Integrating Creative Artifacts into Software Engineering Processes

Seventeenth International Conference on Creative Content Technologies, CONTENT 2025

April 9, 2025 - Valencia, Spain

Hans-Werner Sehring

NORDAKADEMIE
HOCHSCHULE DER WIRTSCHAFT

Agenda

01 **Visual SE Artifacts**
Model-Driven Software Engineering and Creative Software Engineering Artifacts

02 **Content References**
Structured, semi-, and unstructured documents, content of mutable documents

03 **The M³L**
The Minimalistic Meta Modeling Language

04 **Experiment**
Interpretation of semistructured documents, reinterpretation of mutable documents

05 **Conclusion**
Summary and Outlook

01

Visual Software Engineering Artifacts

Model-driven Software Engineering (MDSE)

Various approaches to model-driven software engineering exist.

- **Sequence of Models, created by Model-to-Model Transformations**
- **Example: Model-driven Architecture (MDA)**
  - Early MDSE approach
  - Models are created on (originally) three levels of abstraction

  1) A *Computation-Independent Model* (*CIM*) from the perspective of the subject domain.
  2) A *Platform-Independent Model* (*PIM*) as a first formal model.
  3) Transformed into a *Platform-Specific Model* (*PSM*) used to generate a working implementation.

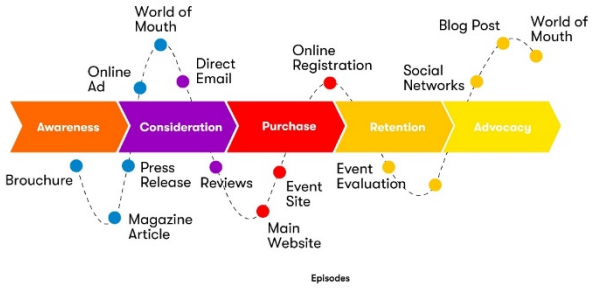
- **Software Generation**
  - Often by **Model-To-Text Transformation**
  - Different approaches, e.g., metaprogramming, templates, generative AI
- **Domain-specific Languages (DSLs)**
  - Languages ≙ Metamodels
  - Defined for a specific domain
- **Generic Software**
  - A domain model was used during the development of the software
  - Configurable software (low code / no code development)

MDSE in Practice

Software engineering processes often incorporate creative work.

MDSE approaches often based on **formal** models and model transformations

Software engineering (SE) **reality** (at least in some domains):



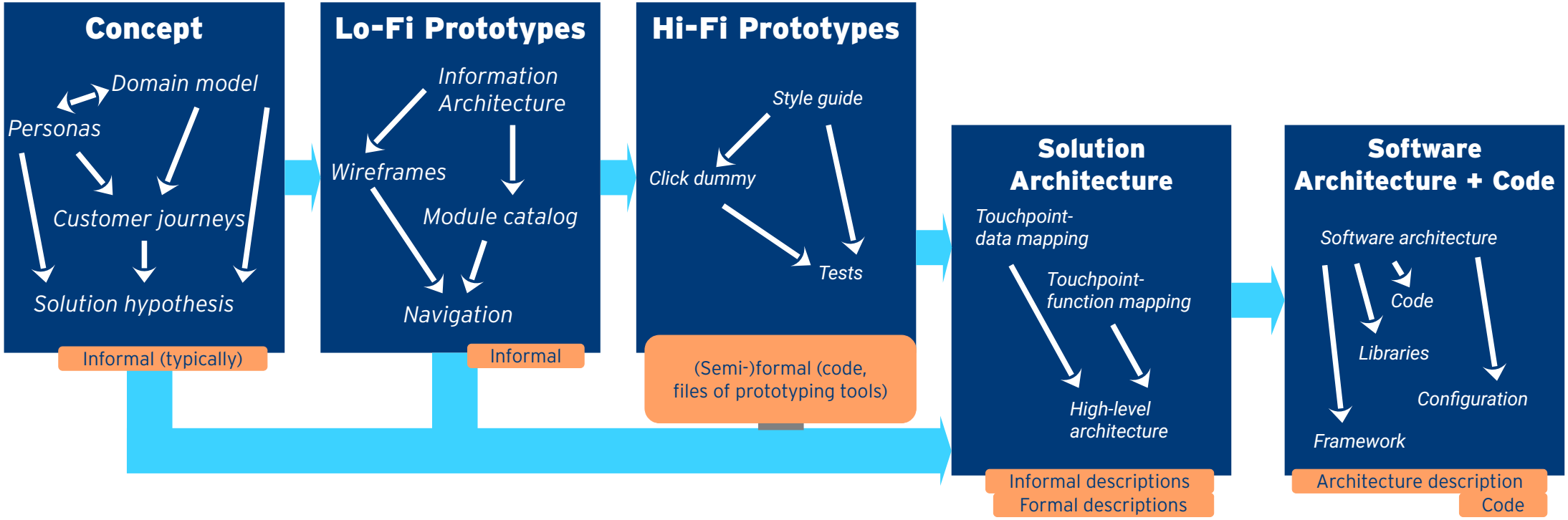
Depending on the kind of software and the kind of project, we find

