Solutions and Challenges for Semantics-Enabled Software Engineering

Rudi Studer & Daniel Oberle

Institute AIFB, University of Karlsruhe &
FZI Research Center for Information Technologies &
Ontoprise GmbH

Keynote
SWESE Workshop @ ISWC 2005

Galway, November 6, 2005

Agenda

- Introduction
- Current Approaches
  - Ontologies in Autonomic Computing Systems
  - Semantic Management of Middleware
- New Developments
  - SAP Enterprise Services Architectures (ESA)
  - Semantic Web Services
  - Component-based Application Development
  - Ontology Definition Metamodel
- Conclusion & Outlook

Introduction

Relevant topics/targets for Semantics-Enabled Software Engineering

- Requirements Engineering
  - ontologies as more expressive domain models
- Model Driven Architectures
  - reasoning for consistency checking, transformation, etc.
- Component- and Service-Oriented Architectures
  - reasoning for discovery, composition, invocation, etc.
- Autonomous Computing
  - reasoning for self-management of software systems

Introduction

Semantics-Enabled Software Engineering - been there, done that?

- Goal-Driven Requirements Engineering
  - since the early 1990s
  - domain models are designed as part of a software architecture
  - meta-models are not ontology/logic-based

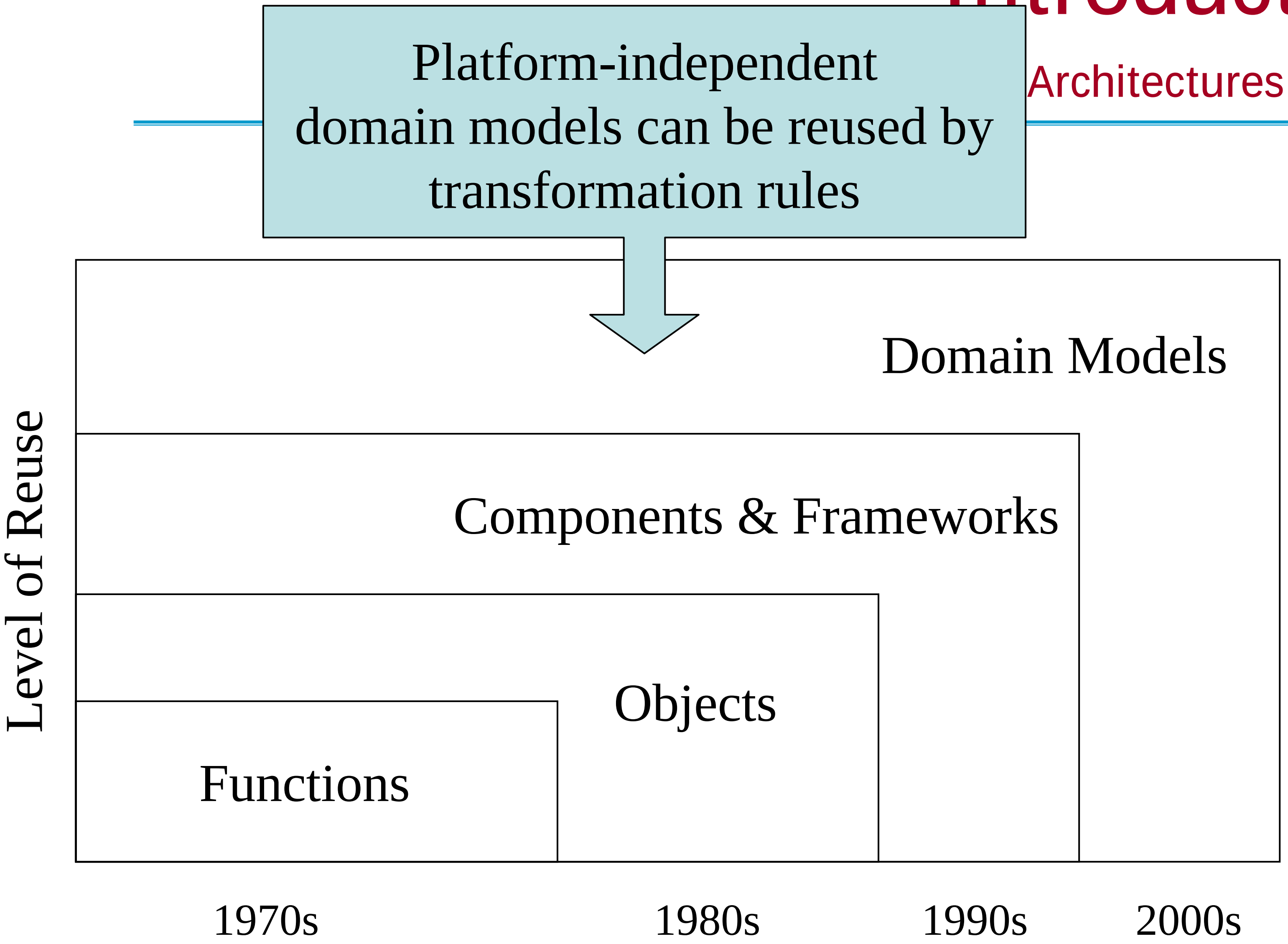
[Pohl 1999]

