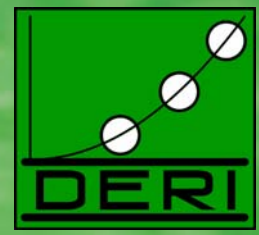


Spinning a Corporate Semantic Web for Product Engineering

Uwe Keller, Alois Reitbauer, Roland Mungenast,
Stijn Heymans, Michael Neswal

Uwe Keller
uwe.keller@deri.org

3rd *International Conference on Web Information Systems and Technologies* (WEBIST)
March 3 – 6, 2007
Barcelona, Spain.

Product engineering today

- Product Engineering (PE) is an ubiquitous task
  - Occurs in nearly all industries
  - Companies develop multiple products at the same time

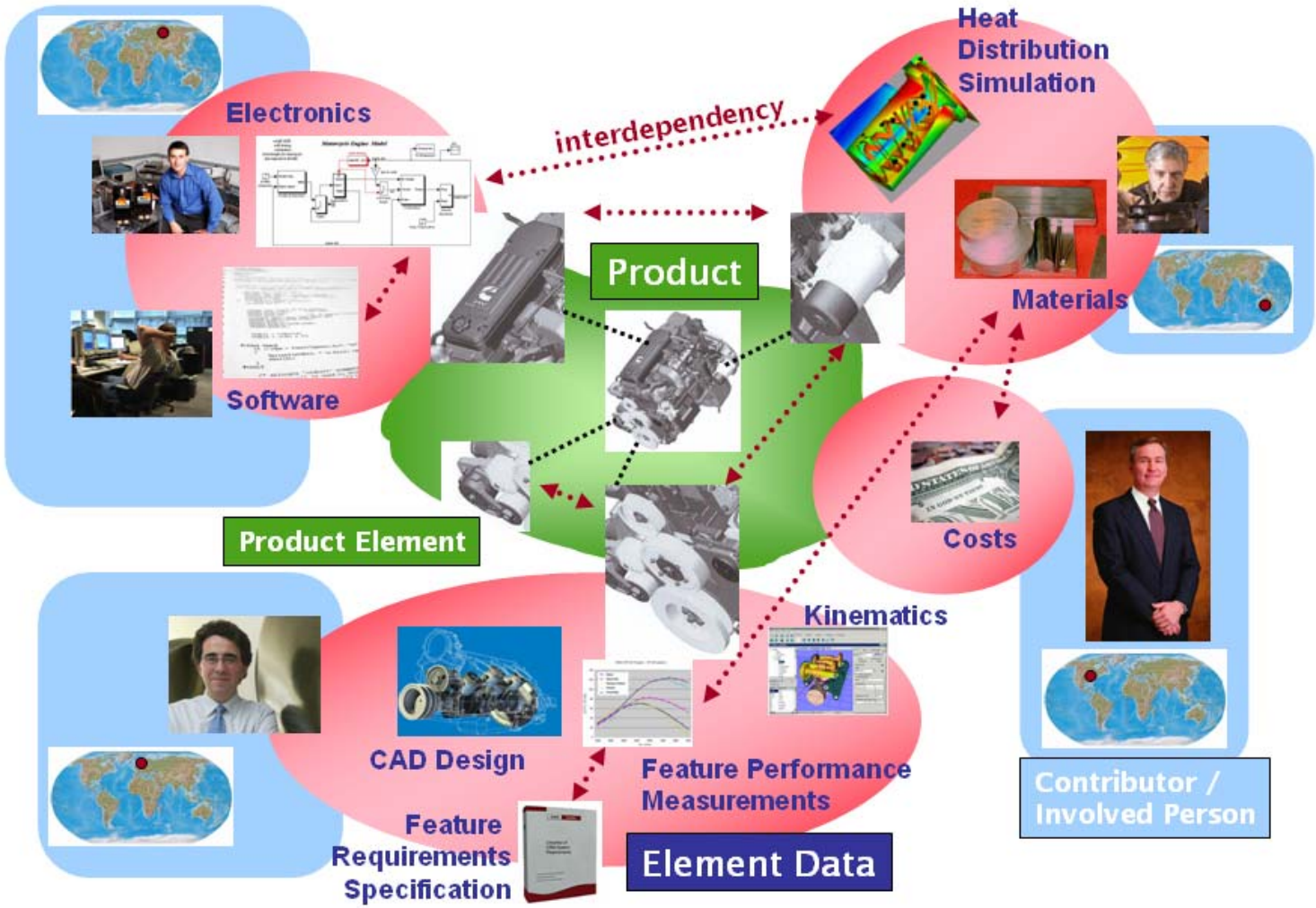
- Requirements on Product Engineering Processes within companies today
  - Decrease costs (of both processes and product)
  - Decrease time-to-market
  - Ensure quality during all process steps at the same time
  - Joint developments between companies (perhaps short-term collaborations)

