Active Knowledge Modeling of Enterprises

Bearbeitet von
Frank Lillehagen, John Krogstie

1. Auflage 2008. Buch. xx, 436 S. Hardcover
ISBN 978 3 540 79415 8
Format (B x L): 15,5 x 23,5 cm
Gewicht: 847 g

2 Customer Challenges and Demands

In this chapter, we structure and describe the challenges that customers have wanted solutions to for years, but never received adequate IT support to approach. Many challenges have been attempted solved over and over again without success, and that is why many of them became industry slogans. A good example of a challenge never adequately solved is "requirements management and analyses," or in other words being able to turn customer requirements into supplier requirements, specifications, and satisfactory designs. The material presented in this chapter is based on results from the ATHENA project, in particular on (Li et al. 2006).

2.1 Background

Many industrial challenges have survived over a long period, actually as far back as the 1980s. These have now been magnified with the arrival of new technologies, mass customization, and globalization. Some of the challenges were discussed so vividly and often that they became industrial slogans, and examples are "Get it right first time," "Stop the brain–drain," "Avoid reinventing the wheel," "Planners plan and doers do," and "Keep it simple stupid – KISS." Altogether we lived through the 1980s and 1990s with some 15 slogans reflecting unsolved industrial challenges that users expected IT experts and providers to solve. Industrial challenges and road maps are described in nearly all industrial research programs, but no categorization or encyclopedia of industry challenges and demands is currently available.

2.1.1 Structure of Chapter

To be able to relate to the other chapters and in particular Chap. 6 on approaches to industrial solutions, the overall structure is naturally according to Enterprise Knowledge Spaces, their dimensions, key roles, and needs for competences and services. This gives an overall structure as illustrated in Fig. 2.1, starting with ways of categorizing community challenges, then looking at business networking the industry project performance, and finally role and personal workplace challenges. Role and personal spaces produce the architectural structures and contents of workspaces. The enterprise innovation spaces produce the core knowledge of product and service design and delivery, of organizational competences and skills, of process flows and best practice work-processes, and of adaptable agile systems and infrastructures that are easily model-configured. The business networking spaces mainly describe industrial innovation and customer delivery projects covering services, networking teams, project processes, and adaptable platforms. Community spaces describe values, resources, initiatives, and common infrastructures needed to operate a community. The chapter therefore structures challenges and demands into these categories:

1. *Society and Community Cooperation* across industry sectors is discussed. Emphasis is on the vast amount of information not yet digitally available and directly linked to engineering work, and therefore not used or updated by industry. Lack of industrial involvement in developing, using, updating, and managing information contents is another major concern. Finally, we look at the challenges of moving from "blueprint" Enterprise Architectures to *operational* Enterprise Knowledge Architectures.
2. *Collaborative Business Networking* also termed *c-Business*: Emphasis is on the lack of support for shared business models, for digital reference models, business interoperability, and methodologies for inter-enterprise collaboration. Lack of reference models and services for designing interoperable, reusable business platforms are described and discussed. The need for smart cross-organizational service-teams and new knowledge sharing services for simultaneous visual knowledge modeling and execution are discussed and explained.
3. *Interoperable Enterprise Cooperation* models and platforms, supporting project collaboration, providing simpler and safer workplaces, views and services, need urgent solutions. Emphasis is on Web platforms, on personal workspaces, on operational enterprise architectures supporting reuse, and on developing coherent and operational business, project, and engineering methodologies, adapted to specific enterprise projects and tasks. The discovery and existence of Enterprise Knowledge Spaces and the need to develop workplaces and services to support new approaches to holistic design and concurrent engineering, and to provide support for managing work-generative, situated knowledge is discussed.

4. *Innovation and Holistic Design* is possibly the most important business driver in the years to come as PLM systems are failing to deliver vendor promises and meet industry expectations. The major challenges can be found in expressing, representing, and activating knowledge, in integrating the existing PLM systems and in sharing data and knowledge.
5. *Knowledge and Data Representation* is not supporting design, creative work, and engineering. User-defined data and knowledge are currently stored as rigid, predefined data models in legacy databases. Product data are updated, extended, and reengineered through the life-cycles of product development and delivery projects, but this is poorly supported by systems engineering approaches.
6. *Workplace Regeneration and Adaptability* must be supported for designers and creative workers. Workplaces and collaboration spaces must evolve with work progress providing updated services, view contents, and new features, reflecting changes of data and creation of new knowledge.

Finally, we summarize the challenges and discuss the risks that solutions to many challenges and demands related to current practices in systems design and engineering will be sabotaged by the IT system vendors. The challenges to rethink university education and research and innovation are also discussed. Focus is on the time it will take for the educational

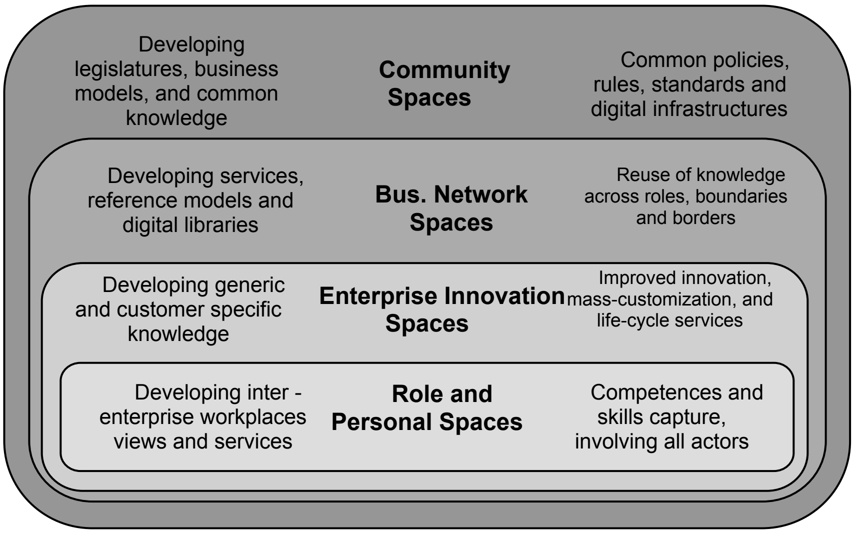

Fig. 2.1. Enterprise knowledge spaces

systems, interest organizations, and industry to acquire the competence and skills and build the confidence and trust to change their approaches.

Only unsolved challenges and challenges with inadequate solutions or where radical improvements are possible are described at any length.

2.1.2 The Evolution of Challenges and Demands

As stated earlier, many challenges date back to the mid 1980s when most actors involved were talking about how to bridge "the islands of automation" caused by application system delivery off the shelf. The evolution of IT systems applied in industry can be categorized in two ways:

1. By Shifts in Market and Business Models
2. By Technology Innovations, Shake-outs, and Exploitation

Looking on this by market and business model shifts, these four stages may be defined:

1. Aggregated industry sector experiences	1980–1990
2. The collapse of vertical markets	1990–2000
3. The push and failure of e-Business	2000–2005
4. The pull and growth of c-Business	2005–2010

Looking on the problem by technology innovation and exploitation, these five stages are identified:

1. Product model integration	1980–1990
2. Business process integration	1990–1995
3. Life-cycle support and integration	1995–2000
4. e-Business services development	2000–2005
5. Model-based engineering and solutions	2005–2010

Now, 30 years after the birth of the first major challenges, we are still devoting lots of resources to make these systems interoperate and exchange and correctly interpret data. The question is: should we rather spend our time designing new approaches to industrial computing?

2.2 Society and Community Cooperation

In today's society, individuals and organizations are confronted with an ever-growing load and diversity of information, causing content management headaches, and with increasing demands for knowledge and skills of turning the information to competence and skills. Coping with these demands requires progress in three closely related areas. First, content must be available as digital libraries providing services for upgrading, accessibility, sharing, usability, and preservation. Second, we need more effective technologies for intelligent content creation and management, and for supporting the capture of human knowledge and its sharing and reuse. Third, individuals and organizations have to find new ways to acquire, contribute, exploit, and manage knowledge, enhancing human learning.

The main challenges, therefore, are to be able to harvest the synergies made possible by linking content, knowledge, and learning; to make content and knowledge abundant, accessible, interactive, and usable over time by humans and machines alike.