- Faceted Software Classification
  - describes software components by keywords
  - keywords are organized in facets
  - goal: facilitate reuse

[Pietro-Diaz 1991]

- Knowledge-based Software Engineering
  - long established field of research
  - main conference SEKE in its 17th year
  - relevant topics:
    - AI Approaches to Software Engineering
    - Automated Reasoning/Software Design
    - Knowledge Representation, Retrieval, Visualization

[SEKE 2005]



[Mellor et al., 2004]

Agenda

- Introduction
- **Current Approaches**
  - Ontologies in Autonomic Computing Systems
  - Semantic Management of Middleware
- New Developments
  - SAP Enterprise Services Architectures (ESA)
  - Semantic Web Services
  - Component-based Application Development
  - Ontology Definition Metamodel
- Conclusion & Outlook

Current Approaches (2)

Ontologies in Autonomic Computing Systems

- Goal: improved self-management capabilities covering
  - self-healing, self-protecting, self-optimizing, and self-configuring

- Ontologies as core components
  - for automated analysis of enterprise-wide event data
  - based on user-defined rules
  - to trigger corrective actions for healing the system
  - to deal with policy based goals on a higher abstraction level
  - to provide new levels of functionality
    - explanation
    - ranking
    - gap analysis

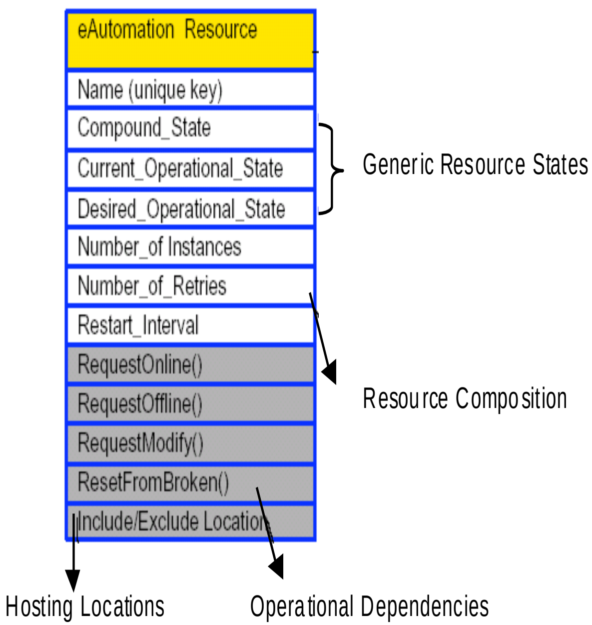
[Stojanovic et al. 2004]

Current Approaches (2)

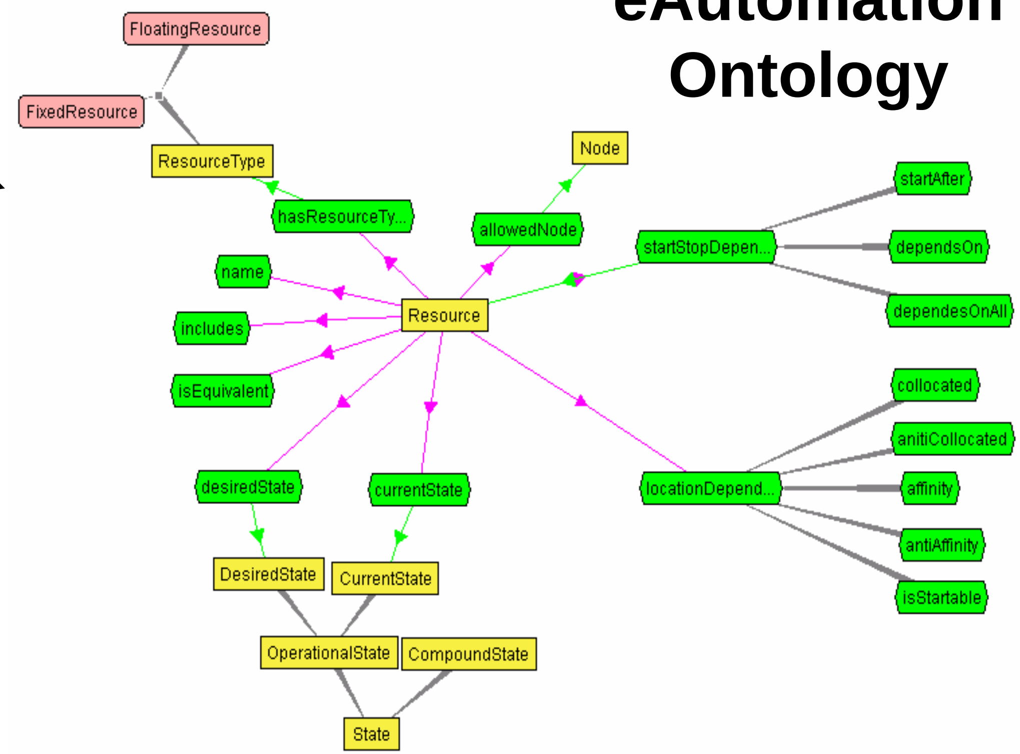
An Example: Ontologies in IBM's Autonomic Computing Systems

