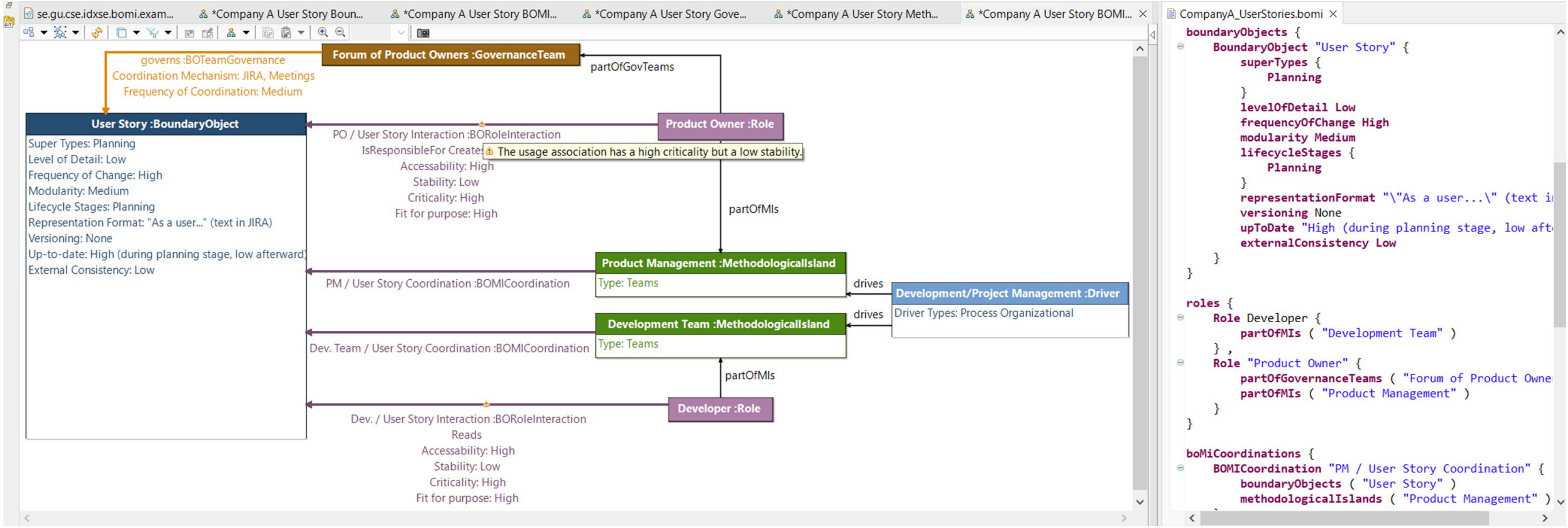
Smell Type	Description	OCL
Within BO	Low Modularity	context BoundaryObject inv LowModularity: self.Modularity = “Low"
	High level of detail, frequent change	context BoundaryObject inv DetailedHighChange: self.LevelofDetail = “High" and self.FrequencyofChange = “High"
Within Usage	Not fit for purpose	context Usage inv NotFit: self.FitForPurpose = “Low"
	High criticality, low stability	context Usage inv CriticalUnstable: self.Criticality = “High" and self.Stability = “Low"
Presence of…	No governance team	context BoundaryObject inv Governed: self.Governed->size > 0
	No one responsible for BO	context BoundaryObject inv Responsible: self.Responsible-> size > 0
Across Elements	Governing roles should use BO	context BoundaryObject inv GovernsUses: self.Governs -> forAll(g | g.PartOf-> select(r | r.uses = self)->size > 0)

