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

# MANAGING DEPENDENCIES IN THE PRODUCT DEVELOPMENT PROCESS WITH SEMANTIC TECHNOLOGIES

**Thomas Syldatke[1], Marcus Lutz[1], Willy Chen[2] and Claudia Hess[2]**

[1]AUDI AG, Ingolstadt
[2]Achievo Inproware GmbH, Ingolstadt

*Keywords: ontologies, product development process*

## 1 INTRODUCTION

Managing the increasing complexity in product development requires describing and analyzing the dependencies within and between multiple domains, such as the customers' requirements or the product components. The Design Structure Matrix (DSM) and the Domain Mapping Matrix (DMM) approach provide well-established methods and tools to address this challenge (e.g. [1, 2]). We claim that Semantic Technologies can facilitate and enhance analysis in the scope of classic DSM/DMM approaches. Semantic technologies are a solution to represent complex domains in a formal manner and are a promising approach to handle complexity within businesses. In the paper, we will show how ontologies can be used for describing and analyzing dependencies in a multi-domain setting and discuss how this approach differs from DSM/DMM approaches and how to use them together. A use case from the automotive industry with focus on the product development process illustrates our approach to using semantic technologies for dependency management.

## 2 SEMANTIC TECHNOLOGIES

Today, technologies originally developed for use on the semantic web are gaining more and more attention from businesses, which recognize their value for building innovative enterprise applications based on formalized business knowledge. Constantly growing number of companies that offer services or tools in this area reflect this trend (cf. [3]). One of its value drivers is the use of ontologies and rules to enable semantic interoperability of heterogeneous data and information ([3], p. 10). First projects have shown the relevance of these technologies for use cases within the automotive industry – in particular to deal with the complexity of the product development process.

"An ontology defines a set of representational primitives with which to model a domain of knowledge or discourse" [4]. These primitives are classes, attributes, and relationships. Using ontologies we can thus create an explicit specification of a simplified view of the world that we wish to represent for some purpose. Whereas database or XML schemata define the structure and integrity of data sets, ontologies enable a data modeling representation that is independent from data structures and their implementations. This not only allows for an easy exchange of data across system boundaries but also for a shift from an IT centric data modeling approach to a business- and user-oriented one.

Due to their expressive power, ontologies are very well suited for describing and formalizing existing structured data sources such as database schemata (cf. [6], [9]). However, they are no sufficient enough to express all semantic relationships between such formal models (e.g., unit conversion, or instance creation, cf. [8]). Rules, which are also part of the semantic web technology stack, provide the required expressiveness, and are thus an important means to overcome with the limitations of ontologies (cf. [5], [1]).

## 3 COMPLEXITY MANAGEMENT WITH ONTOLOGIES

Using ontologies for describing dependencies, we have to take similar preparatory steps as Maurer and Lindemann [7] describe for a DSM-based multi-domain analysis. First, the relevant domains have to be identified. In terms of ontologies, these are the classes. Classes may be structured hierarchically. Different types of relationships can hold between them, so the properties are defined. Based on document analysis and expert interviews, the elements, i.e. the instances of the classes, are identified as well as the relationships between instances of different classes or between instances of the same class. Here, an important feature of ontologies comes into play. The knowledge representation languages used to represent ontologies, such as the W3C standard OWL, are based on formal logic, having thus a well-defined syntax and semantics. Moreover, they provide reasoning capabilities and can be combined with rule engines. This means that dependencies between instances can be inferred by automated reasoning services. Relationships can hence be either explicitly modelled as it is done in traditional DSM/DMM approaches, or be automatically discovered by inferencing on the ontology.

The ontology gathered in this way shows the overall connectivity of the domains, giving the "big picture". Intra- as well as complex inter-domain analysis can be done by computer-based processing of the ontology. The ontology-based approach is thus not restricted to the analysis of a single domain, as DSM, or of two directly connected domains such as DMM. Global analyses encompass for instance navigating through the different classes and their instances by following the relations, or reporting based on queries in the style of SQL (SPARQL). Reasoning and rules also support the analysis. Dependencies can be automatically deduced and made transparent. Clearly, analyses focused on a certain domain, or a pair of domains, are also possible. In contrast to DSM methods where algorithms for analysis and optimization are limited to elements of the same type and a single type of relationship holding between them, in the ontology-based approach, we might analyze different types of relationships simultaneously by defining rules. For example, several relationships describing a certain type of dependency could be considered in the analysis as a single one without changing the modelled data. By exploiting the hierarchical structure, analyses can be done at different levels of detail.

Using a standardized ontology representation language for modelling the domain further fosters the reusability of the captured knowledge. On the one hand, the knowledge captured for the DSM/DMM analysis can be re-used in other applications, e.g. as basis for data integration. On the other hand, knowledge that has been formalized in other related projects can easily be included. Translating the ontology, e.g. represented in OWL which can be serialized in XML, into the format required by DSM/DMM tools allows for applying all analysis and optimization algorithms provided by them.

## 4 INTEGRATING CAX-SYSTEMS WITH SEMANTIC TECHNOLOGIES

The virtual and physical validation of parts and prototypes is an essential step in the product development process of cars. It is supported by a number of computer-aided methodologies – i.e. Design (CAD), Engineering (CAE), and Testing (CAT), which we collectively refer to as CAx. All CAx methods are implemented by appropriate information systems. Their individual focuses within the product development process have led to a great variety of add-on tools that are in use today. As each of them is highly specialized, product models, data structures and technical platforms differ across the CAx tools. This heterogeneous IT infrastructure complicates the exchange of information between different CAx systems which, however, is a prerequisite to flexibly choose the optimal validation approach. Moreover, the actual choice has to take into account the business aspects ranging back to the initial requirements. Finding the optimal combination for a validation need is thus a highly complex task, where we have to deal with business knowledge, IT systems, and in particular, with all the interdependencies between them.

At AUDI AG we have therefore set up a CAx architecture, which uses semantic technologies to capture and integrate various blocks of information influencing the validation process. These include, but are not limited to, user requirements, car functions and properties, business processes, validation methodologies, and CAx services and systems. These information resources as well as the dependencies between single elements thereof have been uniformly represented as ontologies. By using reasoning capabilities, we can now make complex dependencies between specific elements of the CAx architecture transparent and use visualization technologies to present them to the business user. By analyzing the business processes and the use of specific validation methodologies, we can identify complementary CAx services, and thus related CAx systems. In order to apply them flexibly, all required information has to be exchanged efficiently between them. Moreover, we can identify similar CAx services provided by different CAx systems, which might let us reduce redundancy between CAx systems. These findings can now be used for a business-oriented integration of CAx systems. Also, by using semantic technologies we can provide an integrated view on different data from CAx systems, without the need to actually integrate them in a technical manner.