- Redesign processes are getting more important
  - Engineering teams have to deal with design problems where a lot of data already exists (and a lot of design decision have been taken)

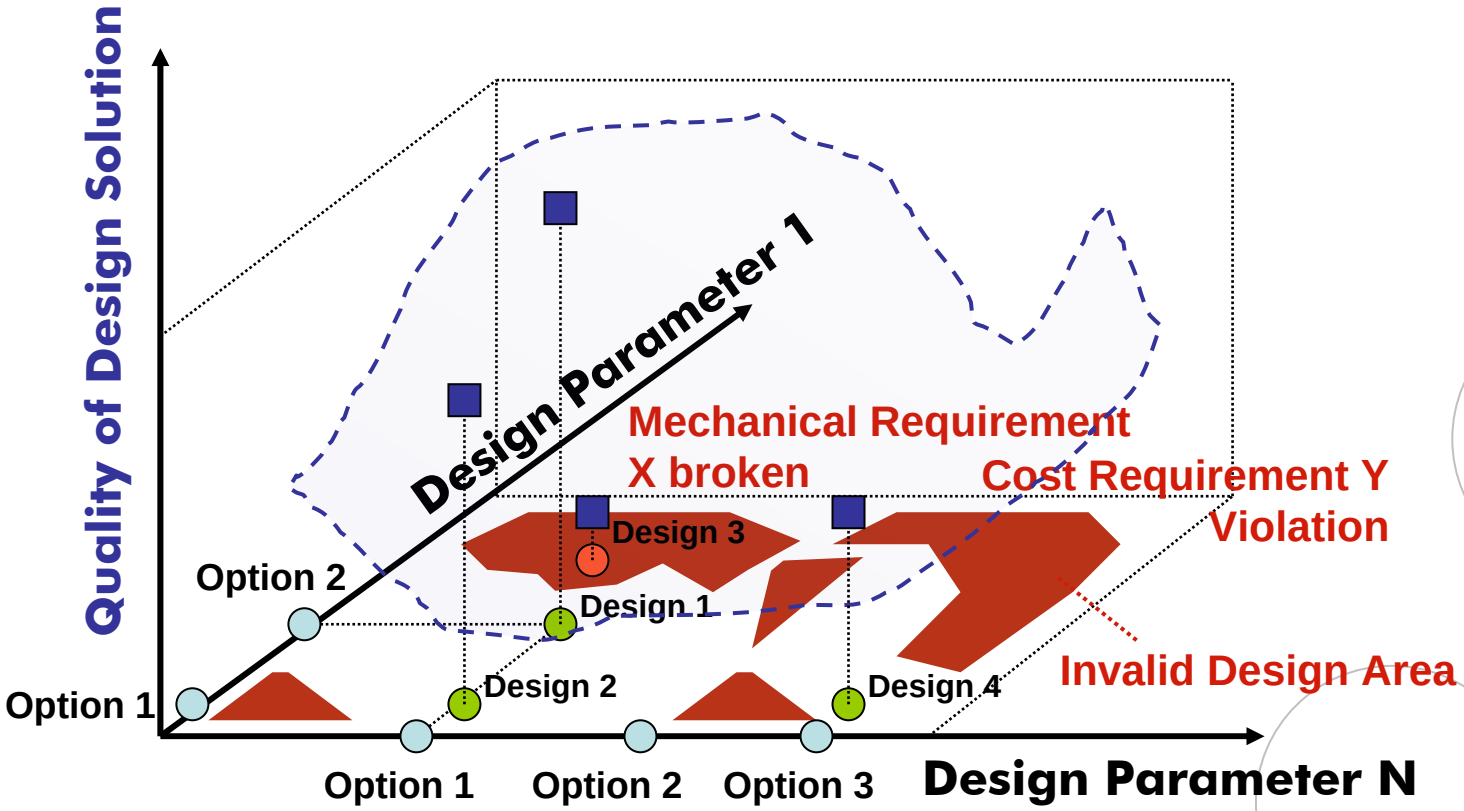
⇒ Companies need to have a good control over the process