BOMI Guidelines

BOMI Guidelines

1. Participants: BOMI modeling session should involve participants who are at a middle management level. Those too new or focused, or those too high up may not care about the view provided by BOMI. Inviting at least one representative for key roles in the model is helpful.
2. Subject matter: During a first BOMI session, it can be useful to focus on a BO which is not particularly problematic, in order to test out the method and allow for BOMI training. However, avoid BOs that are well understood or not problematic in some way.
3. Duration: It is useful to split BOMI training and modeling into two different sessions, giving time to think about the new information. One to two hours for the modeling and discussion session are typically sufficient.
4. View: After creating the full model, consider creating a simplified, condensed view for upper-level management or those who only need to read the model (e.g., new employees).
5. Tooling: The tooling used for a BO should not be a BO itself. Focus on the information contained in the tool, and not the tool itself.
6. Scope: For readability, focus on one main BO at a time. Further BOs could be explored in different models.
7. MI focus: Elicit the MI view. Often, it is easy to think in terms of roles, but also think in terms of the different ways that roles and groups are separate from each other, e.g., by process, distance, etc.
8. Smells: Be open to find and discuss issues highlighted by aspects of the BOMI model, even if they are not explicitly identified by a formal smell.

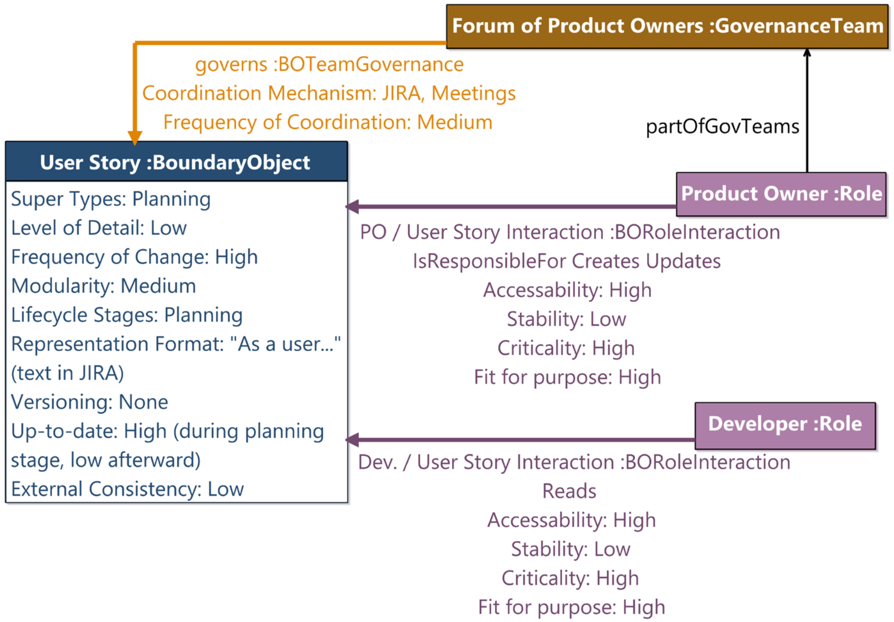
BOMI Tool


se.gu.cse.idxse.bomi.exam...
*Company A User Story Boun...
*Company A User Story BOMI...
*Company A User Story Gove...
*Company A User Story Meth...
*Company A User Story BOMI...
CompanyA_UserStories.bomi
Forum of Product Owners :GovernanceTeam
governs :BOTeamGovernance
Coordination Mechanism: JIRA, Meetings
Frequency of Coordination: Medium
partOfGovTeams
User Story :BoundaryObject
Super Types: Planning
Level of Detail: Low
Frequency of Change: High
Modularity: Medium
Lifecycle Stages: Planning
Representation Format: "As a user..." (text in JIRA)
Versioning: None
Up-to-date: High (during planning stage, low afterward)
External Consistency: Low
Product Owner :Role
PO / User Story Interaction :BORoleInteraction
IsResponsibleFor Create
The usage association has a high criticality but a low stability.
Accessability: High
Stability: Low
Criticality: High
Fit for purpose: High
partOfMIs
Product Management :MethodologicalIsland
Type: Teams
PM / User Story Coordination :BOMICoordination
drives
Development/Project Management :Driver
Driver Types: Process Organizational
Development Team :MethodologicalIsland
Type: Teams
Dev. Team / User Story Coordination :BOMICoordination
partOfMIs
Developer :Role
Dev. / User Story Interaction :BORoleInteraction
Reads
Accessability: High
Stability: Low
Criticality: High
Fit for purpose: High
boundaryObjects {
BoundaryObject "User Story" {
superTypes {
Planning
}
levelOfDetail Low
frequencyOfChange High
modularity Medium
lifecycleStages {
Planning
}
representationFormat "\"As a user...\" (text i
versioning None
upToDate "High (during planning stage, low aft
externalConsistency Low
}
}
roles {
Role Developer {
partOfMIs ("Development Team")
} ,
Role "Product Owner" {
partOfGovernanceTeams ("Forum of Product Owne
partOfMIs ("Product Management")
}
}
boMiCoordinations {
BOMICoordination "PM / User Story Coordination" {
boundaryObjects ("User Story")
methodologicalIslands ("Product Management")