Current research is expected to firmly establish digital library services as a key component of digital content infrastructures, allowing content and knowledge to be produced, stored, managed, personalized, transmitted, preserved, and used reliably, efficiently, at low cost, and according to widely accepted standards. The support of more personalized and collaborative services, particularly within self-organizing communities, should lead to more creative approaches to content and knowledge production.

Improvements are also needed in terms of contents accessibility, usability, scalability, and cost-effectiveness of the resulting methods, technologies, and application services with respect to handling large amounts of data and concurrent users. Also, for users to develop confidence and trust in the information sources, the links between content, knowledge management, and permanent learning processes must be improved. Technology should enable us to master content and knowledge exploitation and proactive participative learning from dynamic working environments.

2.2.1 Developing Digital Libraries

Medium term the challenge for most industries is to reengineer manual information sources into global digital libraries with innovative access services that support communities of practice in the creation, interpretation and use of cultural, industrial, and scientific content, including multiformat and multisource digital objects. They should be combined with robust and scalable environments, which include semantic and role-based search capabilities and essential digital preservation features. Particular attention should be given to cost-effective digitization processes and to the use of digital services in multilingual and multidisciplinary contexts.

Longer term the challenge is to develop new approaches to digital preservation, such as those inspired by human capacity to deal with information and knowledge, exploring the potential of new approaches to automatically act on high volumes and dynamic and volatile digital content, guaranteeing its preservation, keeping track of its evolving semantics and usage context and safeguarding its integrity, authenticity, and long term accessibility.

Enriching Today's Information Sources

In most sectors, vast amounts of project information, such as materials specifications, materials and part catalogues, and codes and regulations, is in the form of printed documents. Information collection from these sources implies manually searching and reading documents and often reinputting the same information and data. Efforts to turn this information into active, manageable, and sharable knowledge with stakeholders should be given the highest priority.

Information encoded in natural language, on paper or in preformatted data models, is hard to access, extract, interpret, adapt, change, and manage. Advanced search engines and parsers might be able to figure out what the information is about, but for semantic analyses and knowledge preservation, there is a need for adding more purposeful semantics. As a consequence, a growing number of textual structures are emerging to support semantic Web techniques, for supporting standards, and for developing identification and classification schemes.

Ontologies, taxonomies, thesauri, and other "name–structures" are being developed and applied to add layers of semantics to digital information sources, enabling automatic processing by semantic search engines. Although traditional search engines understand words and word patterns, a semantic search engine will also understand the context in which the word appears. This will be an improvement for retrieving, sharing, and integrating information content, but to understand the true meaning of information and data, role-specific contexts configured by knowledge models should be provided for. Many ontologies are based on the OWL standard (Smith et al. 2004). However, having ontologies based upon OWL does not assure full compatibility between the ontologies, just as having a standard based upon XML does not guarantee compatibility with other XML standards. But it does provide a range of standard tools to choose from and a range of other ontologies to build on. Creating a metaontology capturing all other ontologies is not feasible as each ontology represents a particular perspective view. Perspective views are human knowledge representations, and replacing them by a metaontology destroys their user value. An alternative to metaontologies is mapping ontologies. Either by mapping one ontology to another or as done within the Information Framework for Dictionaries (IFD) published as ISO 12006-3:2007, mapping ontologies to other ontologies or to standards such as IFD by the use of a generic framework structure can be achieved. A more powerful, configurable, and user-driven mapping technique should be developed. More background on semantic Web technologies is found in Chap. 6.

Automating Information Management

Having semantic structures embedded in the information makes it possible to go one step further and identify rules, requirements, and exceptions from the same source of information. This can be done by further tagging of the information or by importing the information to "Reasoning Engines" or to knowledge modeling and execution platforms. The combination of semantic tagging and model-configured approaches, activating rules, exceptions, and requirements should enable effective data extraction directly from the information source to the user workplace. Using mark-up techniques, we should be able to generate "pseudo rules" from textual information. The pseudo rule plays two roles:

- Making the rules language-independent, which means it should be understood across cultural borders, industry sectors, and among stakeholders
- Making possible automatic code generation and execution of pseudo rules to fit different reasoning engines and execution architectures

These techniques are useful for reengineering legacy information sources that are rapidly becoming degraded and obsolete, and then no one will trust or use them. The big challenges are to allow users to do this from their workplaces without having to call on IT experts, and to provide easy to use services to upgrade the contents, define "name–structures," and configure new use services.

2.2.2 Enterprise-Enhanced Learning

On the job training and learning by performing work or role-play will enable industry to engage in more aggressive bidding and contracting, being able to better predict milestones and costs, and thus industry will raise its competitiveness by calculating more predictable margins and lowering risks.

Medium term the failure to meet the challenge for developing responsive workplaces and environments for technology-enhanced learning and learning by doing is rapidly becoming an obstacle to collaborative Business. To motivate, engage, and inspire learners to use learning, services embedded in the business processes and human resource management systems should be provided. This should also involve the transformation of learning outcomes into permanent and valuable knowledge assets. Focus is on the mass individualization of learning experiences, contextualized and adaptable to age, situations, culture, and learning abilities. Learning by doing, performing work, integrating pedagogical and organizational approaches has the advantage of exploiting visual scenes of action, interactivity, collaboration, and context-awareness, all supporting proactive learning.

Longer term adaptive and intuitive learning services, able to support learning through self-configuration from knowledge architecture contents and experiences of the learners' behavior, should be supplied. Cross-disciplinary research on the synergies between learning and cognition in humans and machines should lead to systems able to identify learner's requirements, intelligently monitoring progress, capable of exploiting learners' abilities to let them learn faster. Learning services giving purposeful and meaningful advice to both learners and coaches for self-learning and for learning in collaborative environments should be developed.

2.2.3 Developing Operational Enterprise Architectures

In the current industrial and economic context, enterprises and their systems need to be constantly and smoothly reengineered to respond to changing market demand and technological evolutions. Enterprise architecture (EA), considered as the foundation of enterprise systems, has emerged as a "tool" to help stakeholders manage system engineering and change. EA is not just about IT, it involves strategy, business, knowledge, human factors, and assets. EA is both a challenging and confusing concept. For decades, construction industry uses architecture in the design and construction of all size of buildings. Their "architecture" utilizes standard symbols that can be recognized and understood by all members of their industry for carrying out the construction work. The systems engineering community by comparison has never had the advantage of this type of "time tested" structure. Instead, since the beginning, many heterogeneous architecture proposals have been developed. They are often overlapping approaches, and the underlying concepts are not explicitly defined. Different architecture description languages and model templates are not interoperable, and are consistent in the concepts they support. These languages and templates are proprietary, lack expressive significance to represent specific features, and cannot support operational IT system solutions. Similarities and differences between EA methodologies cannot be perceived by users; and this creates obstacles for its correct understanding in industry and finally its acceptance and use. The lack of a generally agreed terminology and an enriched knowledge corpus in this domain is also a bottleneck for its efficient application.

Enterprise architecture as a "Skeleton"

EA is a conceptual, simplified, and aggregated representation of the basic structures, processes, and organizational structures of an enterprise. As a market for IT systems, it emerged in 1996 with the US Congress passing of the Klinger Cohen Act. EA does not start with technology, but a strategic framework, focusing the vision, goals, priorities, and business activities. An EA is a specific arrangement of business features and functions. The purpose of a "should-be" (target) EA is to maximize a set of business goals and objectives given a set of constraints, conditions, and challenges. The purpose of "as-is" (baseline) architecture is to document the current arrangement such that a transition to the desired target state can be determined.

Independently of business goals or strategies, EA is first and foremost, the foundation of enterprise knowledge structures and IT systems. According to ISO 15704 (2000), an architecture is a description of the basic arrangement and connectivity of parts of a system (either a physical or a conceptual object or entity). The software community also considers that architecture is the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding their design and evolution (IEEE 1471 2000). Specifically, software architecture is "a set of software components, externally visible properties of those components, and relationships among them."

More generally, architecture must possess the following features and functions:

- Have properties that can be verified with respect to user needs (e.g. open or closed architecture, interoperable or not, centralized or decentralized, flexible or rigid language etc.)