An important aspect within the project has been the direct involvement of the actual users – i.e. the business experts – in creating, using and maintaining the CAx architecture. We have thus built a number of different graphical and textual representations of the contained knowledge to support the various use cases. These components include a matrix editor for creating and visualizing dependencies between two information resources.

Future plans include the extension of the CAx architecture with additional knowledge that enables the automated comparison of validation methodologies depending on the requirements defined for the development process of a specific product.

## 5 DISCUSSION

Ontologies and rules offer the formal expressiveness to represent different kinds of knowledge relevant to complexity management in a uniform and exchangeable manner. Moreover, ontologies enable to create a shared understanding of a domain that can be used by different people. Dependencies between instances of classes within ontologies can be either modelled directly or be discovered by means of logical inference. The gathered knowledge base can be used for building innovative applications utilizing its contents. Different graphical representations going beyond matrix views could help to visualize and navigate within complex domains. Collaboration approaches enable the integration of business experts to maintain these formal models and keep the information stored in our dependency model up to date. Altogether, the application of semantic technologies as a technological foundation for DSM could allow for a more sophisticated use of the gathered knowledge. Most importantly, we can enable the reuse of this knowledge base for other use cases.

## REFERENCES

[1] Angele, J. and Gesmann, M. The Information Integrator: Using Semantic Technology to provide a single view to distributed data. *Datenbanksysteme in Business, Technologie und Web (BTW 2007)*, 2007.

[2] Danilovic M. and Browning T. R. Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Project Management, Vol. 25*, 2007, pp. 300–314.

[3] Davis, M. Semantic Wave 2008 Report - Industry Roadmap to Web 3.0 & Multibillion Dollar Market Opportunities, 2008.

[4] Gruber, T. Ontology. *Encyclopedia of Database Systems*, 2008.

[5] Jones, D. H. and Uschold, M. Role of Rules and Rule Interchange in Semantic Integration & Interoperability. *Rule Languages for Interoperability, W3C*, 2005

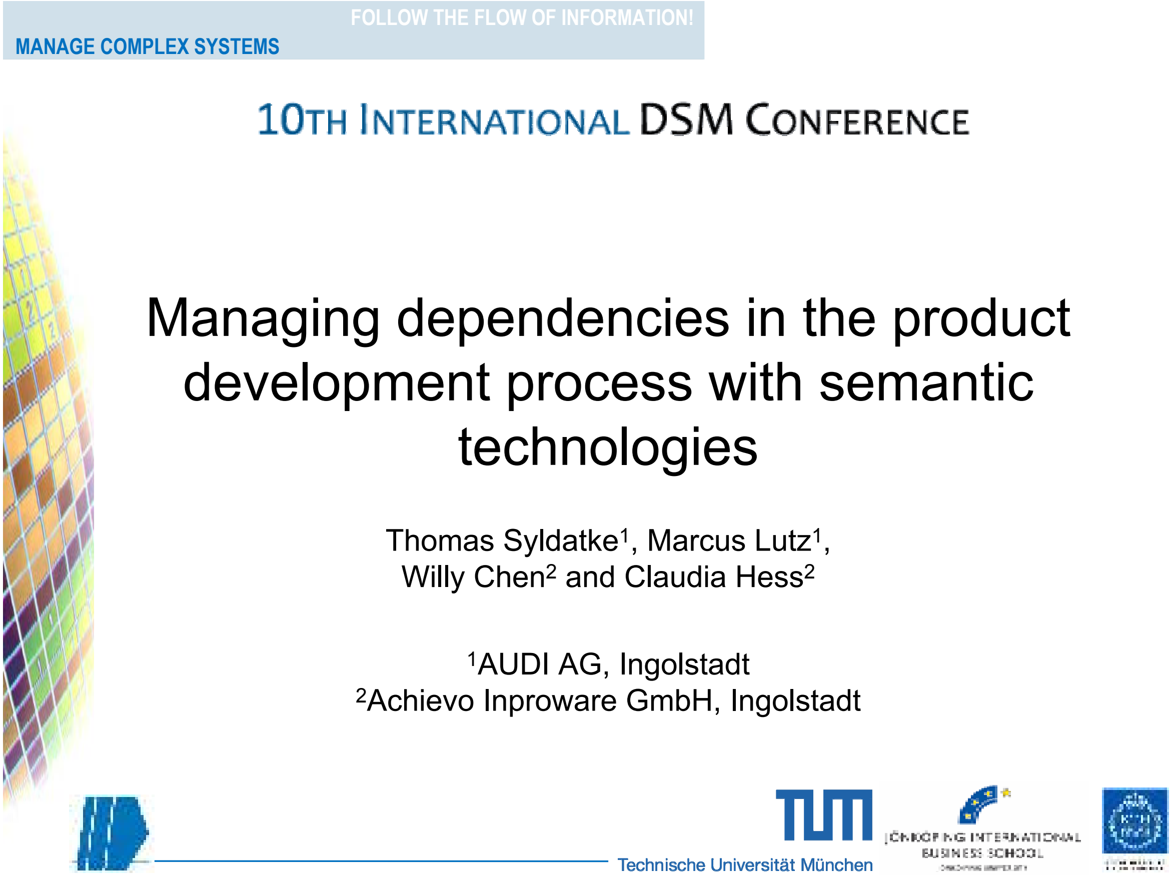
[6] de Laborda, C. P. and Conrad, S. Relational.OWL - A Data and Schema Representation Format Based on OWL Conceptual Modelling, *Proceedings of the Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005)*, 2005.

[7] Maurer, M. and Lindemann, U. Facing Multi-Domain Complexity in Product Development. *CiDaD Working Paper Series, Vol. 03 (1)*, 2007

[8] Stuckenschmidt, H. and Uschold, M. Representation of Semantic Mappings, *Semantic Interoperability and Integration*, 2005.

[9] Trinh, Q. and Barker, K. and Alhajj, R. RDB2ONT: A Tool for Generating OWL Ontologies from Relational Database Systems. *AICT/ICIW, IEEE Computer Society*, 2006.

Contact: Willy Chen
Achievo Inproware GmbH
Innovative Technologies
Ridlerstr. 75
80339 München
Germany
Phone: +49 89 500 597 165
Fax: +49 89 500 597 110
e-mail: willy.chen@achievo.de
URL: http://www.achievo.de



FOLLOW THE FLOW OF INFORMATION!