- Heterogeneous modeling artifacts: varying **degrees of formalism, ambiguity, detail**, etc.
- Artifacts often part of a **methodology or a tool**: notation and representations matter
- Several project stages, **not only software** (engineering) related; from inception to operations stages

A Typical Software Engineering Process

Creative work has to be performed at all early stages of a software engineering process.

Typical artifacts involved in a software engineering process for software with creative aspects may look as follows:



Examples are UI-intensive applications, marketing solutions, etc.

Support for Informal Processes and Artifacts

There modeling and model transformations and creative work seem contradicting.

Given various process steps and artifacts that are

- not formal
- visual
- ambiguous
- not producible by model transformations
- etc.

We cannot have MDSE.

Still, we want …

- support in managing system (of systems) complexity
  - support in managing (modeling) artifacts
  - checks on models
- quick reactions to changing requirements
  - deriving software from specifications
  - traceability
- etc.

We want the benefits of MDSE.

Models and Documents

For software specified informally, try to model the documents describing it.

Not all aspects of software are **described by models**

- Results produced by creative workers
- Results produced using tools that create non-formal representations

But **specific documents** contain relevant information

- Results of creative work
- Alignment with non-technical stakeholders

Approach: extract (formal) facts from visual software engineering artifacts

Such an approach requires:

- Structure in (visual) documents
- Conceptualization of the documents' content

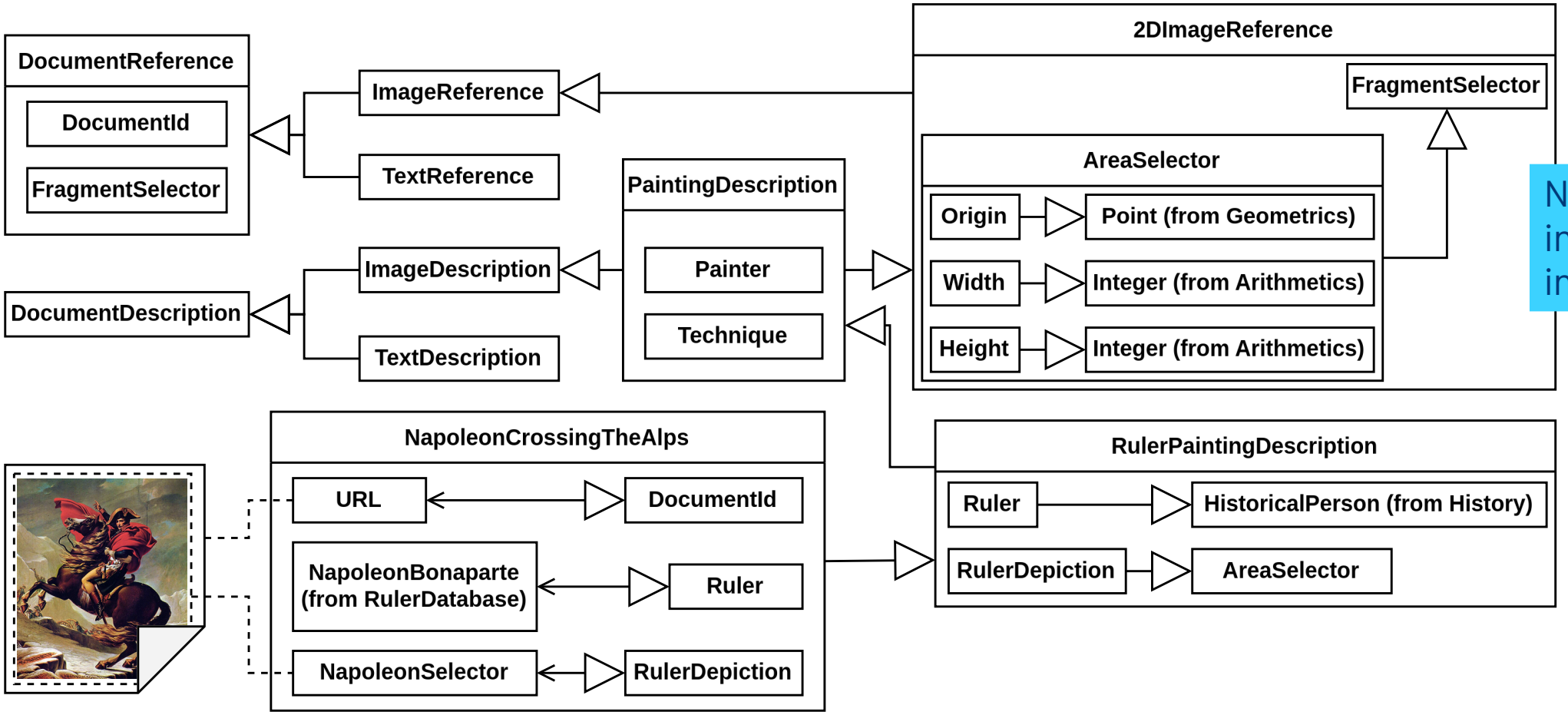
02

Content References

Example of a Static Document Description

An example from a different domain shall illustrate the idea.

We learned a lot about documents, their content, and descriptions of both from art history.



Notation is introduced in a moment

Examples of Description Models for Visual Artifacts

Fragments of static documents can be referenced from model elements.

CIM

(if we use MDA terminology here)

Personas

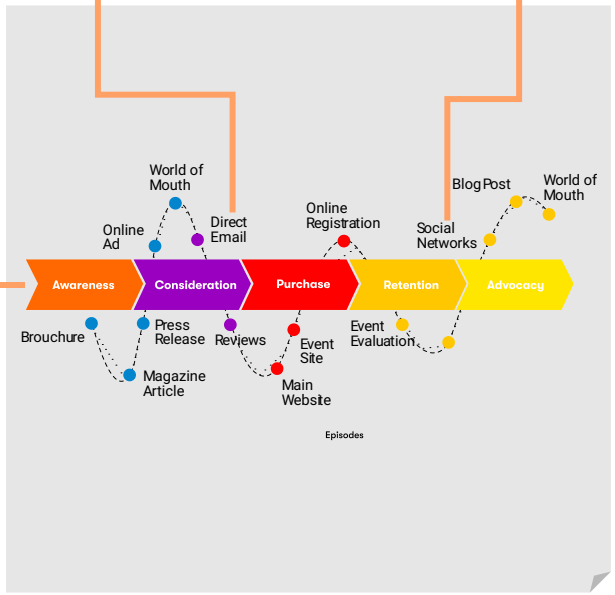
Persona A

Persona B

Customer Journey 1

Touchpoint 1.1

Touchpoint 1.2



Referencing Content of Informal Documents

Particular properties of software engineering artifacts have to be considered.

On top of the **typical problems with referencing content in documents** or fragments of documents

- Subjectivity, incompleteness, and inconsistency of documents
- Immutable structure or stable external IDs
- No strict “schema“ (document structure, layout, …)

Further properties of software documentation need to be considered

- Documents describe same software on **different levels of abstraction**, from **different perspectives**, etc.
- Artifacts in SE processes are **subject to constant change**
  - Working documents used for team collaboration
  - Incremental development (agile, in most cases)
- **Different audiences** are addressed, leading to changing terminilogy etc.

03

The Minimalistic Meta Modeling Language

M³L at a Glance

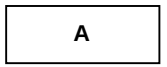
Basic language constructs. More complete descriptions can be found in the literature.

`A`	The declaration of or reference to a **concept** named A
`A is a B` `A is the B`	The **refinement** of a concept B to a concept A; A is a specialization of B, B is a generalization of A (the: A is the only specialization of B)
`A is a B { C }`	Containment of concepts; C belongs to the **content** of A, A is the **context** of C
`A |= D`	The **semantic rule** of a concept of a concept A; whenever A is referenced, D is bound; if D does not exist, it is created in the same context as A
`A |- E F G.`	The **syntactic rule** of a concept A; A is printed out as or recognized from the concatenation of the syntactic forms of concepts E, F, and G; if not defined, a concept evaluates to / is recognized from its name

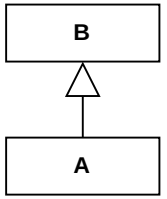
A Graphical Notation for the M³L

For the presentation in the paper, we use a graphical notation.