Product Engineering

Is getting more and more complex today ...



A different angle on the Product Engineering Process



- Product Engineering Process = Collaborative, human-driven search process in a high-dimensional configuration space

How to meet the new requirements on the PE process?

- Key observation
  - The requirements can be met if the search process converges as fast as possible to a valid configuration of acceptable quality
    - Search process is human-driven: performed in a distributed team & happens mainly outside the computer, information an competencies are distributed
    - Search process is knowledge-driven: in each design step, an engineer needs to take into account all available knowledge about all related parts of the product and respective design changes
    - Efficiency of the search process is fundamentally determined by the efficiency of communication (over space and time) between participants in the process

⇒ Support and improve the collaboration within a distributed design team by facilitating communication of knowledge between all participants in the product engineering process by an IT system

⇒ Semantic Engineering Support Enviroment (SEnSE)

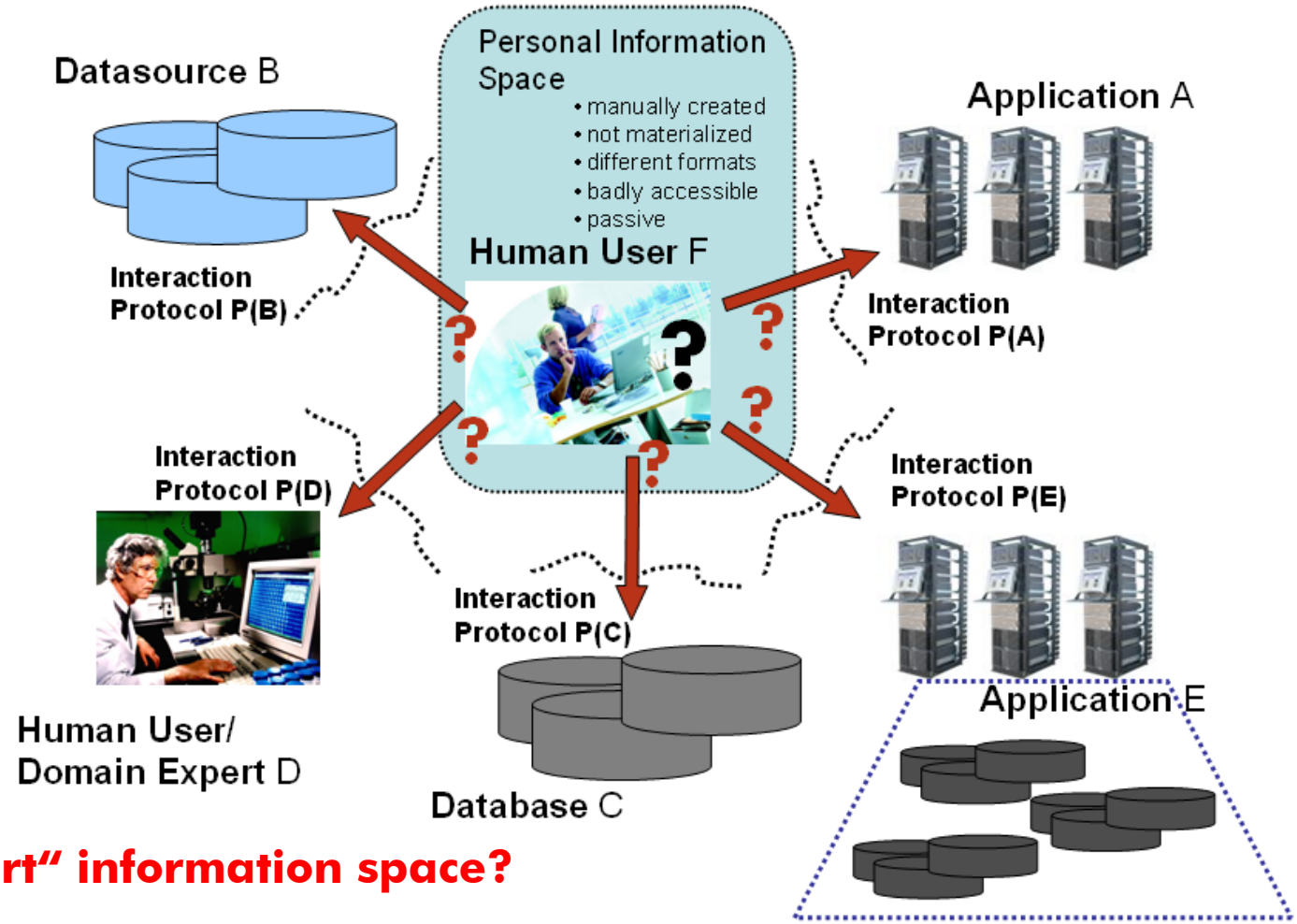
Current situation for an engineer in the process