Richer structure

eAutomation Resource Model



eAutomation Ontology



Rules

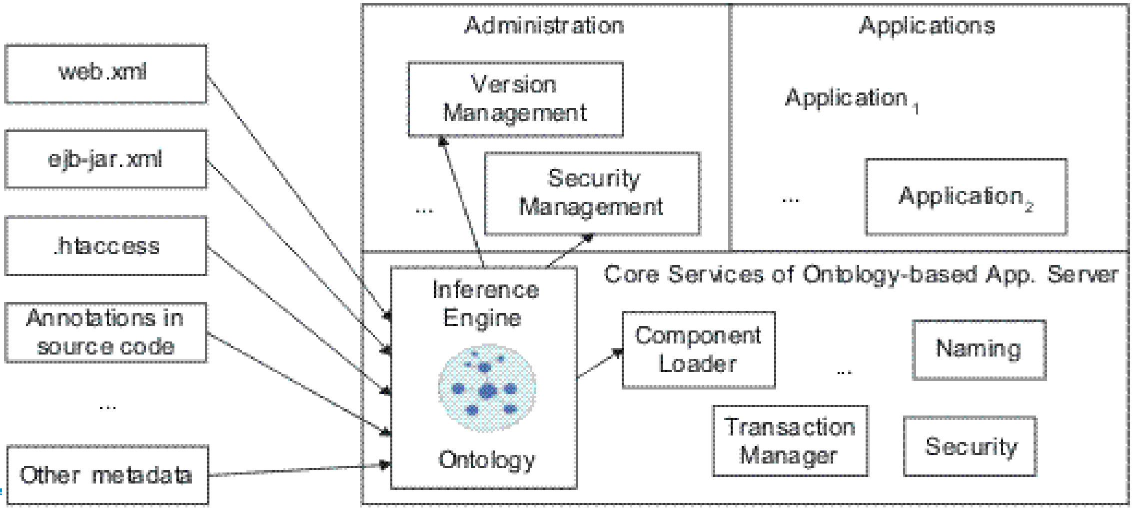
If the resource "A" should start after the resource "B" and the resource "B" is online, Then resource "A" should be online

Current Approaches (3)

Semantic Management of Middleware (KAON Server)

- Application servers are very complex software products
- so far they are managed with admin tools and XML configuration files
  - disadvantage: conceptual model of configuration files only implicit
  - hence, they are difficult to retrieve, survey, check for validity and maintain.

- **Contribution**:
  - ontology-based approach to support development and administration of application server.
  - Ontological descriptions may be queried, may foresight required actions, or may be checked to avoid inconsistent system configurations.

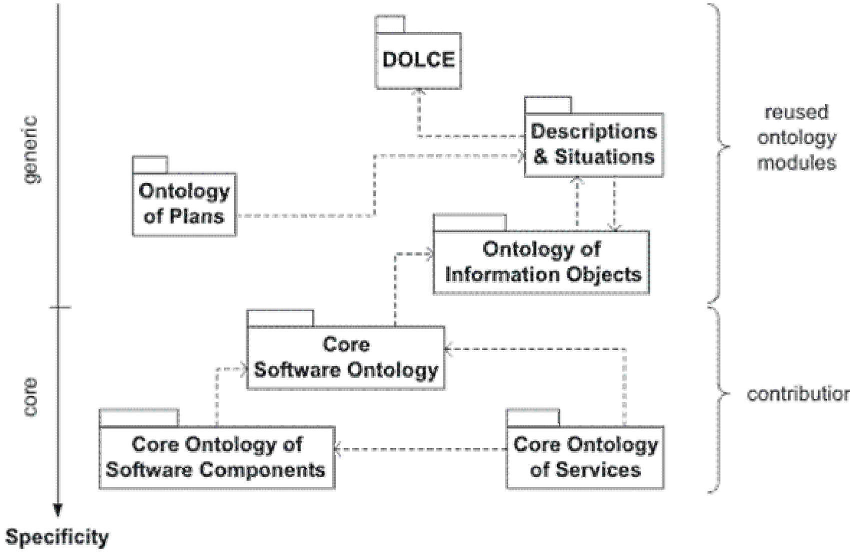


[Oberle et al. 2004]

Current Approaches (3)

Semantic Management of Middleware (KAON Server)