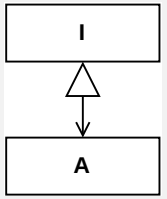
A



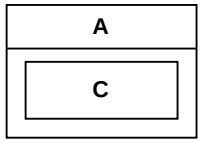
A is a B



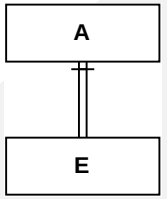
A is the B



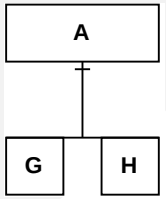
A is a B { C }



A |= D



A |- E F G.



04

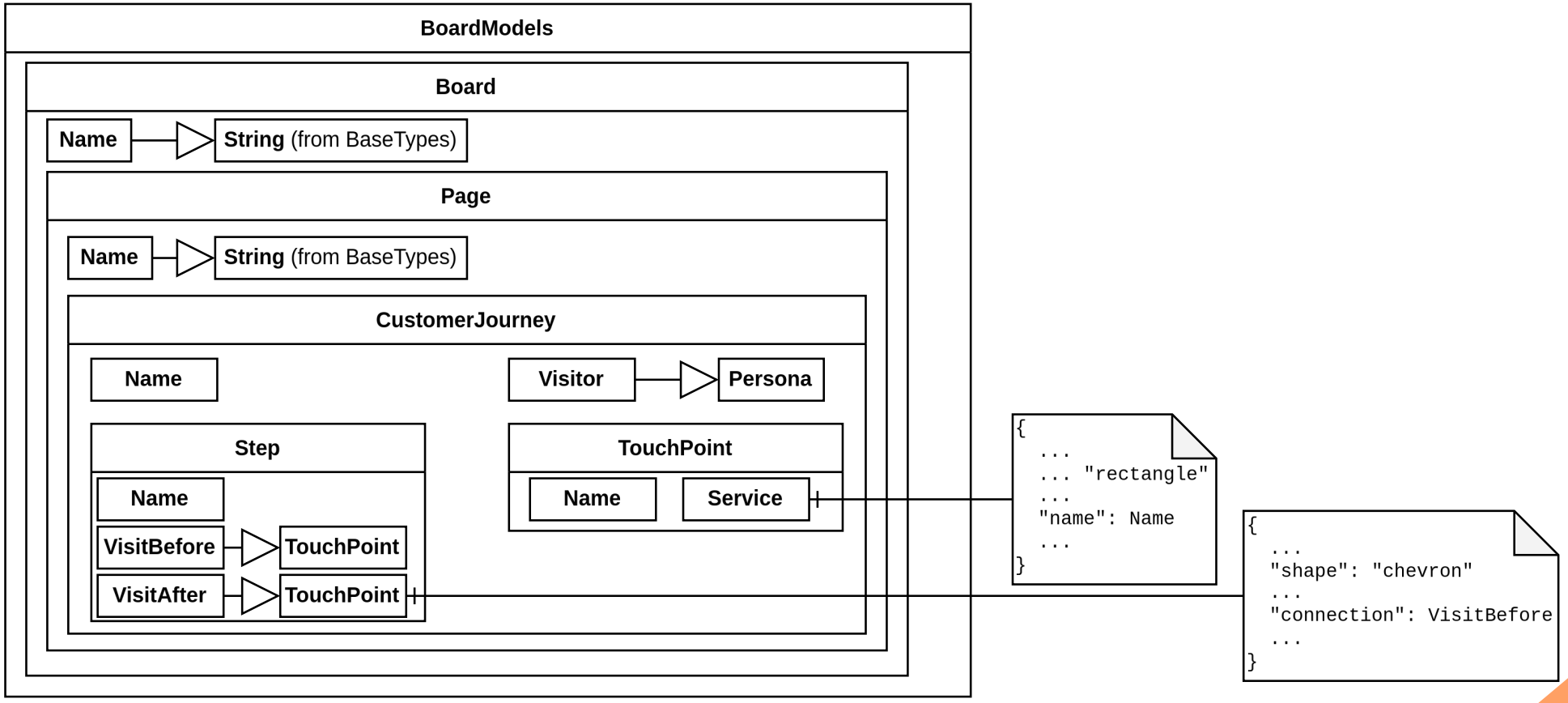
Experiment

Document Templates

Concepts are used as annotated grammars to recognize document content.