Engineer = Knowledge Worker

- Burden of information-gathering is pushed completely to the human user

- Absorbs a lot of attention of engineers

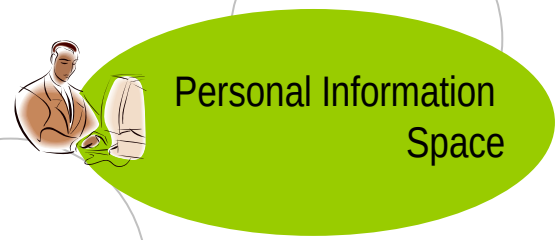
This is lost attention for creative tasks!



What about a „smart" information space?

What do we expect from a support system?

- Explicit representation and management of information spaces
  - Design activitites are information-driven / knowledge-based
  - For this task, people access a virtual information space (i.e. Often not physically represented and managed in a full manner)
  - Virtual information spaces span usually multiple information / knowledge sources (tools, people, ...), using different interaction protocols and data formats
  - People manage their virtual space themselves (with whatever tools) and re-create non-materialized parts of their information space repeatedly
- Hence:
  - (a) Materialize what is managed in peoples heads
  - (b) Information spaces should be represented in a „neutral“ (i.e. not tool specific) and extensible representation model
  - (c) Representation should allow for access information on the knowledge level (semantic-based)



What do we expect from a support system?

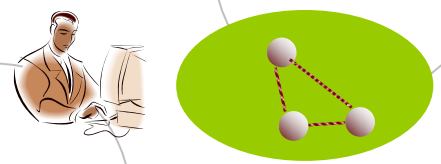
- Organization and access of information spaces according to suitable mental models (or world views)
  - Information spaces might become large
  - Finding information efficiently requires
    - some structure / organization
    - Machine-support for retrieval
  - „Suitability“ of a organizational structure depends on an individual: What is a natural representation of the problem domain for a specific individual?