BOMI Views

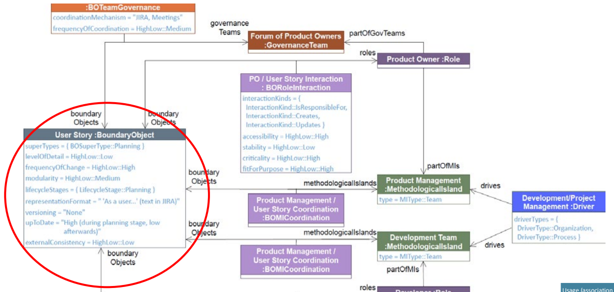
- Five views
  - BOMI details
  - Overview
  - Methodological island (MI)
  - Boundary object (BO)
  - Governance

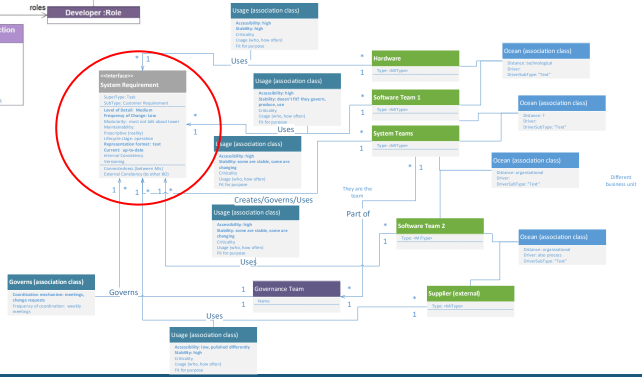
Governance View Example

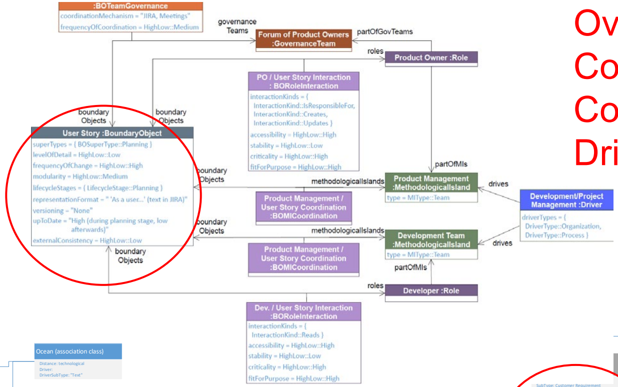


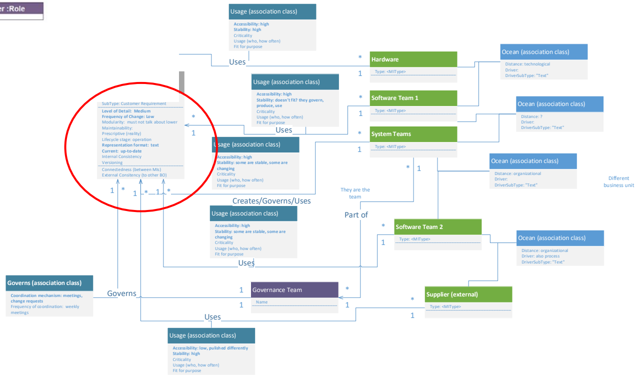
Back to Information Models

Many instance BOMI Models capturing coordination between teams









Overview?
Coordination between objects?
Coordination in Islands?
Drivers?