MANAGE COMPLEX SYSTEMS

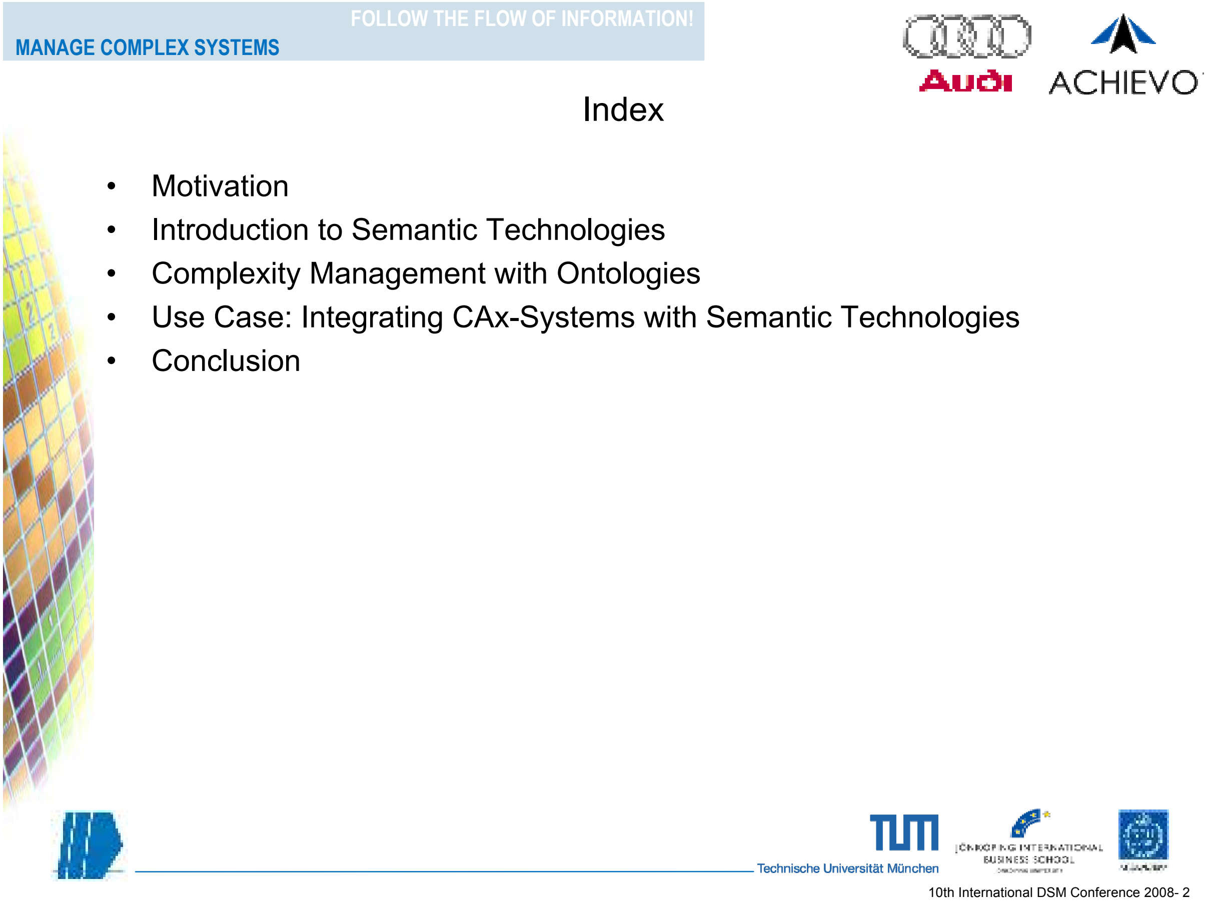
10TH INTERNATIONAL DSM CONFERENCE

# Managing dependencies in the product development process with semantic technologies

Thomas Syldatke[1], Marcus Lutz[1],
Willy Chen[2] and Claudia Hess[2]

[1]AUDI AG, Ingolstadt
[2]Achievo Inproware GmbH, Ingolstadt

Technische Universität München



FOLLOW THE FLOW OF INFORMATION!