- Hence:

  (a) Formal / explicit domain models are needed to structure the domain

  (b) structure should be customizable by individuals

  (c) models need to be machine-processable



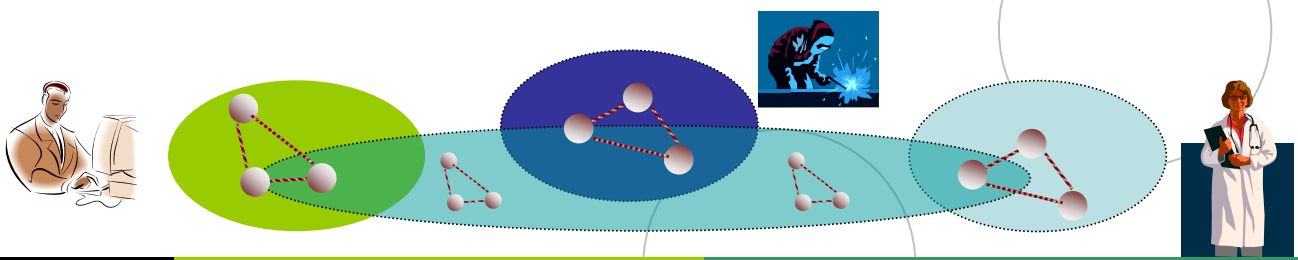
What do we expect from a support system?

- Maintanance of multiple information spaces
  - Information is managed in information spaces
  - Information spaces of individuals are distict and should not interfere a-priori: private information spaces
  - Collaboration by exchange of information:
    - Controlled overlapping of individuals information spaces (shared information spaces)
- Hence:

(a) Information spaces can be shared or private

(b) Information spaces can be structured by controlled overlapping for collaboration

(c) Need for a knowledge representation that can be communicated to others

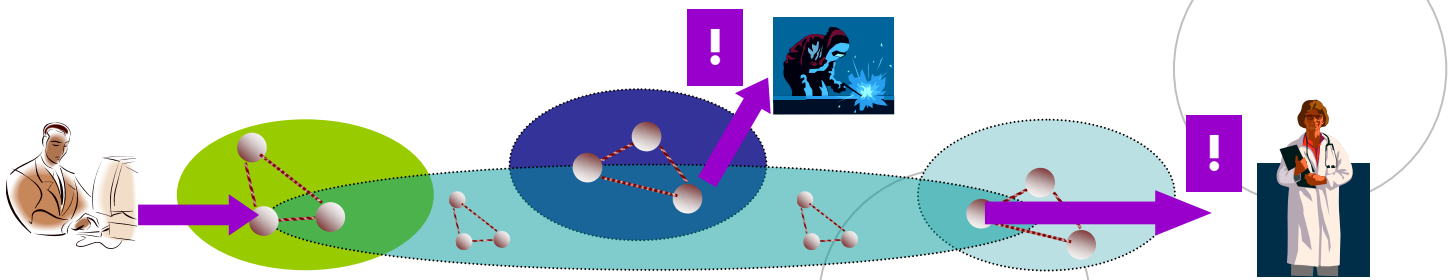


What do we expect from a support system?

- Pro-active support of information needs of engineers by the system
  - Shared information spaces can get large
  - Information in shared spaces pops up at arbitrary moments in time and change the (global) state of the shared information space
  - An individual might become aware unnecessarily late of a change of the global state of the shared information space
  - Structuring the information space does not help to detect efficiently relevant information within a dynamic enviroment (information polling not suitable here)
- Hence:

(a) Pro-actively push relevant information to users of shared information spaces

(b) Allow to specifiy what information is interesting to a client selectively



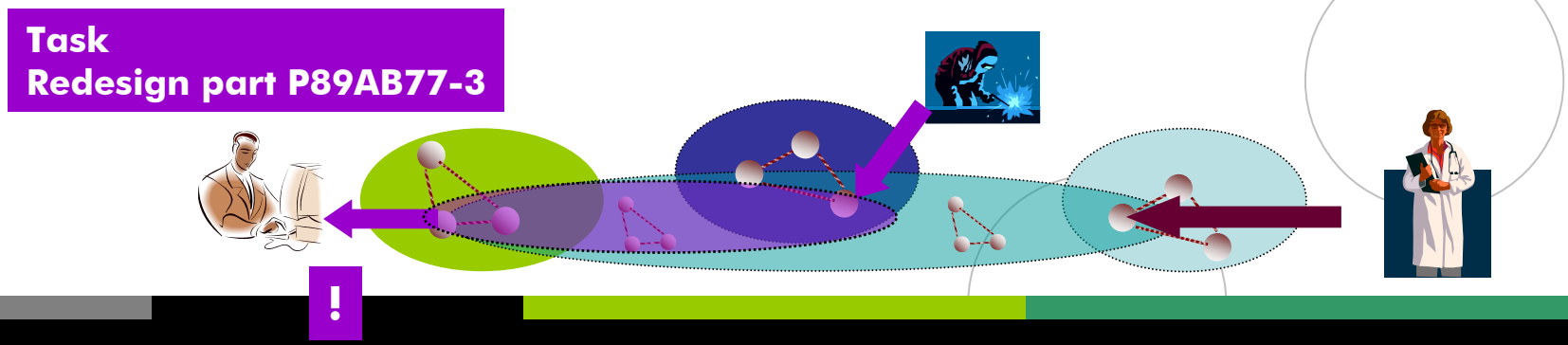
What do we expect from a support system?