Back to Information Models

Need further information models for wider coordination

To link important things together

Scattered BOs are not enough?

Missing big picture -> focus on wider Information Models

Information Model Example: TIM

Managing Traceability Information Models: Not Such a Simple Task After All?

Salome Maro, Jan-Philipp Steghöfer, Eric Knauss, Jennifer Horkoff, and Rashidah Kasauli, Chalmers University of Gothenburg

Rebekka Wohlrab, Systemite AB, Gothenburg, and Chalmers University of Technology, Gothenburg

Jesper Lysemose Korsgaard, Florian Wartenberg, and Niels Jørgen Strøm, Grundfos Holding A/S

Ruben Alexandersson, Volvo Cars

// Practitioners are poorly supported by the scientific literature when managing traceability information models (TIMs), which capture

IEEE Soft. 2021

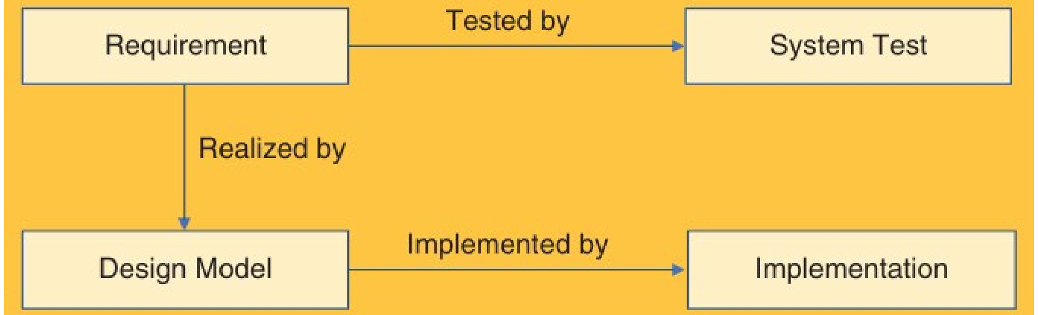


FIGURE S1. A simple TIM as an example. It connects four different artifact types (requirements, system tests, design models, and implementation) with three different trace link types (tested by, realized by, and implemented by).

Information Model Example: RIM

The Journal of Systems and Software 162 (2020) 110516

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

ELSEVIER

Why and how to balance alignment and diversity of requirements engineering practices in automotive

Check for updates

Rebekka Wohlrab[a,b,*], Eric Knauss[a], Patrizio Pelliccione[a,c]

[a] Chalmers | University of Gothenburg, Gothenburg, Sweden
[b] Systemite AB, Gothenburg, Sweden
[c] University of L'Aquila, L'Aquila, Italy

ARTICLE INFO

Article history:
Received 1 July 2019
Revised 4 November 2019
Accepted 23 December 2019
Available online 26 December 2019

Keywords:
Requirements information models
Aligning software engineering practices
Automotive software engineering
Large-scale software development
Mixed methods research

ABSTRACT

In large-scale automotive companies, various requirements engineering (RE) practices are used across teams. RE practices manifest in Requirements Information Models (RIM) that define what concepts and information should be captured for requirements. Collaboration of practitioners from different parts of an organization is required to define a suitable RIM that balances support for diverse practices in individual teams with the alignment needed for a shared view and team support on system level. There exists no guidance for this challenging task. This paper presents a mixed methods study to examine the role of RIMs in balancing alignment and diversity of RE practices in four automotive companies. Our analysis is based on data from systems engineering tools, 11 semi-structured interviews, and a survey to validate findings and suggestions. We found that balancing alignment and diversity of RE practices is important to consider when defining RIMs. We further investigated enablers for this balance and actions that practitioners take to achieve it. From these factors, we derived and evaluated recommendations for managing RIMs in practice that take into account the lifecycle of requirements and allow for diverse practices across sub-disciplines in early development, while enforcing alignment of requirements that are close to release.