- Usage of foundational ontology allows
  - disambiguation of terms
  - improved design
  - high axiomatization



- Example: Software Component

$$
\begin{aligned}
&\text{(D30)} \quad \mathsf{SoftwareComponent}(x) =_{def} \\
&\qquad \mathsf{CSO{:}Class}(x) \wedge \exists y(\mathsf{conforms}(x,y) \wedge \mathsf{FrameworkSpecification}(y)) \\
&\text{(D29)} \quad \mathsf{conforms}(x,y) =_{def} \mathsf{CSO{:}Class}(x) \wedge \mathsf{FrameworkSpecification}(y) \wedge \\
&\qquad \exists i,c(\mathsf{CSO{:}Interface}(i) \wedge \mathsf{DOLCE{:}member}(c,i) \wedge \\
&\qquad \mathsf{DOLCE{:}Collection}(c) \wedge \mathsf{DnS{:}unifies}(y,c) \rightarrow \\
&\qquad \mathsf{CSO{:}implements}(x,i)) \\
&\text{(D28)} \quad \mathsf{FrameworkSpecification}(x) =_{def} \\
&\qquad \mathsf{OoP{:}Plan}(x) \wedge \exists y(\mathsf{DOLCE{:}Collection}(y) \wedge \mathsf{DnS{:}unifies}(x,y) \wedge \\
&\qquad \forall z(\mathsf{DOLCE{:}member}(y,z) \rightarrow \mathsf{CSO{:}Interface}(z)))
\end{aligned}
$$

[Oberle et al. 2004]

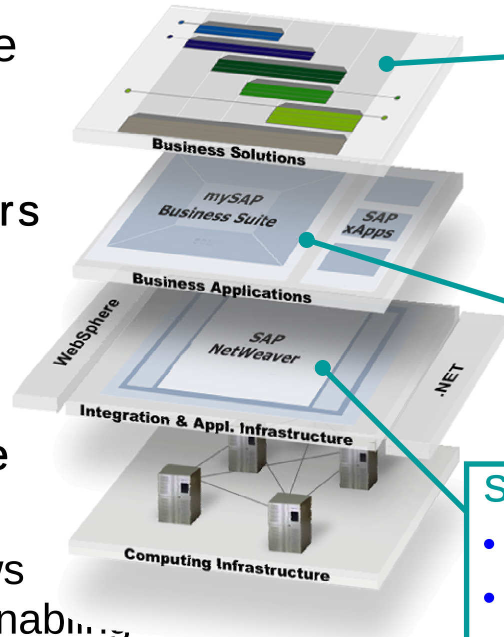
Agenda

- Introduction
- Current Approaches
  - Ontologies in Autonomic Computing Systems
  - Semantic Management of Middleware
- New Developments
  - SAP Enterprise Services Architectures (ESA)
  - Semantic Web Services
  - Component-based Application Development
  - Ontology Definition Metamodel
- Conclusion & Outlook

New Developments (1)

SAP Enterprise Services Architectures (ESA)

- Future Business Landscape
  - **Ecosystem** of individual service **providers & requesters**
- Advantages
  - Rigorous decoupling for improved **maintainance** and **documentation**
  - Rigorous decoupling allows **flexible business** by enabling exchange of business modules



Business Solutions
- Manager's View Layer
- Documentation
- Business Scenarios

Business Process Platform
- Business View Layer
- mySAP Business Suite
- Composite Applications

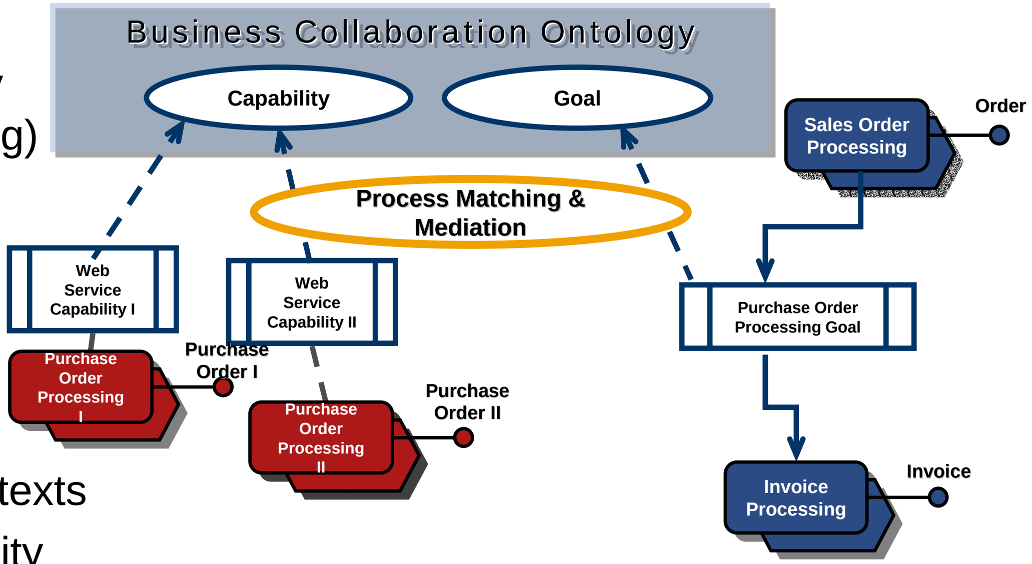
SAP NetWeaver
- Technical Layer
- Acts as Application Server
- Allows Manual Integration