- Task-centered support
  - At any moment in time, individuals are concerned with a particular task
  - For this specific context only parts of the overall information space are interesting
    - e.g. for event notifications that are propagated to the client, for reasoning …
  - Individuals have limited cognitive capacity and should focus as much as possible on solving their current task (prevent from distraction)

- Hence:

(a) The system should treat tasks as first-class citizens, represent them explicitly and use them as a central means for scoping whenever possible

(b) Task-specific event-reporting



What do we expect from a support system?

- System can not replace existing tools, but should make minimal changes
  - Various specialized tools are already used with the PR process, that can not be replaced
    - Have been built with a lot of effort (in terms of people and money)
    - People are used to deal with these tools (using them often complex)
    - Process changes are hard to achieve in general, keep them as minimal as possible
  - Let the user interact with the system as a distinct virtual „computer desktop“

- Hence:
  (a) System must act as an invisible glue between the single tools
  (b) System must be able to deal with manifold tool-specific data formats
  (c) System must abstract information from data source format to the knowledge level
  (d) System should appear to the clients as a specific „desktop“

What do we expect from a support system?

- Enable documentation of design decisions taken
  - Design decisions are a major asset developed (with intellectual and monetary effort) during the PE process
  - Design decisions have attached some sort of rationale for why a certain product element came into being as it is (Reflect design history)
  - Do not make the same mistakes again! Learn from the past! (Corporate Memory)
  - Especially relevant for redesign process, where one starts from an existing design

- Hence:

  (a) Provide some dedicated application-level service that allows a team of engineers to capture design decisions and their rationale, and to retrieve the design decisions for certain design elements (Corporate PE Process Memory)

This is the only actually domain-specific requirement in regard of Product Engineering!

Mapping to conceptual system elements

- Ontologies as the fundamental semantic data model in our system
  - Ontologies provide structured, machine-processable and communicatable representations of domain knowledge in information spaces
  - Enable interaction between agents on the knowledge-level (independent of tool specific data representations): semantically self-describing data elements
  - Ontology understood here as: terminological knowledge + instance data

- Ontology Networks
  - Ontologies can be related by means of ontology mappings (ontology network)
  - Flexible and adaptable means to relate different domain ontologies in the system
  - Allows the design of personal ontologies to structure personal information spaces
    - Clients always operate on their personal ontology (world-view / mental model)
  - Networks of ontologies naturally represent networks of (overlapping) information spaces

Mapping to conceptual system elements

- Ontology Management Component
  - Manages multiple ontologies (shared and non-shared ones)
  - Can deal with ontology networks
  - Allows to retrieve and modify ontologies and ontology mappings
  - Synchronizes (concurrent) changes amongst different clients working on the same shared ontology (requires some sort of transaction mechanism)
  - Supports „Publish-subscribe“ mechanism for publication and notifications of events for changes of information spaces of cooperating individuals