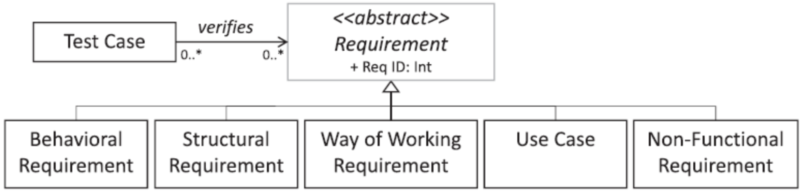


Fig. 3. Excerpt of a minimal RIM at OEM2 with aligned aspects.

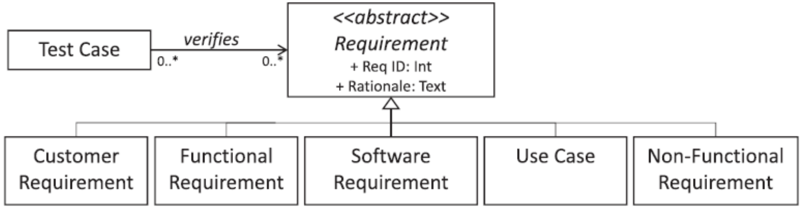


Fig. 4. Excerpt of a minimal RIM at SUP with aligned aspects.

Information Models

- Capture important objects (boundary objects) and their interrelationships
- Could capture also technical elements like architecture/hardware
- Or process/human aspects like teams or methodological islands

- Depends on what the organization needs to coordination over
- We can’t design *IMs for organizations, it depends on what they need
- But perhaps we can provide guidance in IM design and use

Information Model Propositions

To be submitted!

Information Model Design and Evolution in Scaled-Agile: Why, What, and How

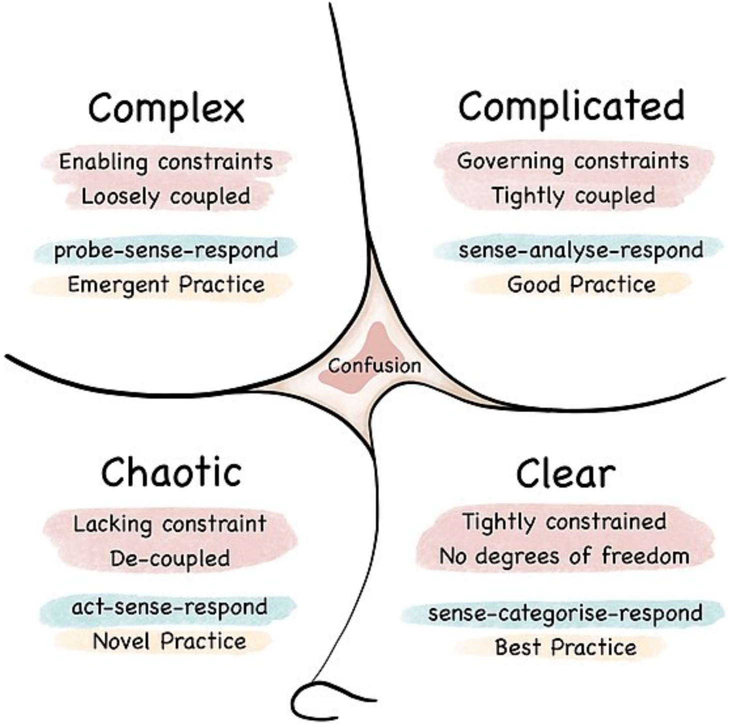
Eric Knauss, Abdullatif Alshriaf, Jennifer Horkoff, Baldvin Gislason Bern, Alessia Knauss, Anders Kvist, Filip Lange, Robert Nilsson, Jesper Thyssen, Jesper Ørting
Chalmers | University of Gothenburg, Sweden