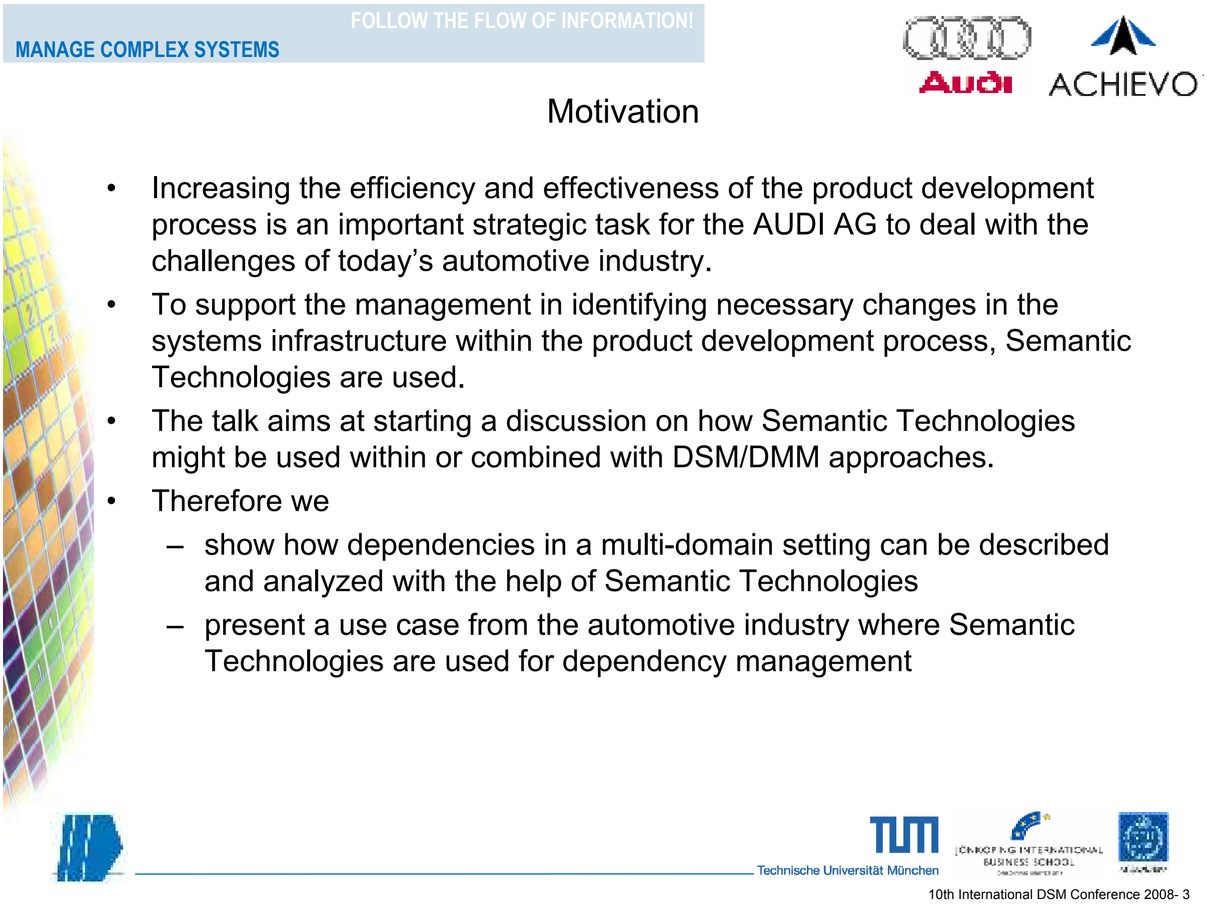
MANAGE COMPLEX SYSTEMS

## Index

- Motivation
- Introduction to Semantic Technologies
- Complexity Management with Ontologies
- Use Case: Integrating CAx-Systems with Semantic Technologies
- Conclusion

Technische Universität München

10th International DSM Conference 2008- 2



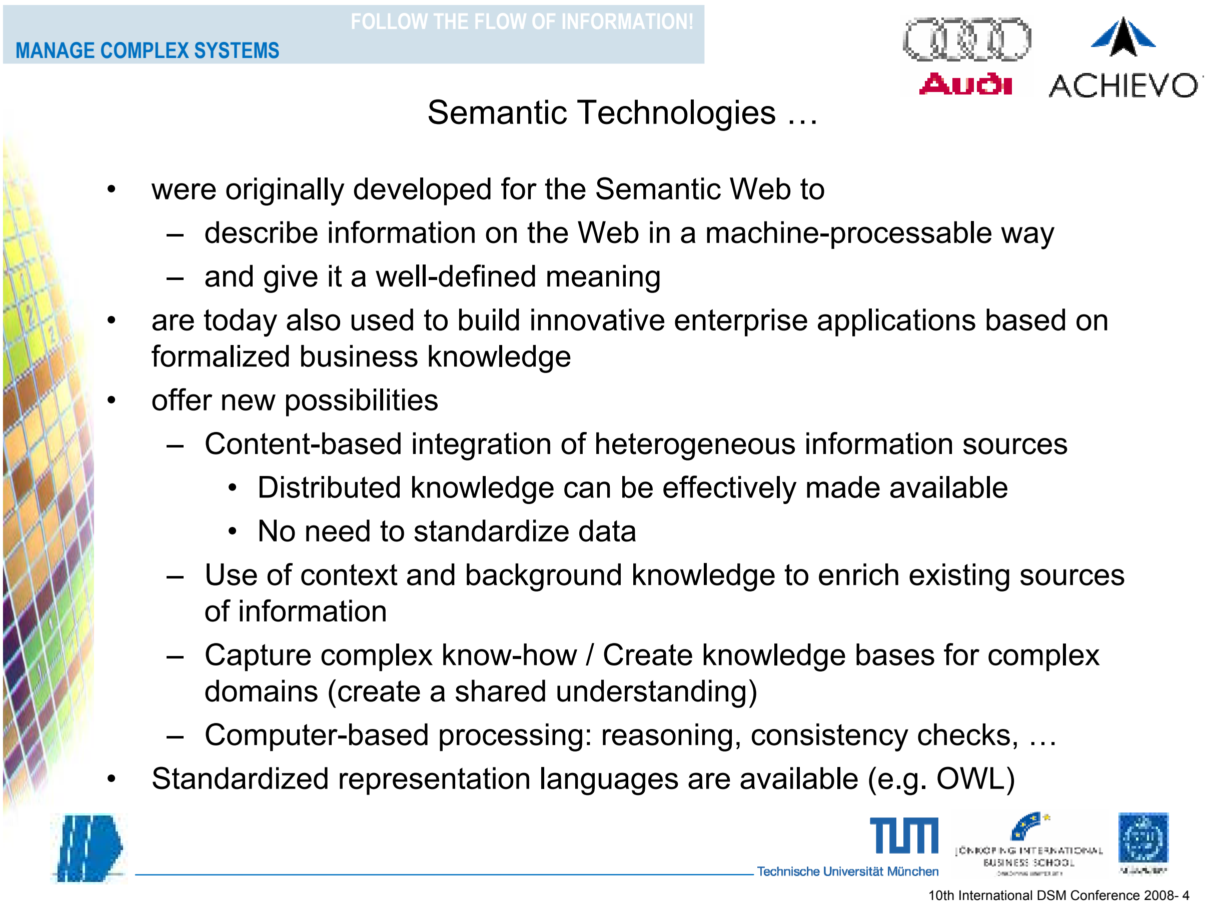
FOLLOW THE FLOW OF INFORMATION!

MANAGE COMPLEX SYSTEMS

## Motivation

- Increasing the efficiency and effectiveness of the product development process is an important strategic task for the AUDI AG to deal with the challenges of today's automotive industry.
- To support the management in identifying necessary changes in the systems infrastructure within the product development process, Semantic Technologies are used.
- The talk aims at starting a discussion on how Semantic Technologies might be used within or combined with DSM/DMM approaches.
- Therefore we
  - show how dependencies in a multi-domain setting can be described and analyzed with the help of Semantic Technologies
  - present a use case from the automotive industry where Semantic Technologies are used for dependency management

Technische Universität München



FOLLOW THE FLOW OF INFORMATION!