- Challenges
  - Up to now, no formal description of processes, therefore still **manual integration**
  - Also, there is no way to ensure that configuration is **consistent**

New Developments (1)

SAP Enterprise Services Architectures (ESA)

- Vision
  - Business Process Flexibility (e.g., to simplify out-sourcing)
  - SAP‘s ESA



- Technical Requirements
  - Defining **placeholders** for process steps
  - Matching of **business** contexts
  - Describing **data** compatibility
  - Alignment with **surrounding process** steps

- Advantages:
  - Automatic **discovery** of suitable services by capturing business semantics
  - Automatic **integration** of new services by capturing behavioural semantics
  - **Self-adjusting business** using goals as placeholders for appropriate services

New Developments (2)
Semantic Web Services

- Effort of the AI/Semantic Web community
- Goal: full automation of all Web Service management tasks
  - discovery
  - composition
  - invocation
  - orchestration
- First applications being realized
- Examples: OWL-S, WSMO, Meteor-S, etc.

[Martin et al. 2004, Fensel et al. 2002, Patil et al. 2004]

AIFB

New Developments (3)

Component-based Application Development

- Goal: Enable SMEs to develop software in a more collaborative, component-based way
- How can semantic technologies help to accomplish this?
- Example use cases
  - Which component fulfills similar functionality?
  - License of component X?
  - Who can help me to modify this component?
- Currently approached in a national project

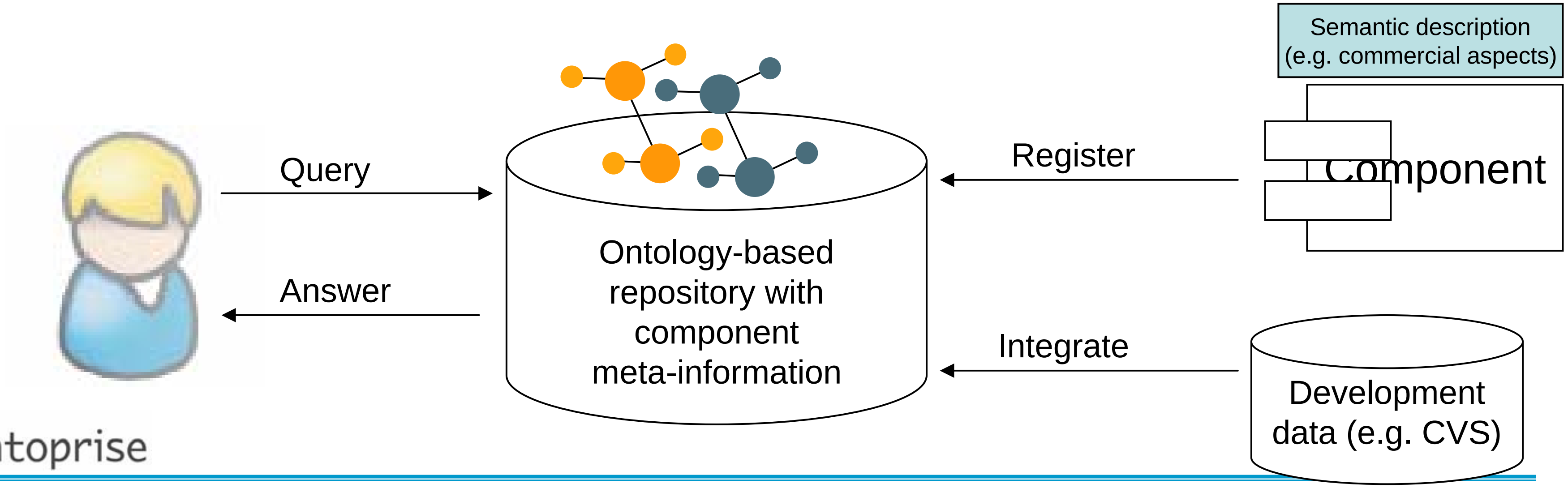
`http://www.collabawue.de/` (in German)