Abstract—**When applying large-scale agile frameworks in systems engineering, it is crucial to balance autonomy and flexibility on team level with managing cross-cutting concerns on system level. We have come to see information modeling as a key enabler to maintain this balance.**

Information Model Propositions

Some necessary background:

Cynefin (1999 Dave Snowden)



https://commons.wikimedia.org/wiki/File:Cynefin_framework_2022.jpg

Information Model Propositions

- Proposition 1: Complex / Complicated (as per Cynefin) situations lead to different usage of information models
- Proposition 2: Information models must be collaboratively designed across systems engineering disciplines
- Proposition 3: An information model is best represented as an Ontology
- Proposition 4a: In complex situations, Information models are best designed bottom-up
- Proposition 4b: In complicated situations, Information models are best designed top-down
- Proposition 5: Generic modeling best practices are sufficient for information modeling
- Proposition 6: An extension of our BOMI guidelines will help with information modeling

FAMER

Facilitating Multi-Party Engineering of Requirements

UNIVERSITY OF GOTHENBURG

RI. SE

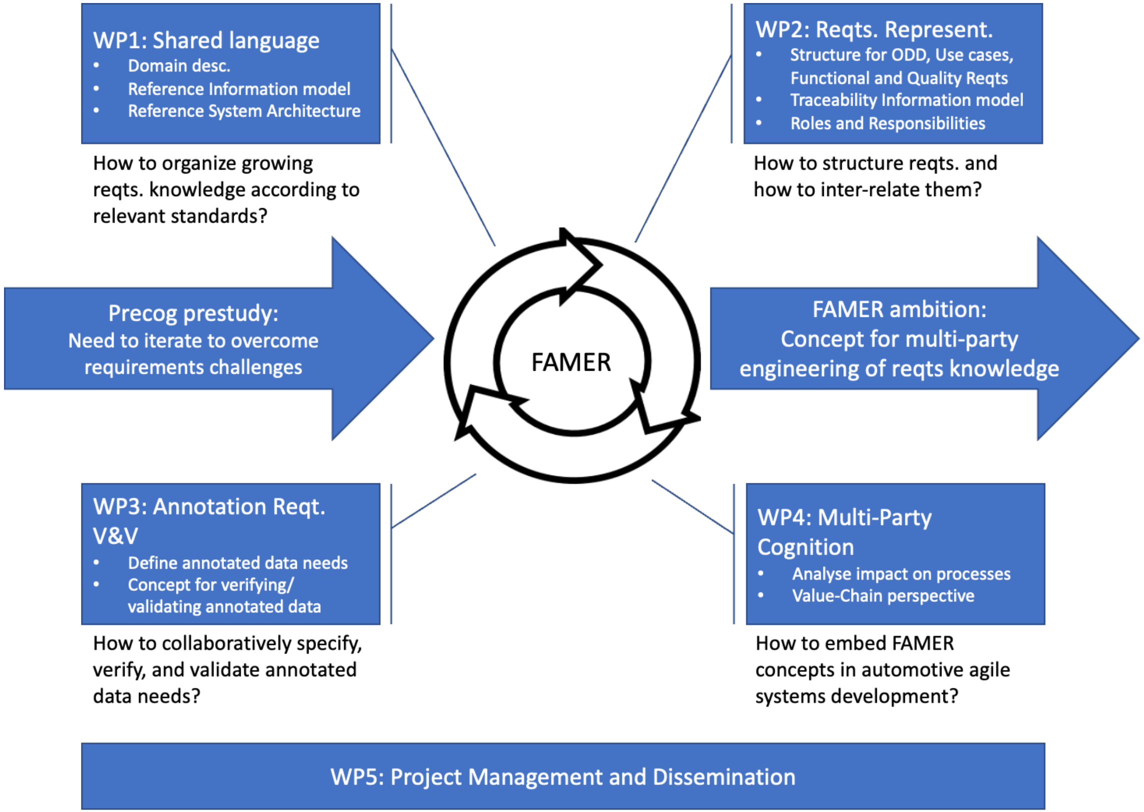
zenseact

KOGNIC

VOLVO

FAMER

FAMER will establish concepts, models, and techniques of effectively building requirements knowledge for safe perception systems.