MANAGE COMPLEX SYSTEMS

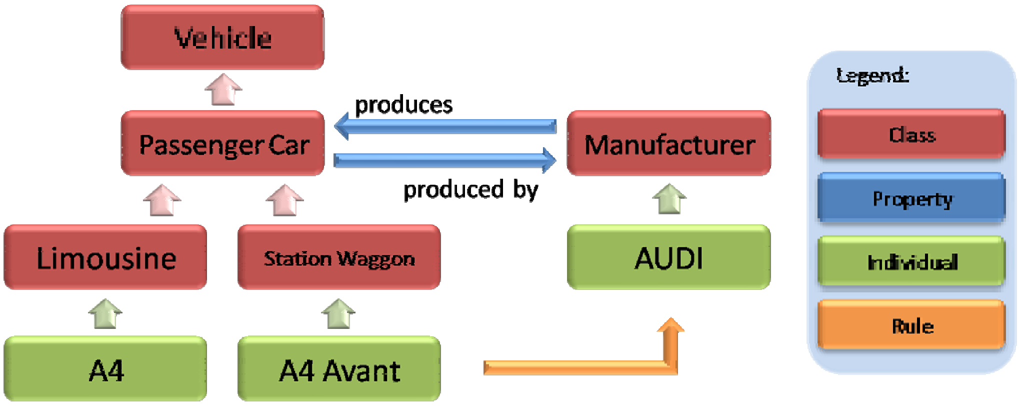
## Semantic Technologies …

- were originally developed for the Semantic Web to
  - describe information on the Web in a machine-processable way
  - and give it a well-defined meaning
- are today also used to build innovative enterprise applications based on formalized business knowledge
- offer new possibilities
  - Content-based integration of heterogeneous information sources
    - Distributed knowledge can be effectively made available
    - No need to standardize data
  - Use of context and background knowledge to enrich existing sources of information
  - Capture complex know-how / Create knowledge bases for complex domains (create a shared understanding)
  - Computer-based processing: reasoning, consistency checks, …
- Standardized representation languages are available (e.g. OWL)

Technische Universität München

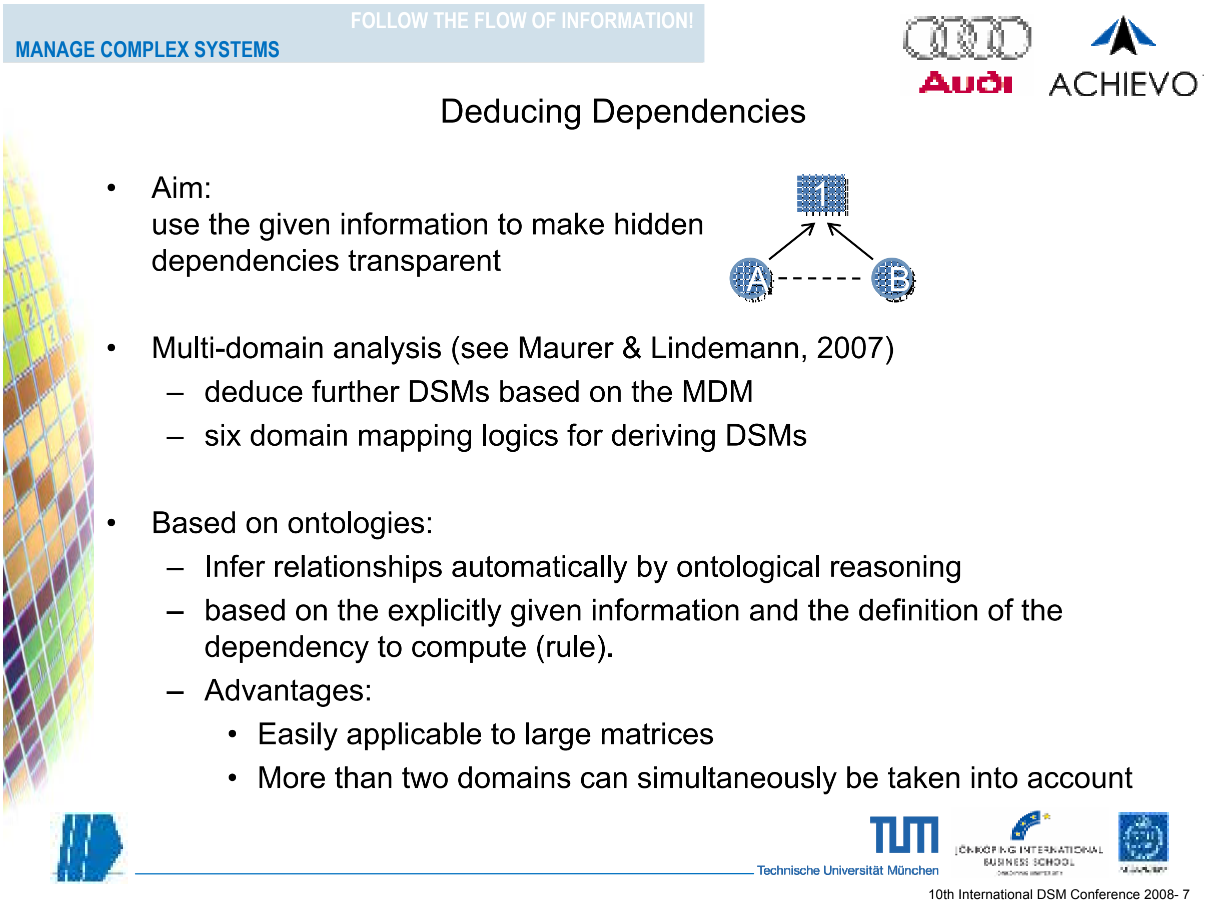
## Ontology

- Definition: An ontology is a formally defined set of concepts and relationships between those concepts for describing complex domains of interest.
- Ontology = Semantic + Structure +  Logic



## Complexity Management with Ontologies

- Approach: Modelling the domains of interest and their dependencies as ontologies
  - Identify the relevant domains, i.e. ontology classes
  - Identify the relationships that may hold
    - between elements of the same domain or
    - between elements of different domains
  - Identify elements, i.e. the individuals of the classes
  - Identify and annotate relationships between
    - Individuals of the same class
    - Individuals of different classes
- The resulting ontologies show the overall connectivity of the domains
- The ontologies can be represented as a multi-domain matrix (MDM) combining inter- and intra-domain matrices
  - DSM and DMM techniques can be applied to these matrices



MANAGE COMPLEX SYSTEMS

FOLLOW THE FLOW OF INFORMATION!