- Be communicated as simple views so that business people can easily understand, check, analyze, discuss as a “language” shared at corporate level
- Have a style (by comparison with construction where architecture can represent some particular characteristics of a building such as “gothic” or “romaine”). EA should be able to characterize enterprise systems (e.g. “fractal,” “holonic,” or knowledge-model configured)

Various Types of Enterprise Architecture

ISO 15704 considers two and only two types of architectures that deal with enterprise integration.

1. *System architectures* (sometimes referred to as “type 1” architectures) that deal with the design of a system, e.g., the system part of an overall enterprise integration
2. *Enterprise-reference projects* (sometimes referred to as “type 2” architectures) that deal with the organization of the development and implementation of a project such as an enterprise integration or other enterprise development program.

In other words, type 1 architecture represents system or subsystem in terms of its structure and behaviors. The type 2 architecture is actually a framework aiming at structuring activities/tasks necessary to design and build a system. For example, Zachman’s architecture (Zachman 2007) is regarded as a type 2 architecture. Other works make distinctions between *conceptual* and *technical architectures*. The conceptual architecture is derived from business requirements; and are understood and supported by senior management. The technical architecture provides the technical components that enable the business strategies and functions. Sometimes conceptual architecture is also called *functional* or *business* architecture; and technical architecture, ICT architecture. TOGAF (TOGAF 2000) considers four types of architecture, which are subsets of EA: Business architecture, information technology architecture, data/information architecture; and application (systems) architecture. Lillehagen et al. (2002a) advanced the concept of “knowledge architecture,” separating perspective views of business operation, knowledge and ICT architectures as illustrated in Fig. 2.2.

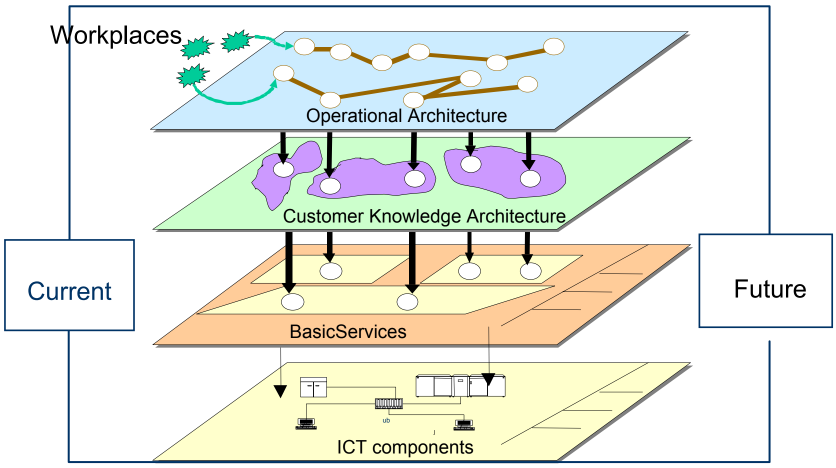

Fig. 2.2. Layers of architecture

To link the four layers of architectures described to one or more of the established EA frameworks is a major challenge, and this will involve new principles of performing visual modeling, linking work environment modeling and work execution, and finally being able to do this and preserve context for the key roles involved, and separating local adaptations and extensions from project and network wide changes. European research projects are attempting to achieve this and more, such as supporting model-configured collaboration spaces, workplaces and services. One of the projects (CoSpaces 2007) is rooted in the TOGAF enterprise architecture model.

General Architecture Principles

General EA principles can be found in the literature such as for example: (a) Business processes drive technical infrastructure; (b) Primary purpose of architecture is to facilitate rapid change; (c) EA must emphasize reusable component building blocks; (d) Architecture must be enterprise-wide; (e) EA must optimize the enterprise system as a whole and many more. Supporting operations is unfortunately not one of the principles.

Another approach can be found in the Government of Canada's Federated Architecture (2001), where these principles are proposed: (1) Reduce integration complexity to reengineer application systems to be "highly modular" and "loosely coupled" to be able to reuse components; (2) Adopt holistic approach with a (whole of enterprise) approach;

(3) Have business event-driven systems; (4) Plan for growth and construct for growth and expansion of services (known requirements) across enterprise; (5) Provide robustness, responsiveness, and reliability with appropriate redundancy to protect against system failure.

Generally speaking, when developing an EA, the principle of fitness-for-purpose should be followed. It means that the architecture should be developed to the point at which it is fit for purpose and not further.

Technical Architecture Principles

Cockburn (2003) have proposed some architecting design principles, for example: (1) Create an interface around predicted points of variation (because things change, and we must protect the system integrity across changes); (2) Separate subsystems by staff skill requirements; (3) Make one owner for each deliverable (People get confused if ownership is unclear); (4) The program is totally program driven, with the user interface just one driving program (because user interface requirements change a lot); (5) Provide a single point of call to volatile inter-team interfaces (Protect developers against rework due to an interface change). Malan and Bredemeyer (2002) also suggested three principles to develop a "Minimalist Architecture": (1) if a decision could be made at a more narrow scope, defer it to the person or team who is responsible for that scope; (2) only address architectural decisions at high-priority architecturally significant requirements; (3) as decisions are added to the architecture, they should be evaluated from the point of view of their impact on the overall ability of the organization to adopt the architecture.

In TOGAF (2000), some principles underlying the design and successful use of specific architectures were proposed, for example: (1) An architecture needs only to specify those services that is required; (2) Elements of an architecture may specify one, more than one, or only part of a service; (3) Elements of an architecture should be defined in terms of standards relevant to the services they specify; (4) Elements of an architecture should be reused from all the categories of the Architecture Continuum and should support reuse of solution elements of the Solution Continuum; (5) Elements of the solution or implementation should be reused from all the categories of the Solutions Continuum; (6) An architecture must be followed, or it is useless: formal IT Governance practices are recommended.

To summarize there is a need to develop commonly accepted architecture representations and specification languages as active knowledge models, enabling architects and key networking roles:

- To describe and represent a common point of departure for families of different enterprise solutions – regional and sector supported operational platforms with EA services to tailor workplaces and services
- To support managers and project leaders in their strategic and business operational decisions to innovate and build new architectures and to pursue new projects and business opportunities
- To answer a key problem: provide architecture service continuity along the whole enterprise and product life cycle (requirements, design, implementation, operation, and replacement)
- To amplify features and functions of various architected platforms so that comparison and choice become easier for users and can be performed upstream, saving time and resources
- To ensure interoperability between enterprises, workspaces, and services, built from various architectures, but with common architectural components and standards
- To effectively support new approaches and methodologies for improving most engineering disciplines in existing and new collaborative networked organizations

The list is by no means meant to be exhaustive.

Community Platform Development and Operation

The major challenge to overcome and develop coherent digital libraries that can be shared by many stakeholders is to provide a generic core platform with some predefined role-specific workplaces and standardized services, supported by generic software components. This should allow members of project teams to build model-driven, role-specific workplaces, and supporting services and views.

For this to happen, model-driven platforms must replace software applications as the means of delivering computing services and capturing industrial knowledge and pragmatic logic. The extreme enterprise knowledge encoding for computer execution support has to seize. Various information libraries, integrated and adapted by modeling, must be made accessible and executable from a range of technology platforms. Future solutions and services to cover growing needs, such as product portfolio management and mass-customized product delivery, must be developed and deployed. This is illustrated in Fig. 2.3. Different categories of knowledge workers will be able to define, share, and manage data, information (documents), and active knowledge models all integrated by a standardized, operational enterprise knowledge architecture.