Overview – Work in Progress

Observations in FAMER working with a hypothetical perception system use case

Proposition	Insights
P2: IM designed collaboratively across disciplines	Collaborations thus far via Miro and PowerPoint, disciplinary differences can make the process challenging
P3: Differ between complex and complicated	Perception systems are complex, but can be narrowed down to be complicated
P4a: Define IM bottom-up in complex situations	Did a mix a top-down and bottom-up, hard to meet in the middle
P3: Represent IM as an ontology	Yes, but a simple one, depends definition of “ontology”
P5: Modeling practices are insufficient	What is available would likely not help us so much
P6: BOMI Guidelines would help	Have followed some implicitly, but have kept BOMI out for simplicity

Conclusions

- Information Modeling for the continuous revolution – if it's not agile, it will be whatever's next
- Balance between methodological autonomy and coordination
- BOMI modeling
- Information Modeling
- Propositions and Applications

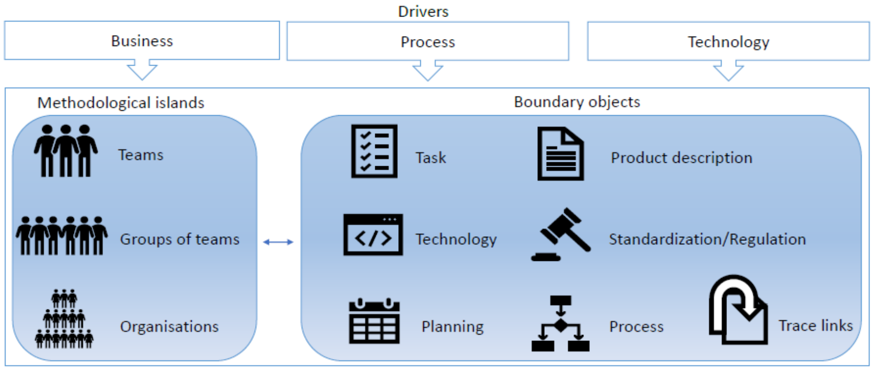
- Modeling as a form of coordination and knowledge management

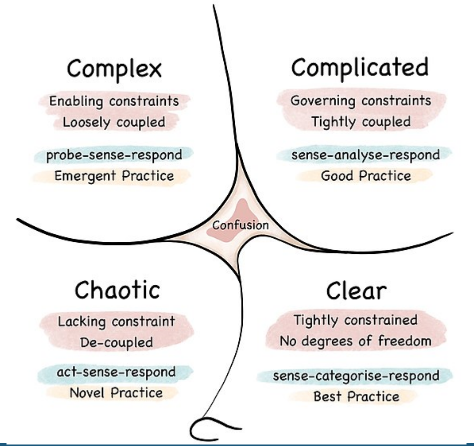
Future Work

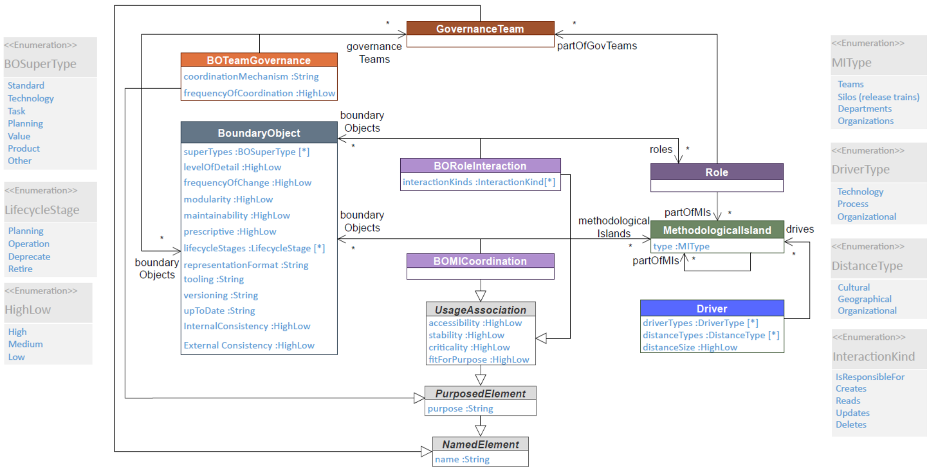
- Working on applying BOMI to model standards
- Support via git-based tools (TReqs) ongoing
- Further IM examples and guidance