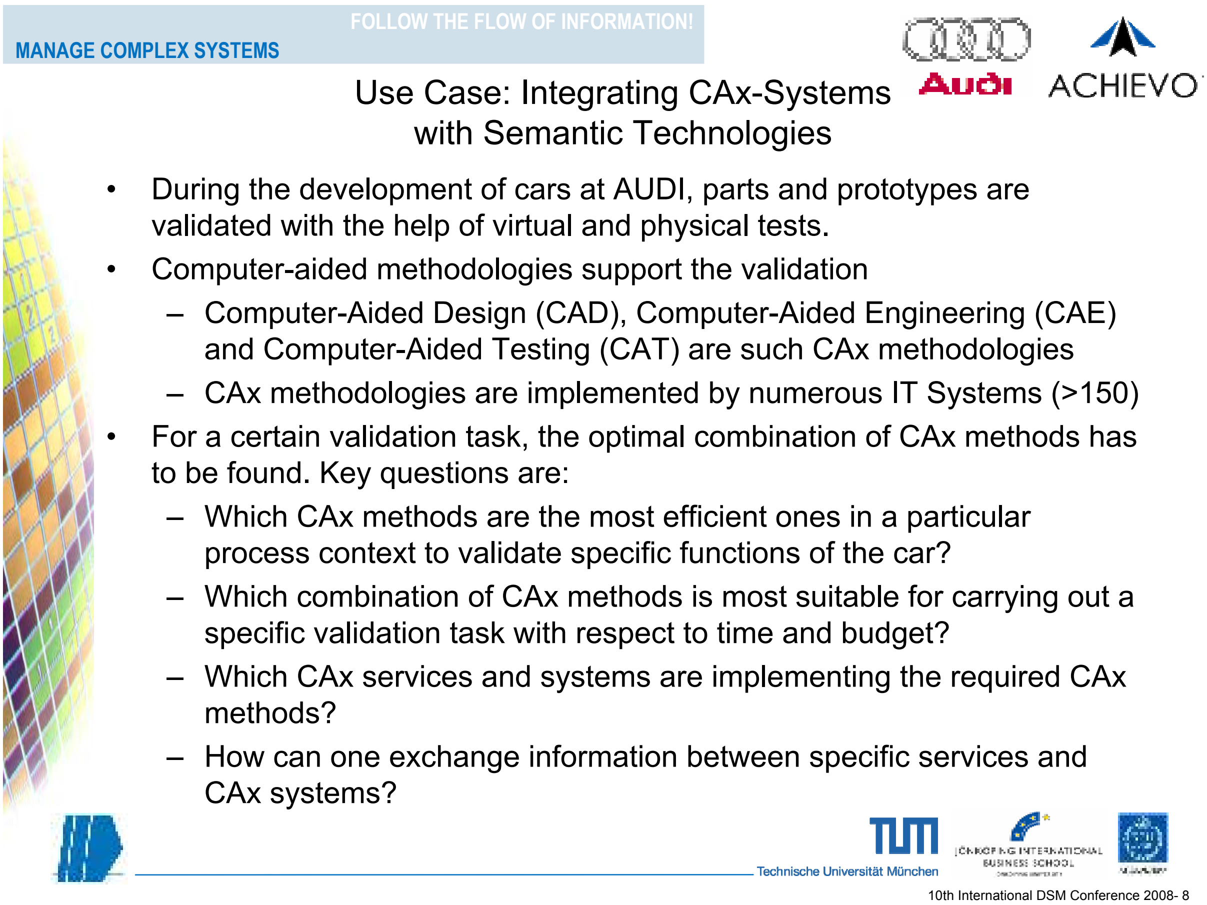
## Deducing Dependencies

- Aim:
  use the given information to make hidden dependencies transparent

- Multi-domain analysis (see Maurer & Lindemann, 2007)
  - deduce further DSMs based on the MDM
  - six domain mapping logics for deriving DSMs
- Based on ontologies:
  - Infer relationships automatically by ontological reasoning
  - based on the explicitly given information and the definition of the dependency to compute (rule).
  - Advantages:
    - Easily applicable to large matrices
    - More than two domains can simultaneously be taken into account

Technische Universität München

10th International DSM Conference 2008- 7



MANAGE COMPLEX SYSTEMS

FOLLOW THE FLOW OF INFORMATION!

## Use Case: Integrating CAx-Systems with Semantic Technologies

- During the development of cars at AUDI, parts and prototypes are validated with the help of virtual and physical tests.
- Computer-aided methodologies support the validation
  - Computer-Aided Design (CAD), Computer-Aided Engineering (CAE) and Computer-Aided Testing (CAT) are such CAx methodologies
  - CAx methodologies are implemented by numerous IT Systems (>150)
- For a certain validation task, the optimal combination of CAx methods has to be found. Key questions are:
  - Which CAx methods are the most efficient ones in a particular process context to validate specific functions of the car?
  - Which combination of CAx methods is most suitable for carrying out a specific validation task with respect to time and budget?
  - Which CAx services and systems are implementing the required CAx methods?
  - How can one exchange information between specific services and CAx systems?

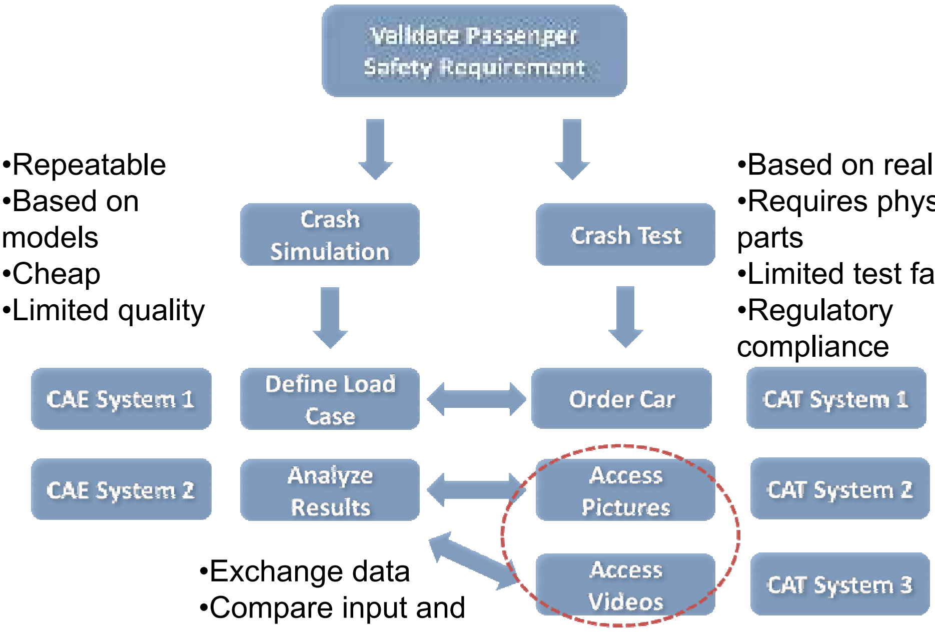
Technische Universität München

10th International DSM Conference 2008- 8

MANAGE COMPLEX SYSTEMS

FOLLOW THE FLOW OF INFORMATION!