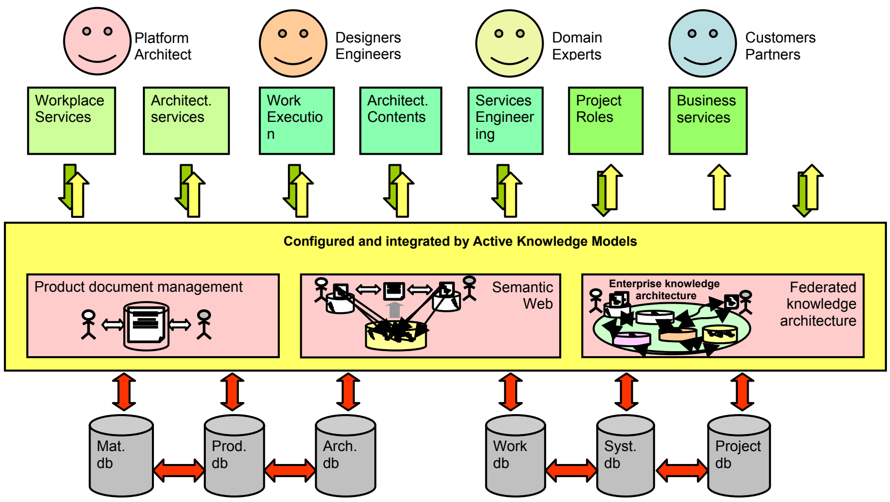

Fig. 2.3. Architecture for role-specific workplaces and services

The enterprise and other knowledge architectures should be implemented and supported by model-composed and configured platform services and methods, implementing an intelligent infrastructure for dynamic solutions engineering. Some of the services must be automatically performed as a consequence of user interaction with natural tasks, but without users being aware of their execution. Otherwise the user dialogue will be much too complex for product designers, industrial engineers, and practitioners to learn how to effectively communicate with the system. Developing the knowledge architectures, the workplaces, the views, the task structures, the basic services, the metamodels binding services to software components, and the overall architecture model are all major interdependent challenges.

2.3 Collaborative Business Networking

Effective collaborative business networking is dependent on industry, research, and system providers solving these major challenges

- Providing services to agree on and renegotiate business models and cooperation rules and issues, even during the execution of a project, including services to share risks and values

- The existence of reference models (reusable sharable knowledge) that can be easily accessed and applied by all stakeholders needing access to their contents
- The development of interoperable systems and established common architectures as models for enterprises and networked organizations to build their own specific models through adaptation to their specific context
- The availability of services to dynamically build collaboration rooms, compose teams and services, and deploy working environments across geographically dispersed enterprises

Services to allow the different actors to share concepts and best practices that make cooperation possible should be part of the core c-Business platform.

2.3.1 Business Models

The theory of network economics is a relatively new topic of research in international economic science. This field of research emerged mainly in the beginning of the 1980s when a growing number of contributions in the field of standards were recognized in the literature. Following Weitzel (2004), it is instrumental that the industry becomes familiar with the basic ideas behind the network economics theory and why businesses should be familiar with the concept. According to Shapiro and Varian (1998), economics of networks are one of the central differences between the old and the new economy: The old industrial economy was driven by the economies of scale; the new knowledge economy is driven by the economics of networks" (Shapiro and Varian 1998). Thus it seems inevitable for survival in the new economy to understand the principles of network economics and their implications for market dynamics.

In Chap. 10, we will take a more detailed look at the many theories of knowledge network economics and what a holistic design approach to business and technical interoperability will involve.

Business Interoperability

In electronic business relationships, interoperability plays a decisive role. Being "interoperable" refers to being able to share information between business partners, understand and process exchanged data, seamlessly integrate it into internal ICT systems, and enable its beneficial use.

Business interoperability is defined as "*the organizational and operational ability of an enterprise to cooperate with its business partners and to efficiently establish, conduct and develop IT-supported business relationships with the objective to create value*."

On the basis of this definition, business interoperability describes characteristics of a company's external relationships. It extends the more technically focused notion of interoperability to cover strategic, organizational, and operational aspects of setting up and running IT-supported relationships. As such, business interoperability builds on the concept of *networkability* (Wigand et al. 1997; Österle et al. 2001), which is a continuation of coordination theory and sees coordination as the management of relationships of dependence. Among the challenging issues that may arise at the business level are the following issues:

- Defining the business cooperation model and identifying target partners
- Defining consistent business goals and the rules of cooperation
- Formalizing these goals and rules in signed contracts and service level work processes
- Aligning business processes and internal work processes among partners
- Making technology choices and integrating architectures and platforms
- Sharing knowledge and linking information systems across company borders

The breakthrough for networked organizations will occur when companies can cooperate with new partners without any additional cost involved, and even small businesses can easily participate in electronic business relationships. This scalability of electronic relationships is called *m:n capability*.

When comparing different industries, it becomes evident that they are characterized by different levels of business interoperability. In the high-tech industry, the supply chain between original equipment manufacturers (OEM), contractors, and component manufacturers is tightly integrated. In many other areas, e.g., in facility management, the fragmentation and specialization within the value chain is still in progress and has not yet produced stable role models. In addition, the size of the companies makes it more difficult to establish a similar level of inter-organizational integration. These examples illustrate that the achievable level of business interoperability depends on industry structure as well as product functionality and maturity with regard to electronic business and characteristics of the target cooperation scenario.

2.3.2 Reference Models

A reference model (Latin; refere: carry back, reporting) is a *general model* for a class of issues with the following characteristics:

- On the basis of a *general model*, specific models can be developed
- The *general model* can be used as an object for comparison with other models that are describing similar issues

A reference model, therefore, provides an ideal master for a class of issues. A reference model should be relevant for a distinct application domain and describe structures, properties, relationships, and the behavior of objects. A reference model is composed of these main components:

- Basic model building blocks; structures, components, parameters, rules, and services
- Architecture of the entire reference model and its modules with data and examples of practice
- Description language that is used for a uniform display and for exchanging information between different competence centers
- Rules and guidelines for applying the reference model to generate a specific model

Many industries have engaged in developing reference models, but there has all the time been some reasons for industrial reluctance to join, such as giving away core knowledge, costs of community meetings, and time spent on reengineering and deploying the models. However, the major reason for industrial reluctance is the fact that some of the most valuable reference models are developed and deployed as paper documents. Extracting and inputting reference model data is a major expense and source of erroneous data, causing many engineering changes and delays.

Categories of Reference Models

Many reference models are by purpose, scope, and contents already categorized by the organizations behind them. Aligned with the purpose of our approach, we propose to provide these models as platform embedded configurable knowledge models and data. We have identified five major categories, just as there are five major categories of interest organizations:

- Application domain focused, such as the Supply-Chain Corporation (SCC) and the Change Management Institute (CMI)
- Information or Architecture focused like The Open Group (TOGAF) and the Federal Enterprise Architecture Federation (FEAF)

- IT or other technology-specific organizations like OMG and W3C
- Standards or norms for data exchange and information flow like the STEP communities
- Industrial standards, norms and rules like the international initiative in the construction industry and the International Association of Interoperability (IAI)

The industrial platforms of the future will provide workplaces for product and process design, organizational learning and development, and systems design and engineering, where digital reference models can be easily integrated, reengineered, and reused.

Importance of a Reference Models

Reference models represent high-quality knowledge and best practices sharable by all stakeholders involved in community projects and work.

Among the most obvious benefits are:

1. Approaches that provide quality and secure solutions
2. Methodologies that lead to good solutions for all actors
3. Solutions that are repeatable, repairable, and replaceable
4. Norms and components for solutions and information on solutions
5. Norms and components for Infrastructures and platforms
6. Reusability and replication of solutions and parts across a community
7. Enterprise knowledge for global industrial training and education
8. Major areas for further research and development

All these advantages and benefits means great potential savings in not having to reinvent and rediscover knowledge that should be considered noncompetitive, easily accessible and adaptable by qualified stakeholders.

Reference models should be major repeatable knowledge components of the enterprise knowledge architecture (EKA) of networked enterprises. However, today they are mostly used as fragmented information in documents to support disjoint engineering disciplines.

Quality of Reference Models

Experiences from applying EA tells us that any reference model developed by “slide-show” or textual models will not represent a consistent, coherent, and compliant set of reference models for networked modern enterprises. The manual work in developing them is alone becoming too costly, never mind accepting the costs to develop community services to allow users to adapt and apply to business specific solutions. The manual toil and costs of these processes and the lack of services for composing and managing project-specific models are preventing communities from developing models of high quality and value to industry and industrial users. Any model has its own architecture, part of a holistic knowledge domain, and these can be made active or interactive components of operational solution platforms. They should be part of the second layer of EA – the EKA). The groups that develop reference models have little or no contact, so the models will not be compliant or coherent. This is potentially a major source of noninteroperability. These reference models should be integrated in the EKA by standardized metamodels and modeling languages as extensions to the IRTV language.