FZI

ontoprise

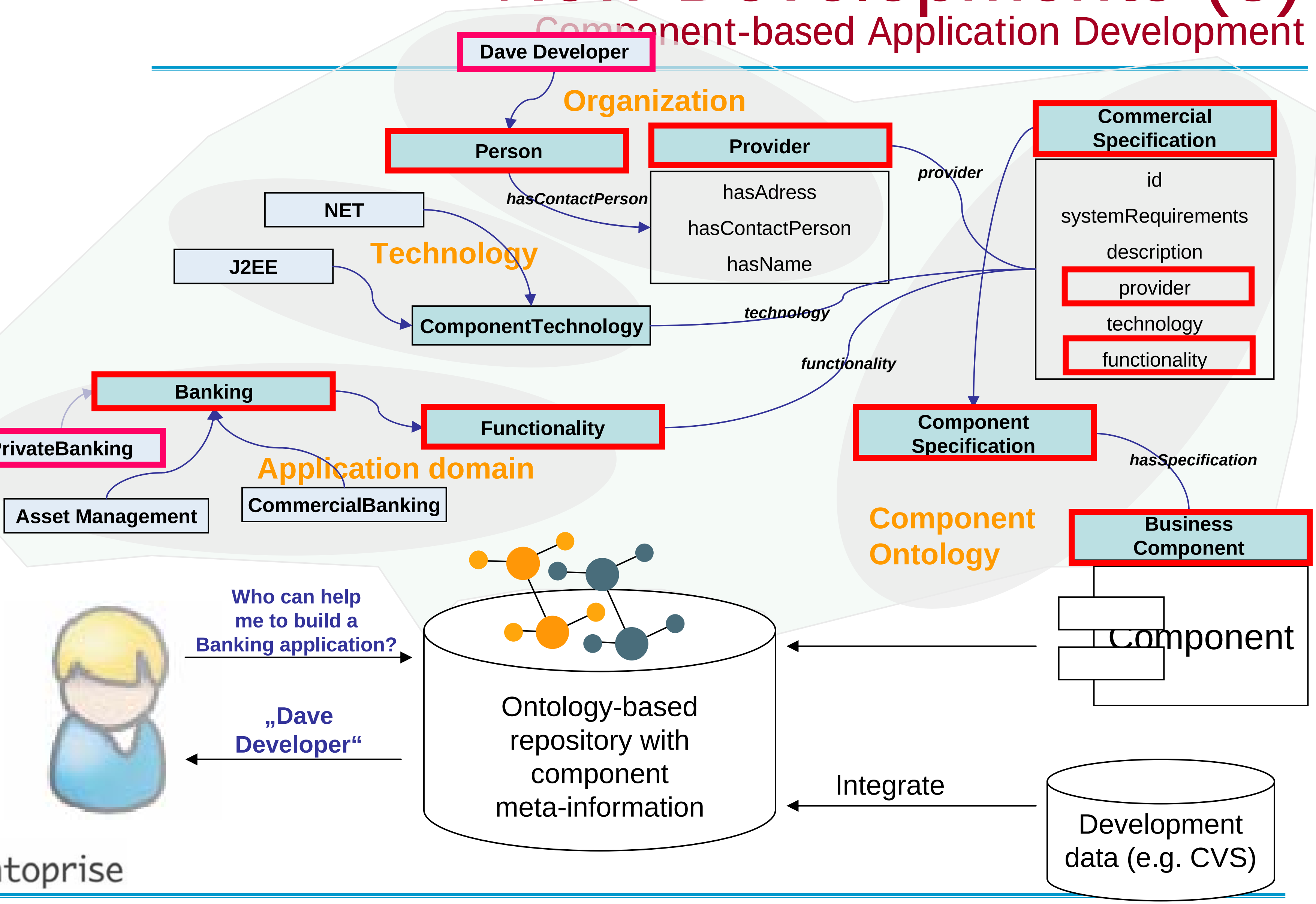
New Developments (3)
Component-based Application Development

- Vision: „Semantic components in an intelligent infrastructure“
  - integration of software engineering *and* knowledge management aspects
  - make metadata machine readable
  - integrate available knowledge about software artifacts



New Developments (3)