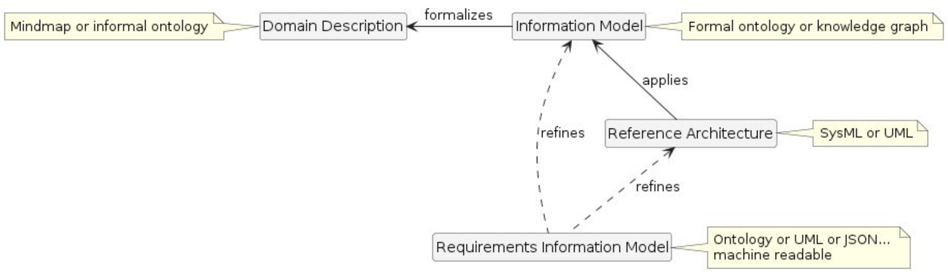
- Publish more ☺ (Challenging)

Discussion ☺


Drivers
Business
Process
Technology
Methodological islands
Teams
Groups of teams
Organisations
Boundary objects
Task
Product description
Technology
Standardization/Regulation
Planning
Process
Trace links


Complex
Enabling constraints
Loosely coupled
probe-sense-respond
Emergent Practice
Complicated
Governing constraints
Tightly coupled
sense-analyse-respond
Good Practice
Confusion
Chaotic
Lacking constraint
De-coupled
act-sense-respond
Novel Practice
Clear
Tightly constrained
No degrees of freedom
sense-categorise-respond
Best Practice


<<Enumeration>>
BOSuperType
Standard
Technology
Task
Planning
Value
Product
Other
<<Enumeration>>
LifecycleStage
Planning
Operation
Deprecate
Retire
<<Enumeration>>
HighLow
High
Medium
Low
GovernanceTeam
governance Teams
partOfGovTeams
BOTeamGovernance
coordinationMechanism :String
frequencyOfCoordination :HighLow
boundary Objects
BoundaryObject
superTypes :BOSuperType [*]
levelOfDetail :HighLow
frequencyOfChange :HighLow
modularity :HighLow
maintainability :HighLow
prescriptive :HighLow
lifecycleStages :LifecycleStage [*]
representationFormat :String
tooling :String
versioning :String
upToDate :String
InternalConsistency :HighLow
External Consistency :HighLow
BORoleInteraction
interactionKinds :InteractionKind[*]
roles
Role
partOfMIs
methodological Islands
MethodologicalIsland
type :MIType
drives
BOMICoordination
UsageAssociation
accessibility :HighLow
stability :HighLow
criticality :HighLow
fitForPurpose :HighLow
Driver
driverTypes :DriverType [*]
distanceTypes :DistanceType [*]
distanceSize :HighLow
PurposedElement
purpose :String
NamedElement
name :String
<<Enumeration>>
MIType
Teams
Silos (release trains)
Departments
Organizations
<<Enumeration>>
DriverType
Technology
Process
Organizational
<<Enumeration>>
DistanceType
Cultural
Geographical
Organizational
<<Enumeration>>
InteractionKind
IsResponsibleFor
Creates
Reads
Updates
Deletes


Mindmap or informal ontology
Domain Description
formalizes
Information Model
Formal ontology or knowledge graph
applies
refines
Reference Architecture
SysML or UML
refines
Requirements Information Model
Ontology or UML or JSON…
machine readable