Reengineering Reference Models

Many reference models need to be reengineered for increased correctness, consistency, compliancy, and cohesion among them, and be developed and delivered as knowledge models accessible from inline repositories. These repositories must provide services to make the reference models easily available and adaptable to changing enterprise solutions for increased user value. In order to support life-cycle stakeholder involvement and user interaction, one must support knowledge and data service provisioning to users. The projects to develop these reference models with the qualities needed and the services mentioned will have to be performed in industry sector focused projects with competent users to engage in holistic knowledge modeling approaches with industrial users in the driver's seat. The kind of initiative required is best exemplified by the IAI initiative (IAI 2007) in the construction sector. The sector oriented initiatives may take a holistic approach to reference model design and engineering as they can involve and accommodate most industrial perspectives, application models, and standards particular to that sector. Sector initiatives start with their own approaches and methodologies to develop, build, and deliver solutions with increased use of and support for reference models, ranging from design rules and norms for construction details to project approaches and process models for multisite plant management.

Challenges using Reference Models

The way reference models are being developed today, they are a major source of noninteroperability. Some major points are listed below and are further explained in this section. The problems and challenges in using reference models are caused by:

- Mostly textual-based description makes adoption and application difficult
- Too detailed information without layers reduce the applicability of the reference models
- Too high expectations to reference models lead to wrong implementation
- A lot of incompatible standards and de facto standards reduce interoperability
- Reference models are often not aligned with the entire enterprise architecture and goals

Integrity and consistence cannot be ensured on textual descriptions, which makes adaptation difficult and resource consuming. The motivation to fit these reference models to the own enterprise conditions and especially in order to join enterprise networks or force collaboration is low. The consequences are along two dimensions. The reference model will be not adapted, so the effectiveness and unique competitiveness points will be lost. Or the model will be adapted to the own business, but this is connected with costs and risk. Especially a continuous adoption is still a problem. In case of using the reference model of an enterprise application like ERP, an additional risk appears: When the software provider is changing the reference models by providing a new software release, the consequences cannot be foreseen and possibly the specific reference model and the specific customization of the IT System cannot be maintained according to the requirements of the new software release.

Industrial reference models, in particular, the IT-system reference models contain too much detailed information defined inside the reference models. Often only a single layer abstraction with lots of details exist, such as in UML. The adaptation by business people is awkward if not impossible. Another shortcoming is the fact that such models carry a-not-invented-here stamp. In industrial use, very often unrealistic expectations to reference models exist. The fact that a given model is either designed for defining requirements, for providing a high level framework and guidelines, or for providing best practice solutions is often forgotten. Because of various standardization bodies and de facto standards, overlaps exist between reference models to similar business items (e.g., ITIL vs. eTOM). Additional integration effort is clearly needed. By applying both frameworks the user has to integrate not only different terms for the same issue, but has to fight with different level of granularity provided by the reference models. The needed adaptation leads to double effort for development and maintenance. The reason is that most of the reference models do not refer to general enterprise business architectures, which should be independent from the technology and the business aspect of the reference model. Different languages for describing the reference models make merging and synchronization very cost and time intensive.

2.4 Interoperable Enterprise Collaboration

Problems of distributed collaboration are central to the effective management of the product lifecycle, particularly where heterogeneous technologies, tools, and working practices are involved. Many products are developed by means of different technologies on the basis of both hardware and software components. While this is true today for high end and complex products (e.g., cars, aircraft, or mobile phones), this trend is expected to extend to almost all products in the near future (e.g., household appliances).

Three issues are crucial for the proper functioning of product: management, life-cycle support, and reuse, if we are going to succeed in coping with the market trends and meeting the customer demands. The three issues are:

1. The need for virtual work environments and collaboration spaces
2. The need for interoperable knowledge architectures, securing optimal reuse
3. The need for new approaches to Systems Engineering and solutions management

The more multifunctional and complex the product is, the more complicated are the work processes for its design, engineering, customized delivery, and life-cycle support. Industrial knowledge can be flexibly architected and reused, but the manufacturing plants and assembly lines, built with hardware and physical constraints, must also be able to manufacture the customized products. This requires a methodology for modularization of manufacturing and maintenance processes that must be available as design rules to the product designers and engineers. To achieve this support, simultaneously designing for customizable solutions, manufacturing and life-cycle modularity is a challenge that will stay with us for some more years.

2.4.1 Virtual Enterprises: Collaboration Spaces

The period from 2000 to 2005 was dominated by research toward creating the virtual enterprise (VE). A VE was defined as *"a customer solution* *delivery system created by a temporary and IT enabled integration of core competencies"* (Tølle et al. 2002):

Infrastructure development for virtual enterprises faced three highly intertwined challenges (Jørgensen and Krogstie 2005):

- *Heterogeneity*, incommensurable perspectives, software infrastructures, working practices, etc., among the partner and customer companies
- *Flexibility*, many interdependent knowledge dimensions, the need for learning, design changes and work process alignment, and exception handling
- *Complexity*, the richness and uncertainties of interdependencies among partners, their activities, resources, skills, and products

From some 50 research projects known to the authors, not a single virtual enterprise was made operational. However, some very useful discoveries were made, and some important lessons were learned. Among the important discoveries was the need for building active knowledge architectures, supporting both user-configured and model-configured work environments and role-specific workplaces. In other words, it takes more than software engineering technology to build a real operational VE. The most common and important lesson learned was that industrial product and process knowledge can only be understood and improved when working intimately close with industry. Interpreting specifications written by consultants has for many years been a major challenge for IT people. The VE research drive has now been replaced by the drive for more concrete industrial collaboration spaces, and indeed many projects are recognizing one lesson learned from these VE projects: "No collaboration without contextual roles, views, tasks and industrial data and information!" It is no longer a tools game!

2.4.2 Process Structures: Emergence and Evolution

Unstructured creative activities are often most important for the competitiveness of an enterprise. Even in seemingly routine work, exceptions and uncertainties permeate the environment. Workers reflect upon and manage these problems in a sophisticated manner (Wenger 1998). To some extent, on the one hand, most work can thus be regarded as knowledge intensive. On the other hand, most work processes also have routine parts, which can be structured and automated. Many companies have prescribed quality management procedures for administration, audit, approval, etc. Systems must thus integrate support for ad-hoc and structured work (Haake and Wang 1997; Jørgensen and Carlsen 1999).

Users must be supported in selecting a suitable degree of plan specificity for the current state of their process, balancing plan complexity with the need for guidance and control.

In software engineering, researchers have defined process classification schemes, e.g., to select appropriate methodologies. Reflecting the wide *diversity of processes*, even within a single industry, up to 15 classification dimensions with 37,400 process types have been proposed (Cockburn 2003). This number suggests that predefined ways of working cannot be constructed for all process variants. Instead, process methodologies should be selected and model-configured to the particular circumstances of each project.

2.4.3 Knowledge, Communication and Learning

Inter-organizational and multidisciplinary cooperation requires not only information exchange, but also knowledge sharing. Effective teams must form across local cultures. Common frames of reference, reference models as discussed earlier, are established through working together, so support systems must allow the meaning of terms, plans, and artifacts to evolve. In communities of practice, this learning process is called *negotiation of meaning* (Wenger 1998). Ambiguous models are required because the meaning of formal, well-defined terminologies cannot be negotiated. A VE infrastructure must be intelligent that is it must support the process of creating, negotiating, and reconciling diverging views and interpretations.

Lack of integration into everyday work practices is a reported shortcoming of Knowledge Management (KM), enterprise modeling, and process improvement (Davenport and Prusak 1993). KM too often becomes the domain of outside experts that lack a full understanding of the complications of work and the local language of the work community (Wenger 1998). Work performers become sources of information to KM activities, not active participants. Standardization and codification, rather than local innovation, organizational and social learning, become the focal points of KM. Failure rates above 50% are common (Lawton 2001).

The gap between what people say, observe, and do makes it difficult to use enterprise models and other official accounts of work as input to KM (Argyris and Schön 1978). It must thus be straightforward to modify enterprise information locally. Still some knowledge cannot be articulated and will remain tacit, but visual collaboration spaces supporting proactive behavior will take us a long way. Most descriptions are incomplete while they are used, subject to an ongoing elaboration and interpretation. Change and learning demand that modeling infrastructures be open and be integrated with execution platforms. Knowledge models are completed only when they are no longer in use, and may no longer be elaborated to reflect exceptions and changing circumstances, but then they are most likely obsolete.