Syntactic rules form a **grammar** for document representations, for example, in a JSON format used by APIs of online tools.

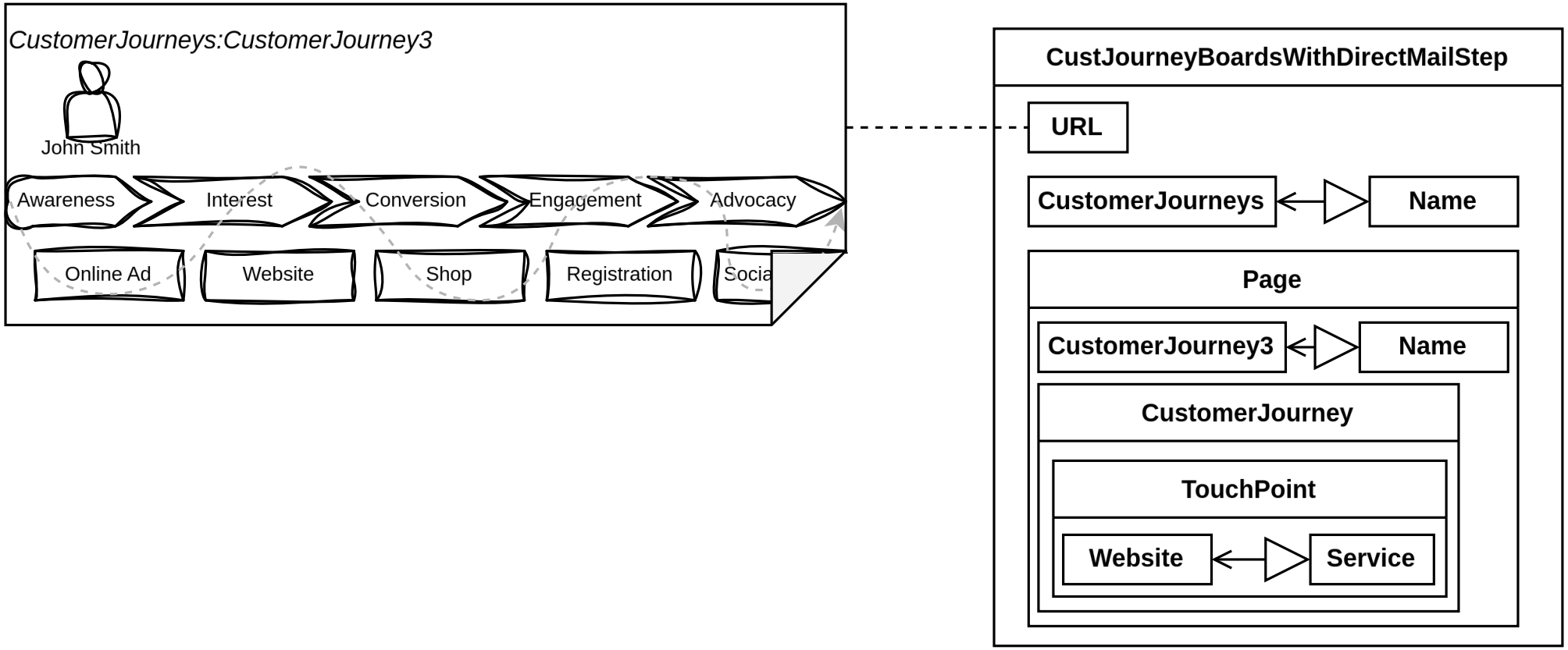
M³L concepts provide a **link to domain semantics** that is required to interpret documents



Document Queries

Queries / patterns are used to (re-) interpret mutable documents.

A **prototypical concept** reference can be used as a **pattern**.
When reading in a document, concepts are created / updated according to such a pattern.