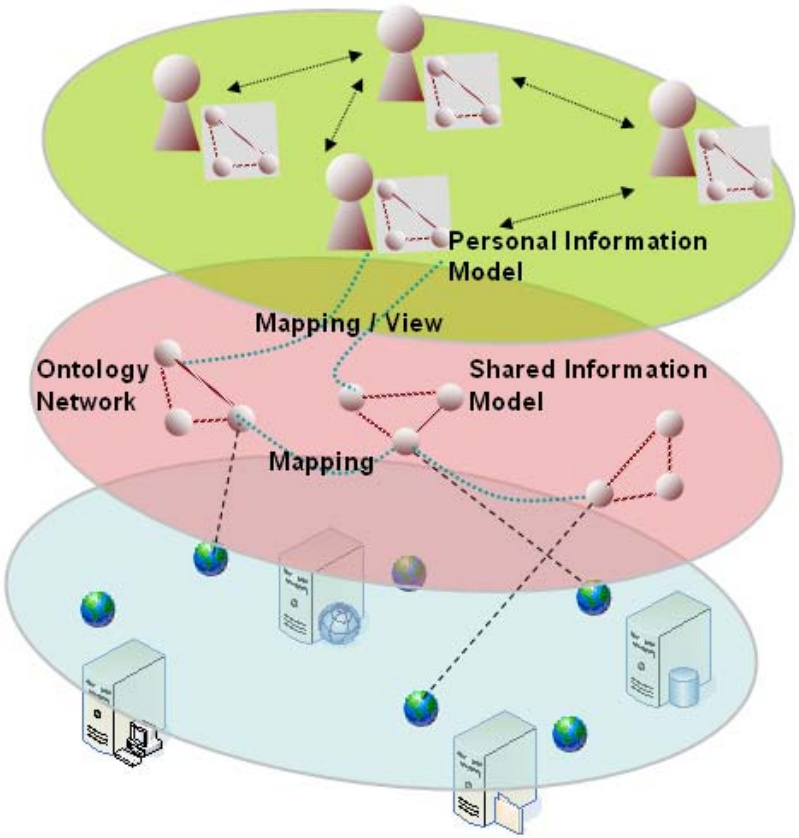
- Task-centered support
  - Dedicated task and user model (as an ontology) to document task related knowledge
  - User task needs to be known to other components: explicit notification by user / prediction
  - Single computational elements in the system are informed about the current task of the user, to make use of it

Mapping to conceptual system elements

- Use an multiagent-based architecture to integrate flexibly various tools and application-level functionality into an open system
  - Every actor is represented by an agent
  - Every application level functionality is integrated by a distinct set of agents
  - Hybrid coordination mechanism: agents can communicate semantically enriched messages both (a) synchronously and (b) asynchronously
  - Integration of legacy sources (file systems, files (PPT, OpenDocument, PDF), eMail Accounts, RDBMS) by means of Semantic Facades
    - Semantic Facade = components that abstract from tools specific data formats and interaction protocols to ontological representations (and the other way around if possible) and modifcations

- Documentation of design decisions
  - Implementation of a Rational Engineering Framework as a specific set of agents on the application-level

Conceptual System Architecture of SEnSE



Agent Layer
How to **interact** with the environment
How to **make use** of the semantic network