2.4.4 Intelligent Infrastructures: Integration and Customization

The unique nature of each VE, and the dynamic set of partners, seldom makes it economically viable to integrate information systems through developing new software interfaces. Standardization (Chen and Vernadat 2003) requires that the domain is static and well understood, and is thus seldom appropriate for knowledge work. Consequently, we need flexible infrastructures that allow shared understanding and semantic interoperability to emerge from the project, rather than being a prerequisite for cooperation and collaboration.

Such flexibility is seldom offered by the tools currently available for virtual enterprise integration, like e-business frameworks, workflow management, enterprise resource planning, etc., (Alonso et al. 1999). Consequently, how to achieve flexibility, configurability, context preservation, exception handling, and learning are important research topics in these disciplines.

Simple tools invite use. Software that offers a wide range of functionality often becomes overwhelmingly complex, complicated to use, and incomprehensible. Consequently, only a small portion of the available services is utilized. This condition is known as *featuritis*. We thus need role and task-specific user interfaces, emphasizing what is needed in the current context. Interfaces and semantics should also adapt to the local needs of each project. Enterprise models, articulating who performs which tasks when and why, are powerful resources for such adaptation.

Workplaces should also adapt to the skills and preferences of each individual. Where experts should be given freedom to exercise skilled judgment, novices need detailed guidance. Personalization fosters a sense of ownership, motivating active participation. Studies have shown that personal templates and configurations spread informally through the organization, improving processes and disseminating knowledge in an emergent manner. We will, however, contend that VE integration is as much a social problem as a technical one. Current modeling infrastructures emphasize technical integration, but the understanding of virtual enterprises as socio-technical systems must be improved. In particular, we seek to replace the common approach of using formal computer languages to control social interaction, with the application of human languages to control and customize computing infrastructures.

2.4.5 Enterprise Interoperability

Interoperability among enterprises wanting to collaborate for stages of or whole life-cycles of processes and products is becoming a competitive lever for industry. The first extended enterprise to build a bidding network involve the stakeholders in negotiating a potential winning bid and interact with partners to assess performance factors will have a tremendous advantage in future c-Business markets. Now, interoperability is as Peter Drucker says not a technology, it is a property of an enterprise or any knowledge space or knowledge dimension, just like scalability, transformability, and similar capabilities that may add up to defining agility.

Enterprise Interoperability can be achieved in at least three ways, and most often by a combination of the three:

1. By reconciliation of business objects and services
2. By reengineering the legacy
3. By enterprise design and development

The three approaches and their supporting technologies are all needed, but the platforms and services offering these capabilities to industry as user services are still at the research stage.

2.4.6 System Engineering Approaches

The trends in systems engineering (SE), aiming to create more agile and better quality systems, are toward more model-architected, model-driven, or model-based solutions, and toward supporting user-driven visual communications among stakeholders and users across communities, projects, and product life-cycles. People with a strong SE background, see INCOSE (INCOSE 2007), believe in either a mathematical foundation or a mix of mathematics, semantics, and pragmatics for progressing systems engineering. The Microsoft Software Factory initiatives have definitely discovered that pragmatic knowledge is the key to any product design, and maybe also holds the key to SE. Some efforts are based on pragmatics and the nature of knowledge modeling and human learning life-cycles.

Research indicates that there are five or more distinct categories of systems and SE approaches emerging and that should be considered, just as there are many approaches to product design and engineering. Ongoing model-based SE efforts will lead to various approaches depending on the specificity of the information and tasks to be supported and the degree of user involvement in the life-cycle stages of the systems. The five major types of IT solutions and systems currently emerging are:

1. Global data collection, analysis, and presentation
2. Process monitoring and control systems
3. Customizable business, trading and transaction systems
4. Technical calculation and analysis systems
5. Product and process design systems

Examples of these five types are under development around the world. The NOSA project (NOSA 2007) in US is a good example of category 1. Category 2 is found in most process industries, in energy producing industries, and wherever human judgment of complex scenes must be assisted in real-time. The seven major IT vendors are employing modern technologies to develop category 3 systems. Also most EU research projects are focusing this system category. Scientists have used category 4 systems for quite some time, but the way they were engineered they could not easily be reconfigured to deal with new artifacts. Category 5 is the most demanding approach as design involves concurrent learning, problem-solving, and collaboration. However, in all five categories, needs for creating and managing local adaptations and configurations, personalized workplaces, and content viewing exist, so model-based approaches and reuse from operational knowledge architectures is the common denominator.

2.4.7 Embedded Systems Engineering

Traditionally, software and hardware development has been performed in separation with little or no interaction. Today, this border between hardware and software products is vanishing. An increasing number of industrial products integrate both hardware and software components, and the decision whether a specific function should be implemented in hardware or software may come late in the project and may even change during the product's life cycle. When the border becomes vague and even emergent, then it is no longer possible to keep the development organizations separate and to use different life cycle processes. So there is a need for unifying traditional product design and engineering with systems development and software engineering. We believe both camps could learn and benefit from each other. However, the requirement for such integration points out a number of problems, such as rethinking and reengineering: work processes, information structures, data management and information flow, infrastructure support, resource management, tool integration, and leveling cultural differences. Finding homogeneous and consistent ways to manage processes, data, and tools have proven to be difficult and challenging, and experiences of quality problems and catastrophic failures are many. Most experiences are associated with the introduction of new electronic systems in automobiles. Several attempts to integrate tools from product and system domains are known, but all report limited success. The main reason for this is that enterprise integration is not achieved by systems and software tool integration. Industrial experiences tell us that four factors play a crucial role for successful integration: role-specific knowledge capture, pragmatic work processes and tasks performed, people's mental models and attitudes, and the overall availability and access to shared information and data. Currently, the functionalities in partially but inadequately integrated system configuration management (SCM) and product data management (PDM) systems are so complex and inconsistent that collaboration across any border is prohibited by usability problems and excessive cognitive load.

The fundamental differences between product and software engineering stem from the fact that, in product engineering, structures and parameters are processed simultaneously by many stakeholders, while in software system engineering, the software components are processed one by one by individual experts. Therefore, product engineering focuses on modeling and sharing knowledge of the total product, while software engineering focuses on building and testing reusable components of the end product. Product engineering emphasizes the creation and management of knowledge as engineering artifacts for roles along the entire product life cycle, while software engineering emphasizes code engineering, individual programming, and debugging of the software components. Clearly, the two approaches have their respective merits and limitations, but being faced with different issues, they end up with different solutions to, apparently, some of the same problems. Therefore, it is not a surprise that the tools provided in one camp cannot fit the needs of the other camp. What makes the problem really difficult is that the existing tools (e.g., SCM and PDM), being based on radically different assumptions, cannot be extended to include the "missing" functionality. For the same reason, combining (integrating or interoperating) one tool from each camp, at best provide a functionally awkward system, too complicated for practitioners to use.

Industrial solutions will require deep rethinking of the very nature of the work processes and product and system knowledge artifacts created and managed, what are the underlying architectures and methods, and how can a common platform be created. Currently, the problem manifests itself at the system level where hardware and software subsystems/components have to be integrated. At least at system level, technology-independent product representations and working processes are essential for achieving a properly functioning product. This is lacking in today's Product Life-cycle Management solutions, and in SCM and PDM systems; it constitutes serious challenges for parties from research to tool vendors and business partners.

Estublier (2000) discusses how to provide a high level view of SCM applications, which is independent from the particular tools. The application's behavior, services, and properties can be described at a high level of abstraction (process control, paradigm control, security, etc). Their experimental system and metamodel shows that advanced state-of-the-art features could be easily included into a federated architecture, where systems can be fed by data, which in turn can be used in an extended enterprise solution focusing product design or other major tasks.