## Interplay between CAx Methodologies



•Repeatable
•Based on models
•Cheap
•Limited quality

•Based on real cars
•Requires physical parts
•Limited test facilities
•Regulatory compliance

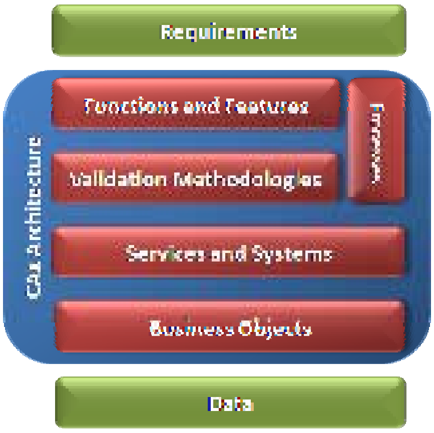
•Exchange data
•Compare input and results
•Propose tool integration

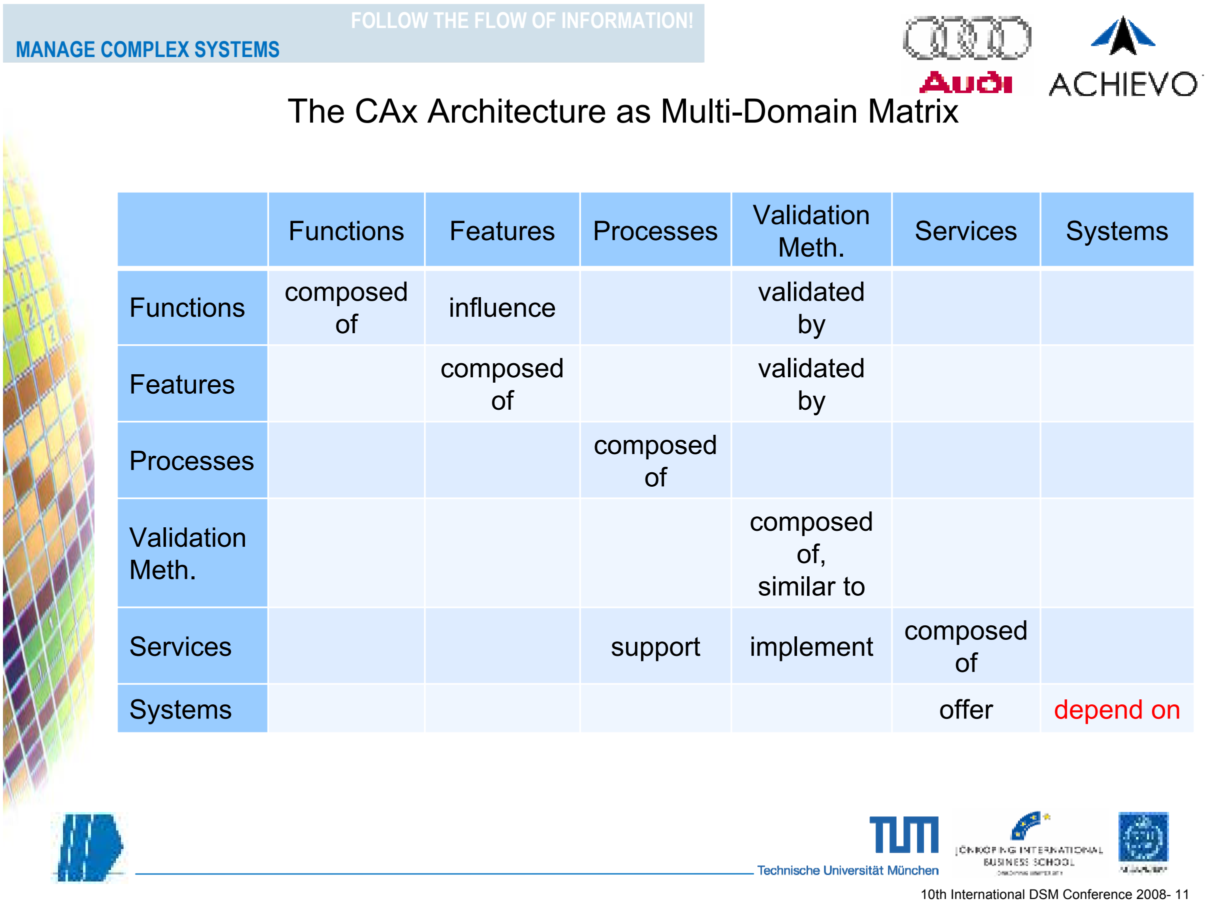
MANAGE COMPLEX SYSTEMS

FOLLOW THE FLOW OF INFORMATION!

## Our Approach: The CAx Architecture

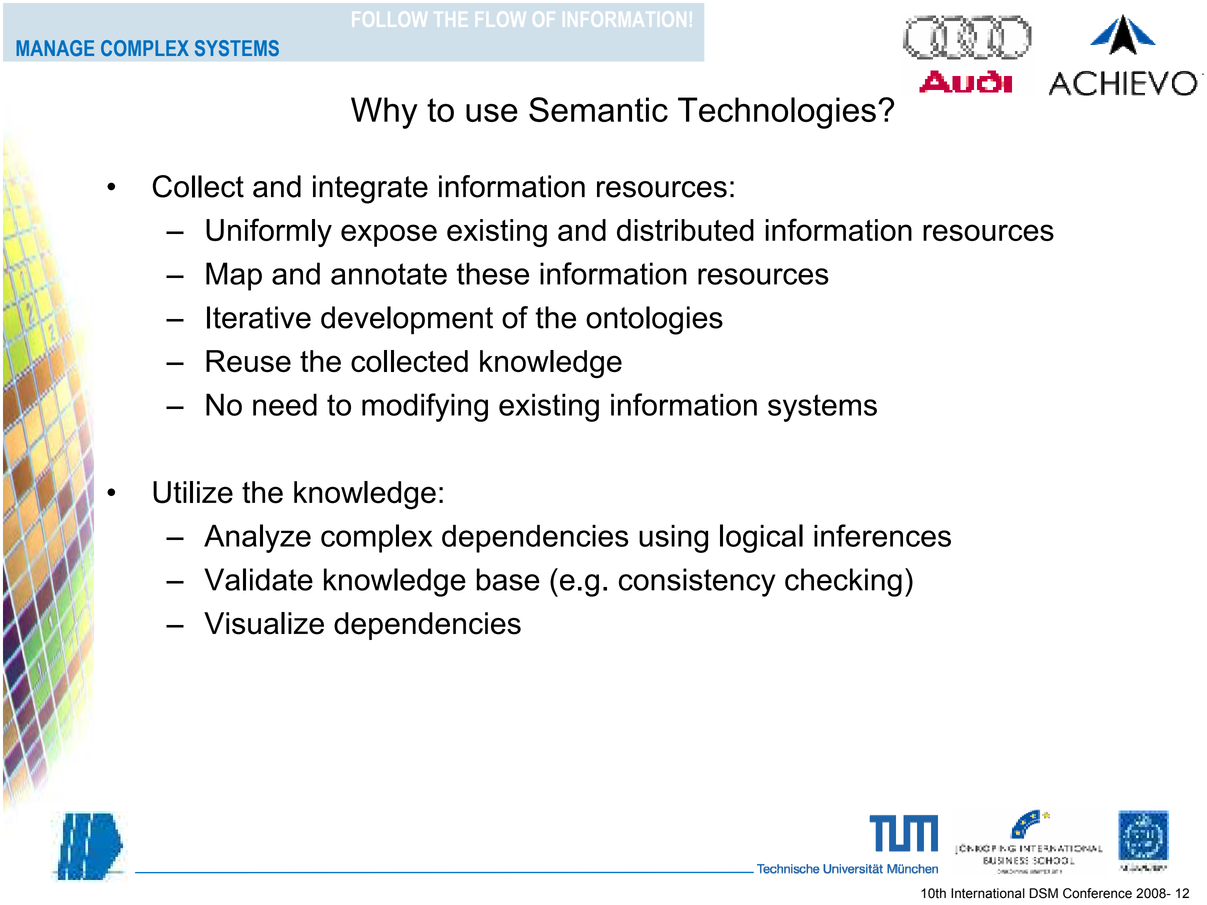
- Identify relevant information resources within AUDI
- Define classes and properties for describing the information resources on the different layers
- Connect related information resources
  - e.g. car functions and features
- Infer dependencies by using reasoning