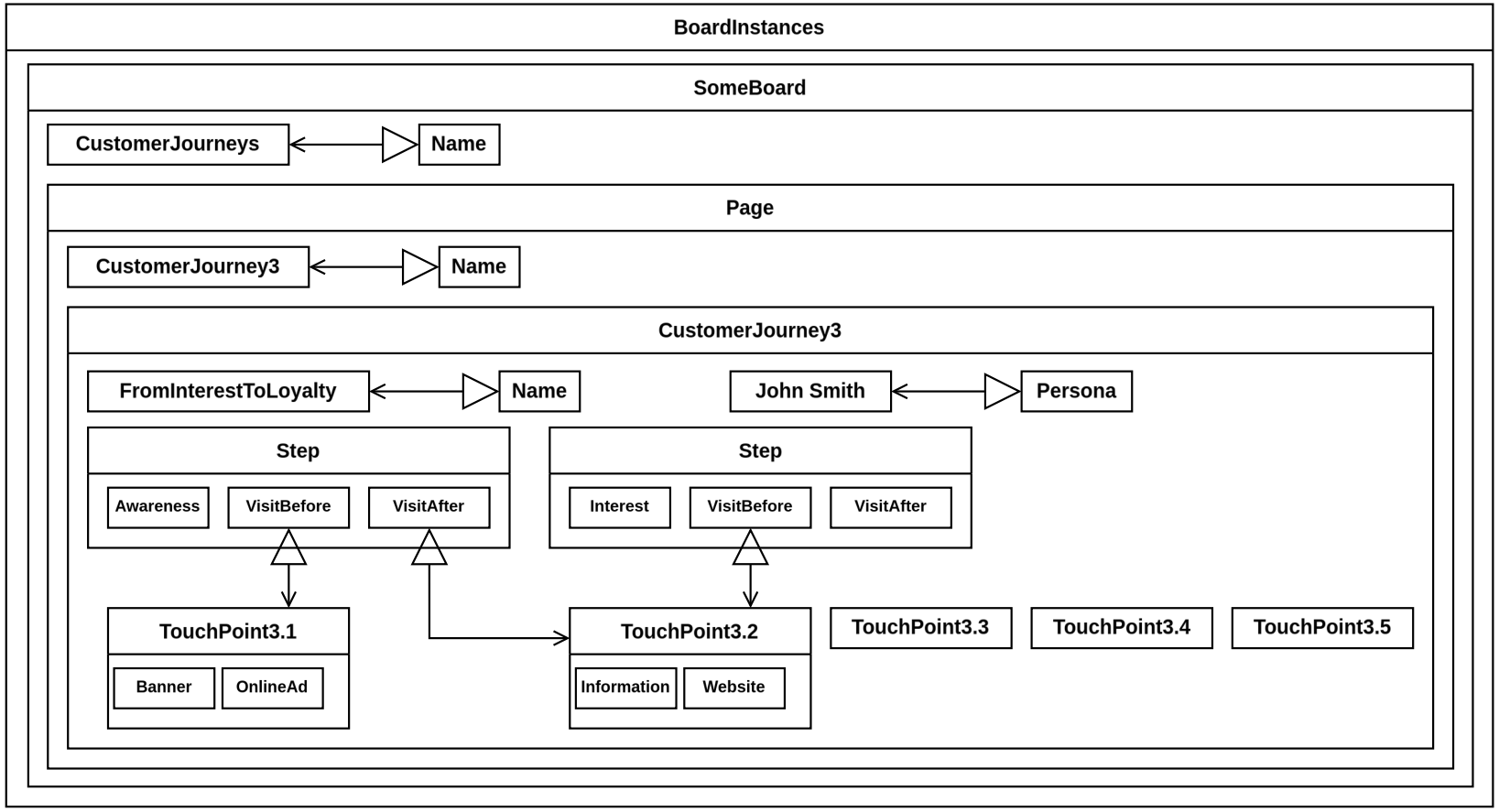


Recognizing Content in a Document

The result of reading in a document is a concept graph that result from the grammar.

Each recognized concept will be created as a refinement of the grammar concepts, with actual **content from the document**, structured **in terms of the domain model**.

This structured content serves as a model for the MDSE process.



05

Summary and Outlook

Conclusion

Summary and Outlook

Summary

Visual artifacts are an important part of software specification for a range of software solutions.

Per se, visual artifacts represented by informal documents do not integrate well with model-driven approaches.

First experiments showed cases where software models can be extracted from informal documents by means of a document “grammar” and links to domain models.

Outlook

Experiments with virtual whiteboard solutions showed that the simple structures provided in JSON formats required well-defined names to be used. So, basically, there are still stable IDs that need to be maintained.
We need to investigate how to overcome such need, for example by artificial IDs hash values of constructs provided as metadata.

NORDAKADEMIE
HOCHSCHULE DER WIRTSCHAFT

NORDAKADEMIE gAG Hochschule der Wirtschaft

Köllner Chaussee 11 · 25337 Elmshorn · Tel.: +49 (0) 4121 4090-0 · E-Mail: info@nordakademie.de · Web: www.nordakademie.de