Work at Chalmers University of Technology (Vinnex 2007) has aimed at developing an integrated product lifecycle model framework, connecting information models representing a product through its lifecycle ranging from customer needs to product retirement. To achieve this, product model theories from different domains such as mechanical, electronic, and software engineering were compared. Similarities and differences were found between these models. The Chromosome Model theory tells how to implement information requirements posed by mechanical/electronic products. So some work has been done to find ways of unifying product design and SE, but there are still major challenges with respect to theory and industrial practice, including the achievement of;

- A deeper understanding of the industrial requirements for collaborative development in the area and of the shortcomings of current commercial solutions vs. these requirements
- A shared terminology for interdisciplinary product development enabling engineers from different domains to communicate and collaborate effectively
- A clear understanding of what PLM functions can be generalized across businesses and what function that need to be adapted as services in a business context
- A coherent theoretical basis and concepts that can guide the development of digital product models and of generic work processes, including how to maximize the generic part and how to minimize the business-specific parts of PLM solutions

- Collaborative design, performed by globally dispersed teams, needs to have a holistic approach, considering aspects of technologies, methodologies, and organizational services
- Proof of concept prototypes, meeting concrete requirements, must be validated in industrial settings, including distributed development and multisite installations and applications
- The usability of model-designed knowledge architectures, workplaces, collaboration spaces, task-structures, and tools must be tested in holistic approaches to enterprise engineering

New technologies and services in modeling, such as component-based and model-based development and enterprise integration, building operational knowledge architectures, based on standards, will have to be developed, piloted, and validated in industrial settings. Getting industrial commitment and involvement to start this work is a major challenge.

2.5 Innovation and Holistic Design

Most industrial products are designed as multifunctional systems with complex structures, employing complicated methods. Modern products consist of a growing number of interacting functions, realized by possibly thousands of parts and components. Industrial product models are today poorly integrated as each engineering discipline and major development step has their own disjoint product structure with their local parameters. To support innovation and modern product design, the product model should be integrated with other enterprise knowledge dimensions such as organizational competences and skills, process views and work processes, and systems, tools, and services. The product model development may involve several suppliers, so the product and knowledge integration should happen automatically or by user interaction as design work progresses. Mastering the interactions between product functions and the interdependencies between all systems and parts needed to build the product is today a major challenge.

Quickly and safely connecting, communicating with and coordinating customers and suppliers is becoming crucial for the survival of any manufacturing industry. Reducing the time to market by facilitating concurrent engineering, increasing productivity by improved work processes and information quality, and reducing costs by improving work environments are still key operational objectives. Also innovative ideas and concept development must be more tightly integrated with product design and customer delivery, with the value-networks involved and with the entire product life-cycle. Innovation is no longer a threat or the foe of successful customer delivery, on the contrary, without a tight integration between innovating new design principles and concepts and customer delivery, there is no way that the value networks will remain competitive.

2.5.1 Industrial Customer Delivery

The authors have during the last years worked with a dozen Scandinavian companies from diverse sectors of industry, and we interviewed them to understand their short and longer term priorities. Among the questions raised, they were asked to list and prioritize their five most important challenges. This is the dominant priorities from a majority of the responses:

1. Poor connectivity and associations between customer and supplier requirements and expectations, causing many erroneous data entries, interpretation, and use of data
2. Too much unfiltered and erroneous data and information cluttering the user interfaces, working environments, and collaboration spaces
3. Contradicting data definitions and input from users throughout life-cycles
4. Industry keeps repeating mistakes, but has problems repeating their successes
5. Knowledge and insight is lost when key people leave or are unavailable, what is known as the "brain-drain problem."

To solve these urgent high priority challenges, most industries will rely on enhancing their product life-cycle management (PLM) systems with knowledge management tools. Some industries explicitly stated that solutions must become more role-specific and context preserving to ease knowledge capture and reuse. This indicates that industry is becoming aware of the shortcomings of present IT systems. Industry also gave high priority to capabilities that would improve their innovative abilities, their effective collaboration and communication with customers and suppliers, creating proactive collaboration spaces and model-configured, visual working environments.

Some of the key customer requirements for holistic design and concurrent engineering are:

- Designers must be able to model their own concepts and define visual languages to describe their designs to fellow designers and engineers

- Designers should not have to learn to use any IT tools to communicate their designs. Most software tools should be hidden from the designers and engineers
- Suppliers should be able to contribute their ideas, provide specifications, and prepare bids through workplaces generated by purposeful models
- Customers, suppliers, providers, vendors, consultants, and contractors should be able to access the same c-Business networks and collaboration spaces to qualify for work
- Partners, particularly SMEs, must be able to join business opportunities and delivery projects with a minimum of investments in IT competences and systems

Services to help them coordinate work, capture experiences and lessons learned, grow their knowledge and have knowledge integrate, and drive their business projects are at the top of the longer term wish-lists.

Now, a rapidly growing number of them are questioning whether or not current IT systems and SE practices will do the job.

2.5.2 Industrial Innovation

State of practice in most industries is that in which innovation and customer delivery projects are kept strictly apart, just as business process modeling and improvement, and product development or engineering processes are kept apart. This is because today there are no methods or IT systems that support the need to configure the systems according to market and customer demands.

Lack of support for innovation, the creation and articulation of new ideas of product or process, is today hurting mainly because growing and expressing knowledge from human mental models have no or little support from IT. This is not alone to be blamed on the IT community, but also reflects the lack of knowledge cultivation and language to express and share knowledge among industries. Most industrial sectors do not have the concept of architecture, and no layers of abstraction are available to represent product ideas, concepts, and layouts/arrangements as digital artifacts. Most industrial approaches to product design are supported by fairly static drawing and diagramming techniques. The design process, spanning from the most abstracted requirements interpretation to product-end-of-life, is stepwise and supported by disjoint and specific diagrams, drawings and frozen digital models governed by proprietary application systems. This has manifested the belief, even among designers and pragmatic experts, that industry needs a specific product structure for each major step and engineering discipline. Consequently, methods and tools to integrate between the many structures have also been developed and introduced. Now, prototypes exist to prove that this may not necessarily be required and definitely not desired, as it portrays industrial design as a sequential, step-wise process. So developing integrated product structure models, and extending integration across other domains, is one of the most demanding challenges.

Collaborative Product and Process Design (CPPD)

Some digital reference models already exist to support collaborative product development. They address different aspects and viewpoints for design collaboration, including:

- Change and configuration management are key work processes for collaborative product design (CPD) that need to be shared by all the actors of an industrial project
- Information models like STEP Application Protocols that support product data exchange, sharing and retention all along the lifecycle of a product and within the supply chain
- Project management reference models, supporting change and version management, portfolio management and possibly more services
- Sector standards related to the previous aspects, for automotive, aerospace (AECMA), or military (MILS) product specifications

Unfortunately, the models are being used as physical models, but they should be considered as conceptual models. As soon as they become operational, i.e., adapted to specific business and technical ICT contexts, important challenges exist to establish model adaptability, extensibility, and interoperability. Model interoperability cannot be established if no shared business concepts are built to allow high level communication between processes and applications, and extensibility is not possible at all with current IT-systems. These models, as used, address only one particular aspect of what is required to establish effective cooperation and interoperability. The STEP application protocols for instance focus only on product data and information exchange and management issues. Collaborative design for increased customization and life-cycle support are key requirements for dynamic service-oriented industrial communities. As a prerequisite for competitive offerings of products and services, collaborative design is a catalyst for business success, growth, and customer satisfaction.

2.5.3 Service-team Organization

Collaborative holistic design will demand model-composed, configured, and life-cycle managed services operated by teams in a smart organization. The teams with clearly defined roles and responsibilities collaborate as mutually supportive service-teams. Each team owns well-defined basic services, takes on responsibilities for providing, configuring, and adapting project and customer-specific services and service responsibilities. This means industry should own their own project services to be able to recap and reuse any customer delivery process. Figure 2.4 illustrates the interplay between different such teams. We will return with a more detailed description of this concept in the main contribution part of the book.