## The CAx Architecture as Multi-Domain Matrix

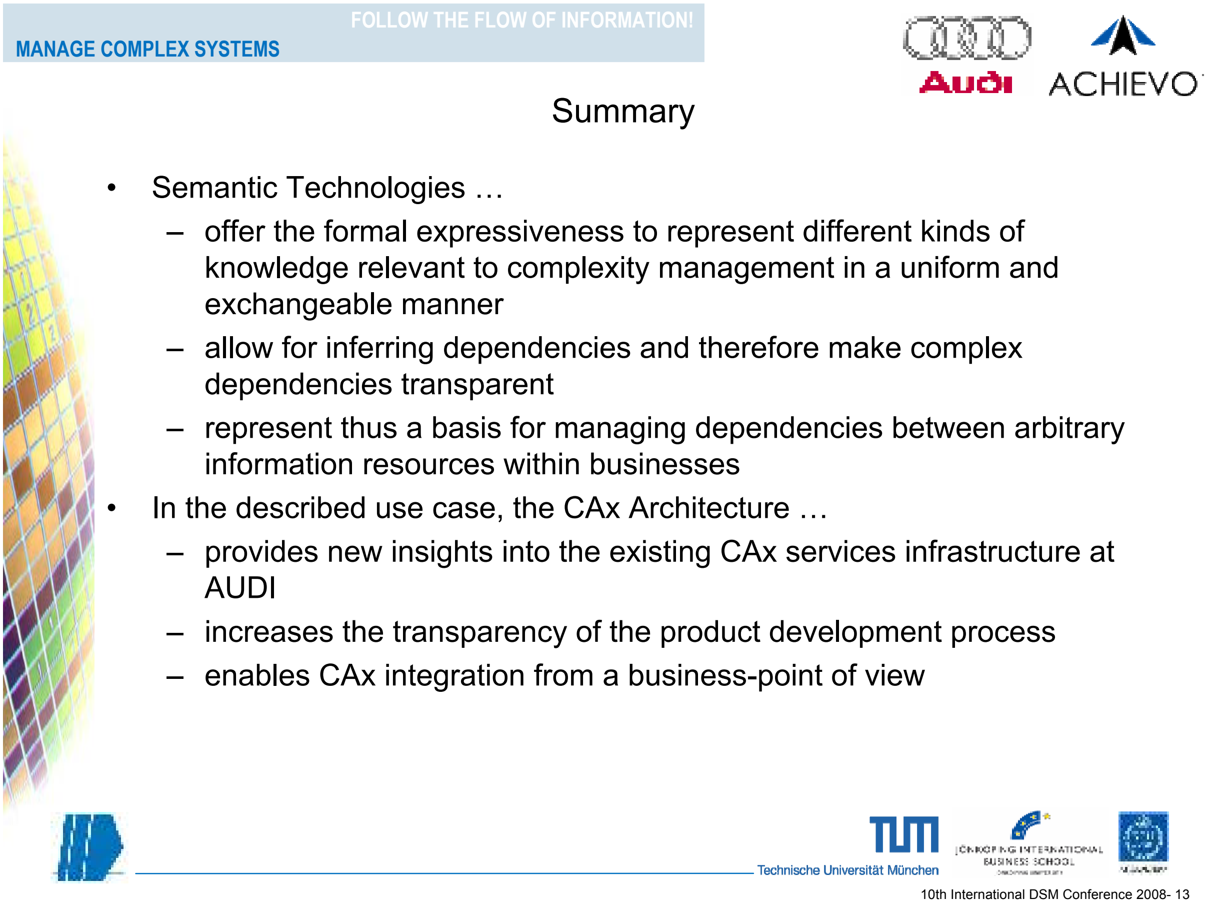
| | Functions | Features | Processes | Validation Meth. | Services | Systems |
|---|---|---|---|---|---|---|
| Functions | composed of | influence | | validated by | | |
| Features | | composed of | | validated by | | |
| Processes | | | composed of | | | |
| Validation Meth. | | | | composed of,<br>similar to | | |
| Services | | | support | implement | composed of | |
| Systems | | | | | offer | depend on |



## Why to use Semantic Technologies?

- Collect and integrate information resources:
  - Uniformly expose existing and distributed information resources
  - Map and annotate these information resources
  - Iterative development of the ontologies
  - Reuse the collected knowledge
  - No need to modifying existing information systems

- Utilize the knowledge:
  - Analyze complex dependencies using logical inferences
  - Validate knowledge base (e.g. consistency checking)
  - Visualize dependencies



FOLLOW THE FLOW OF INFORMATION!

MANAGE COMPLEX SYSTEMS

## Summary

- Semantic Technologies …
  - offer the formal expressiveness to represent different kinds of knowledge relevant to complexity management in a uniform and exchangeable manner
  - allow for inferring dependencies and therefore make complex dependencies transparent
  - represent thus a basis for managing dependencies between arbitrary information resources within businesses
- In the described use case, the CAx Architecture …
  - provides new insights into the existing CAx services infrastructure at AUDI
  - increases the transparency of the product development process
  - enables CAx integration from a business-point of view

Technische Universität München

10th International DSM Conference 2008- 13