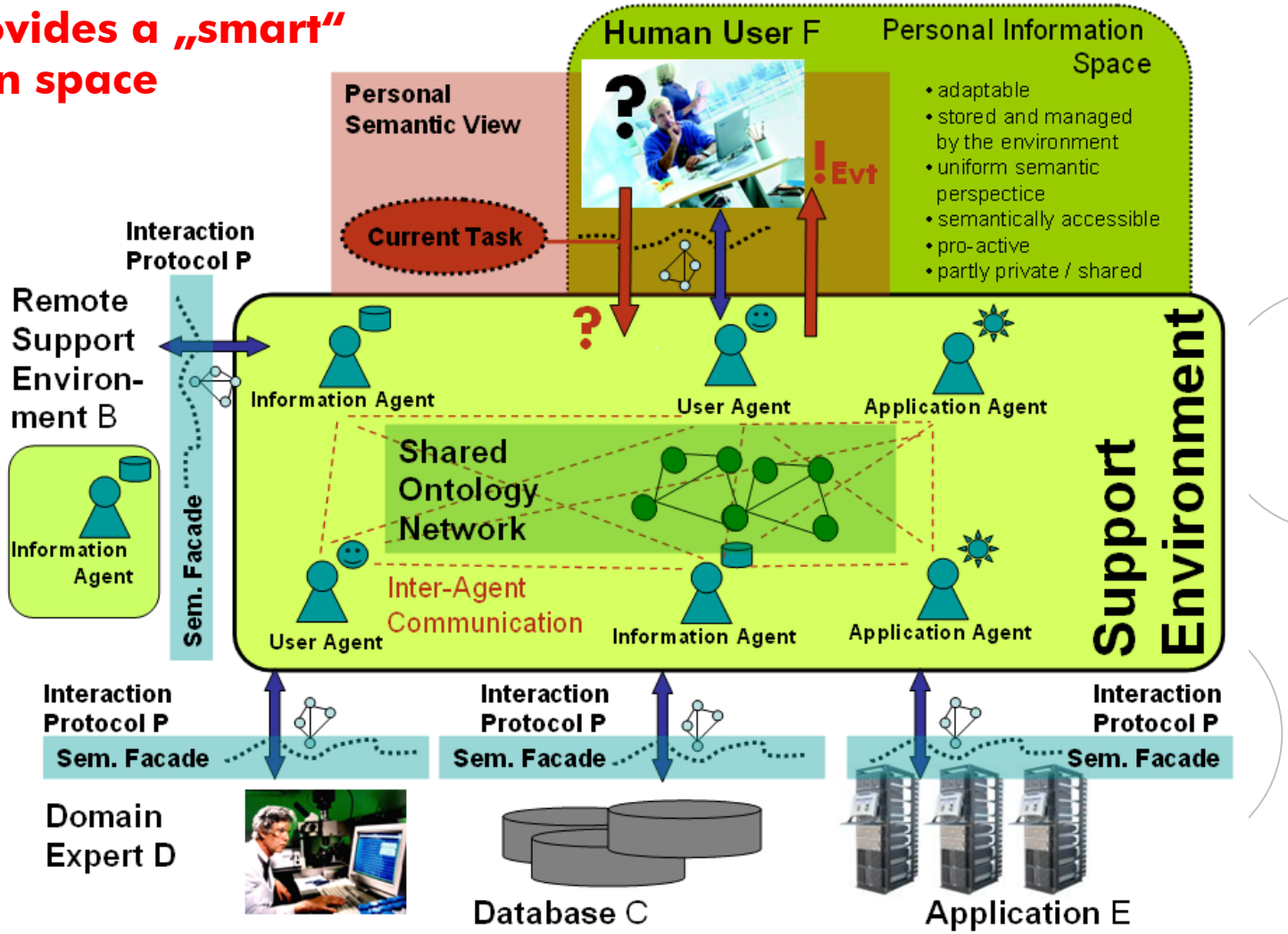
Semantic Layer
How to **semantically describe** information
How to **share semantic information**
Shared knowledge medium: Asynchronous, semantic and pro-active communication channel for distributed entities, **independent** of agents

Web Layer
How to actually **access** information, (Getting physical resources, Web Protocols, distributed access, URIs …)

Layered architecture successively abstract from the specific of the underlying infrastructure

Towards „SEnSEful“ Product Engineering Support

System provides a „smart“ information space



Conclusions

- Semantic Engineering Suppport Environment (SEnSE) =
  Semantic Web Technology + Semantic Desktop + Organizational Memory Systems + Multi-agent Systems
- Semantics and explicit representation of models gives flexibility for changes and simplifies extensions of the system (open environment)
- Central use case for the prototype: MAS-based Rational Engineering Framework
- Prototype Implementation is work in progress
  - Implementation in JAVA, based on the Eclipse plattform
  - currently focusing on completion of infrastructure components and the static part of the system (knowledge management)
  - Integrates various types of file systems and file formats (both ways: read & write), integrates RDBMS in a preliminary manner (read only)
  - Allows a user to integrate resources found via Semantic Facades in his personal information space using his personal ontology