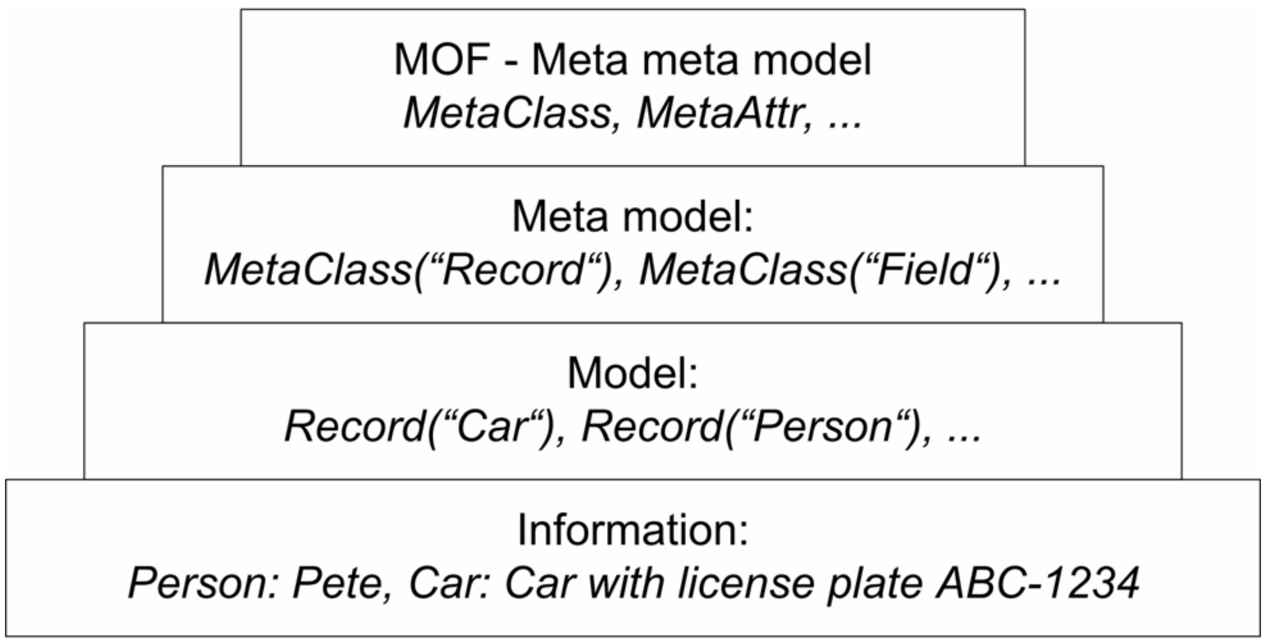
Component-based Application Development



New Developments (4)

Ontology Definition Metamodel

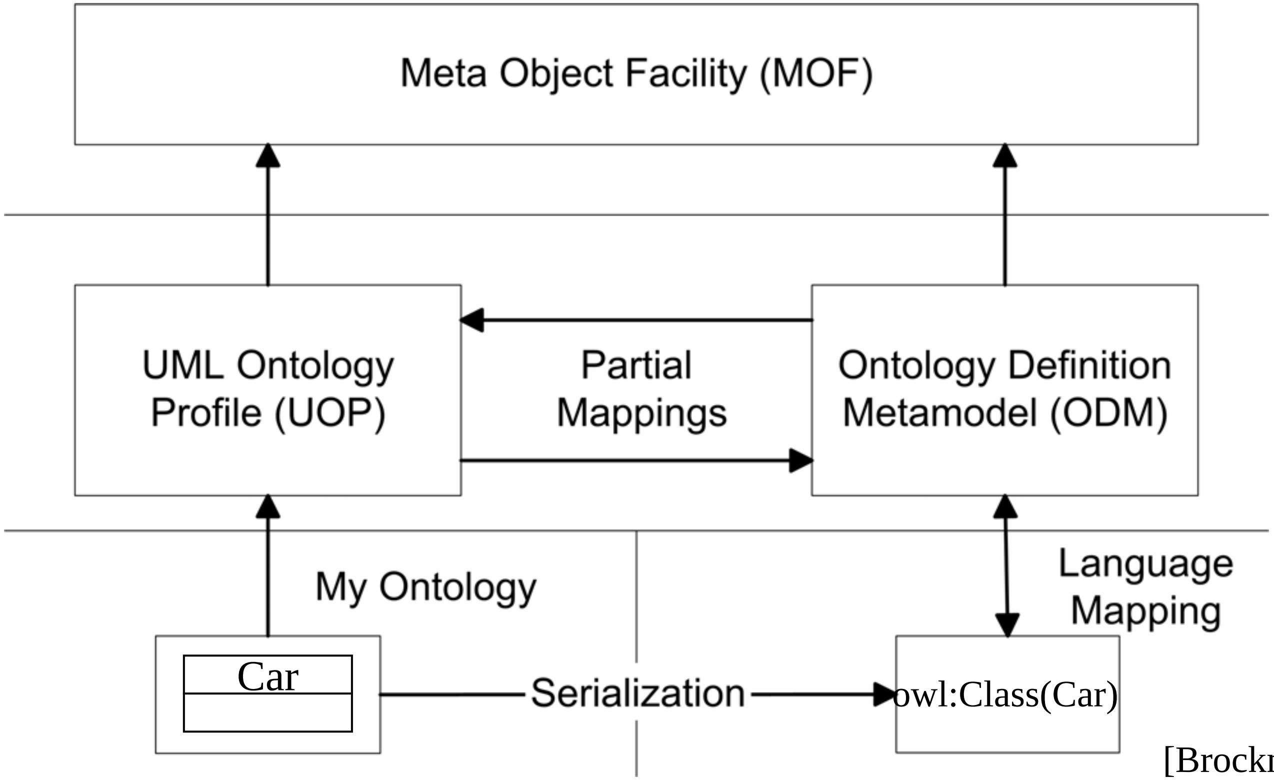
- MOF allows to define modeling languages and forms the core of OMG standards
- Definition of a „record“:



New Developments (4)

Ontology Definition Metamodel

Ontologies are defined with a UML-based notation (UML Ontology Profile) for a MOF-based data model (Ontology Definition Metamodel)



[Brockmans et al., 2004]

Agenda

- Introduction
- Current Approaches
  - Ontologies in Autonomic Computing Systems
  - Semantic Management of Middleware
- New Developments
  - SAP Enterprise Services Architectures (ESA)
  - Semantic Web Services
  - Component-based Application Development
  - Ontology Definition Metamodel
- Conclusion & Outlook

Conclusion

- Increasing complexity of systems make more intelligent efforts a must
- Older efforts already paved the way
- Added value of Semantic Web technologies:
  - Standardization
  - Integration
  - Web compliance
  - Reasoning
- Tradeoff:
  Modelling efforts have to be justified by savings in other tasks
- To which extent do we need gray-box modelling?

Conclusion

- **W3C Software Engineering Task Force**

  `http://www.w3.org/2001/sw/BestPractices/SE/`

- **W3C Notes:**
  - Ontology Driven Architectures and Potential Uses of the Semantic Web in Software Engineering
  - A Semantic Web Primer for Object-Oriented Software Developers

- **This Workshop!**

Thank You!

For further information and relevant publications see

http://www.aifb.uni-karlsruhe.de/WBS

http://www.fzi.de/ipe

http://www.ontoprise.de

References

- [Brockmans et al. 2004] Saartje Brockmans, Raphael Volz, Andreas Eberhart, Peter Loeffler, *Visual Modeling of OWL DL Ontologies using UML*. In F. van Harmelen, S.A. McIlraith, D.P.(eds), *The Semantic Web ISWC 2004*, pages 198-213, Springer-Verlag.
- [Fensel et al. 2002] Fensel, Dieter and Bussler, Christoph (2002). *The Web Service Modeling Framework WSMF*. Electronic Commerce: Research and Applications, 1:113–137.
- [Jarke et al. 1993] Matthias Jarke, Janis Bubenko, Colette Rolland, Alistair Sutcliffe, Yannis Vassiliou, *Theories Underlying Requirements Engineering: An Overview of NATURE at Genesis*. IEEE Int. Symposium on Requirements Engineering, 1993.
- [Martin et al. 2004] Martin, David, Burstein, Mark, Hobbs, Jerry, Lassila, Ora, McDermott, Drew, McIlraith, Sheila, Narayanan, Srini, Paolucci, Massimo, Parsia, Bijan, Payne, Terry, Sirin, Evren, Srinivasan, Naveen, and Sycara, Katia (2004). *OWL-S: Semantic Markup for Web Services*. http://www.daml.org/services/owl-s/1.1/
- [Mellor et al., 2004], Stephen J. Mellor, Kendall Scott, Axel Uhl, *MDA Distilled*. Addison Wesley. 2004
- [Mili et al. 1998] A. Mili, R. Mili, and R.T. Mittermeir. *A survey of software reuse librarie*s. Annals of Software Engineering, 5:349 – 414, 1998.
- [Oberle et al. 2005] Daniel Oberle, Andreas Eberhart, Steffen Staab, Raphael Volz *Developing and Managing Software Components in an Ontology-based Application Server*. In Hans-Arno Jacobsen, Middleware 2004, ACM/IFIP/USENIX 5th International Middleware Conference, Toronto, Ontario, Canada, volume 3231 of LNCS, pp. 459-478. Springer, 2004.
- [Oberle 2005] Daniel Oberle, *Semantic Management of Middleware*. In A. Sheth, R. Jain (eds.) The Semantic Web and Beyond, vol. I, Springer, Dec 2005
- [Patil et al. 2004] Patil, A., Oundhakar, S., Sheth, A., and Verma, K. (2004). *METEOR-S Web Service Annotation Framework*. In The 13th International World Wide Web Conference Proceedings, pages 553–563. ACM Press.
- [Pietro-Diaz 1991] R. Pietro-Diaz. *Implementing Faceted Classification for Software Reuse*. Communications of the ACM. Special issue on software engineering., 34(5):88 – 97, May 1991.
- [Pohl 1999] Pohl, K.: *Requirements Traceability – Models of the Development World*. In: M. Jarke; C. Rolland; A. Sutcliffe: The NATURE of Requiremtents Engineering. Shaker Verlag, Aachen 1999.
- [SEKE 2005] *17th International Conference on Software Engineering and Knowledge Engineering* http://www.ksi.edu/seke/seke05.html
- [Stojanovic et al. 2004] Ljiljana Stojanovic, Jürgen Schneider, Alexander Maedche, Susanne Libischer, Rudi Studer, Andreas Abecker, Gerd Breiter, John Dinger. *The Role of Ontologies in Autonomic Computing Systems*, IBM Systems Journal Vol. 43 (No. 3). August 2004.