2.5.4 Concurrent Platform Engineering

Concurrent engineering of layers of project platforms, extending the knowledge architecture with customer models is fundamental to support

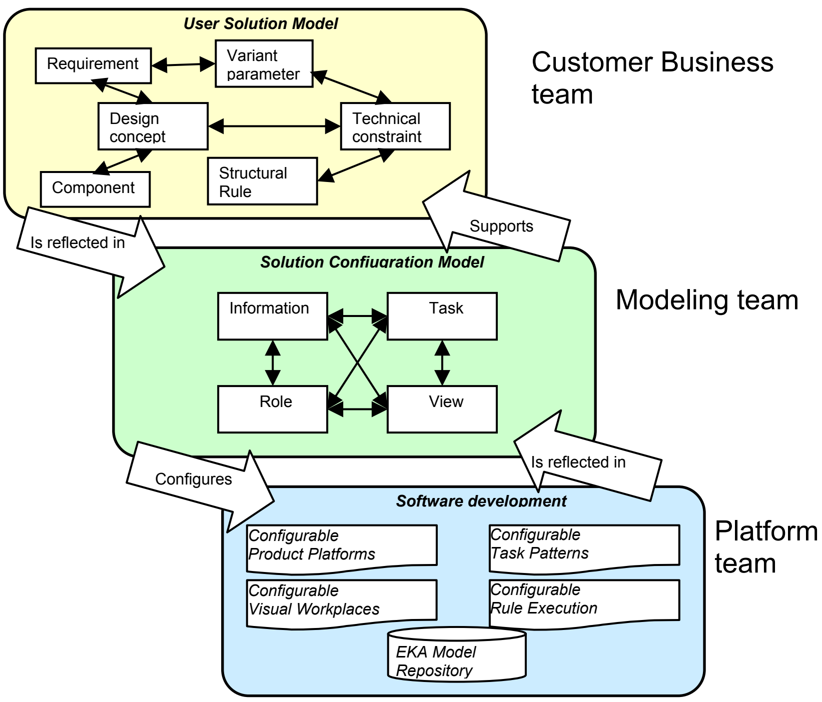

Fig. 2.4. Service-team interaction

holistic design, capturing growing needs, designing new ways of working, cutting lead times, all depend on model-generated shared workspaces and more role-oriented workplaces, separating issues, and supporting simultaneous variants avoiding sequentially forced versioning.

2.6 Knowledge and Data Representation

Information and data exchange between IT systems decides work patterns, and determines work process flows and many tasks that appear in user workplaces. This is evident in all PLM system workplaces where the user is file and content manager for what information is received and sent.

In conceptual and early design, designers must have access to services for defining their own data in their own views, including data formatting views, and associated views to provide context for precise meaning and reuse of the contents by whoever will need access to the data. Conceptual artifacts cannot easily be stored on their own in databases as they do not have any predefined data types. They should be represented as abstract objects combined with their defining tasks. These constructs are what is referred to as artifacts. The knowledge architecture should provide contextual storage allowing designers and engineers to create and recover ideas, concepts and knowledge artifacts. Capturing the sudden good ideas in a form that easily allows industry to recover, interpret, evaluate, and assess their feasibility for realization within an enterprise knowledge architecture and platform that provide flexible and powerful piloting, testing, and learning services.

Supporting collaborative design and concurrent engineering will require solutions to these challenges:

- Securing stakeholder involvement from day one, providing services for role-specific perspectives on and interpretations of the enterprise knowledge dimensions and model domains, managing their particular aspects, methods, and data and parameter values.
- Improving innovation by enabling idea externalization and conceptual design in distributed design environments, enabling robust, dynamic workplaces, and languages.
- Design knowledge externalization and sharing from idea to end-of-life. Team learning and collaboration require simultaneous modeling in *multiple knowledge dimensions*, organizing models into dynamic enterprise knowledge architectures.

- Reducing change and version management by closing the gap between evolving business operations, alternative knowledge structures, and model-configured software support.
- Generating effective role-specific workplaces as well as services and views for portfolio management and agile collaborative decision-making.
- Configuring services for enterprise knowledge capture and architecting, building and adapting *modeling templates, workplaces, and services*, partly automating knowledge management and organizational learning.
- Runtime extensions and adaptations, effectively including SME's in design projects by model-generating simple to use workplaces without demanding IT investments and extensive, time-demanding training.
- Flexible, interoperable, and reusable platform building workplaces on the core platform, isolating software changes. Model execution services should be loosely coupled by *event notification*, making the platforms robust, extensible, and configurable.

Defining, calculating, and balancing design parameters and their valid value-ranges, deciding design bandwidth for product families is an extra challenge to meet customized product design and delivery that will imply more role-specific views of parameters and their acceptable values.

2.7 Personal Workplaces and Interaction

Industrial innovation is dependent on information being collected, harnessed, and shared as knowledge (reflective information views) in context, and converted to operational knowledge that can be activated to contribute to new and improved workplaces and dynamic work environments, if possible avoiding any in-between interpreters of information and data, thus being able to close the learning loop of participative learners. Learning-by-doing or by performing work is a must in order to support distributed design and engineering team-work, as is supporting automatic workplace enhancement from work performance, experiences, and lessons learned.

Successful model-based platforms should have extendable modeling and execution capabilities to support and be supported by these role-specific workplaces:

- *The designers and engineers* deal with evolving and dynamic multidimensional data, such as product structures, properties, and rules. Their workplaces must be knowledge-driven and model-configured with user-composed services and views, and adaptive to changing contexts. The more design and local customization the more need for services to extend and adapt the knowledge architectures and platforms.

- *The knowledge manager* has a very demanding job, having to deal with dynamically created data-models, services, views and workplaces and to support them all in enterprise knowledge architectures. Platform services, views, tasks, and workplaces will change from project to project, and must be coordinated locally as well as across enterprises. User-driven model-based service orchestration must be supported by the basic knowledge engineering platform, which should provide support for tailoring these model-configured workplaces.
- *The model builder* must perform metadata definitions concurrently with product design, and must provide services enabling users to change and extend the data model and its supporting services. New workplaces, tasks, and views will have to be model-designed and configured.
- *The platform manager* must perform platform extensions and adaptations to integrate new IT systems and tools, to support data exchange and data definition and sharing services, and implement protocols to support data collection from various external sources.
- *The workplace designer* must build workplace design models for new roles with tasks, views, and sources of content, and perform workplace configuration modeling, and workplace generation with the preconfigured contents and behavior.
- *The knowledge architect* defines new metaviews, metadata, and methods, and performs the analysis, classification, and standardization of the data and knowledge content of models and the operational enterprise knowledge architecture.
- *The support engineer* needs services to translate and extend EKA structures, services to define and adapt new view types, and also services to integrate data and parameters from partner sources, representing project and business data and knowledge.

2.7.1 Innovation and Knowledge Repositories

Repository services to support the knowledge architectures must be role, task, and view accessible, so services to access the repository following any of the three dimensions must be supported. This would be like navigating a visual map. The more classical identification schemes based on characteristics properties and categorization structures, reflecting user perspectives, should also be supported for those that do not relate to any roles defined.

Common design artifact expression, language definition, and extension and task-structure navigation must be supported. This will change the way we perceive of, engineer, use and manage repositories, enhancing the services on top of relational and object-oriented databases.

2.8 Summary

Summarizing we have defined the major industrial challenges to be:

1. Building searchable digital information libraries of present common information sources, to improve data and knowledge sharing and use
2. Developing consistent reference models that are easily integrated with Web-platforms, to allow more effective community and project extensions and adaptations
3. Developing knowledge engineering platforms and services that can add value to and integrate present IT application systems, "the islands of automation"
4. Developing operational enterprise knowledge architectures and platforms to concretize and make operational current blueprint architectural frameworks
5. Develop methodologies as descriptive templates to support the building of industry platforms, for example the CPD methods to build collaborative design platforms
6. To model reference models that can be reused and drive knowledge standardization initiatives across projects and sectors
7. To support holistic design implying that multidimensional modeling capabilities to express mental models of designers and engineers must be supported
8. To provide modeling team services and role-specific workplaces and views to support concurrent knowledge engineering for collaborative product design
9. To provide model or knowledge architecture configured workplaces to enable new approaches to model-based systems engineering and solutions deployment
10. To provide services to enable data definition and sharing without being dependent on IT-defined data-models, thus supporting idea capture and conceptual design

In addition to these mostly technical challenges, there are educational, organizational, and managerial challenges that must also be dealt with. However, with the Web transforming into a knowledge-sharing medium, "a knowledge reflector and amplifier," we believe the technology will serve to augment the human mental models and help us use more than 7% of our left hemisphere of the brain; the visual part.

The definition of what is an IT-system may have to be rethought and systems engineering will need to align with product engineering and take advantage of knowledge architectures and holistic design approaches. Another consequence is that many architecture and systems standards will have to be reengineered and put into their correct context.