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

3 Industrial Evolutions

This chapter presents a number of cases representing early approaches to Active Knowledge Modeling (AKM) solutions in precommercial EU projects, specifically the EXTERNAL (2003) and ATHENA (2004) EU projects. Although the current AKM approach is described in more detail in Chaps. 5 and 7–10, it is important to look back briefly at the developments that have brought us to where we are. The AKM technology is the result of a long ongoing learning process that we expect to go on for a number of years.

3.1 History of AKM Development

The industrial needs and thinking that sparked the initial development of the AKM technology, in the late 1980s, was inspired and influenced by so-called *Industrial War Rooms*. War rooms, see Fig. 3.1, were created in most aircraft and automotive industries. The industrial war rooms have four dimensions of core innovative knowledge. In early industrial war rooms, each wall was covered with engineering drawings, plots, and familiar paper images depicting traditional aspects and views of enterprise knowledge, described in Chap. 1 as the POPS dimensions:

- Product and Services: depicting the many disjoint product structures, designs, engineering methods, parts, and classes
- Organization and People Development: organizational structures, positions, teams and roles, and their competence and skill profiles
- Process Modeling and Work Management: process and task models, work execution, and management views
- System and Tool Development: use, solutions and maintenance architectures, components and constructs

War rooms were meeting places to discuss the many known, but not all described and considered, and often forgotten, dependencies and relationships between objects, structures, views, and responsible people. Attempts to model holistic life-cycle views of product data were performed, revealing some interesting information. In these models, more relationships had to be drawn between relationships than between objects, simply because the abstract objects defining the relationship were not known and could not be represented. This identified a lack of language for expressing product concepts, systems, and life-cycle evolution, representing layers of abstraction of product knowledge.

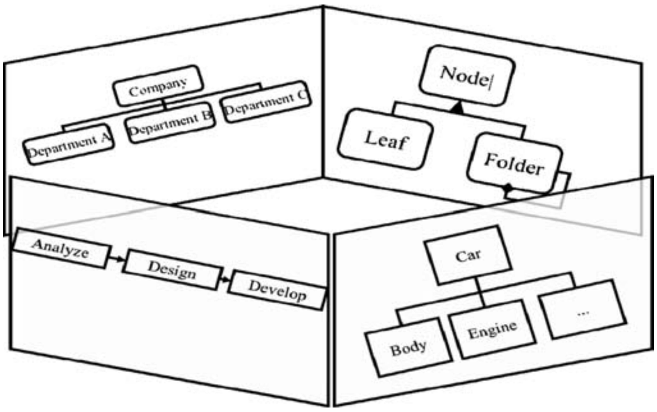

Fig. 3.1. The industrial war room-inspired AKM thinking

The foundations of the AKM technology were discovered in industrial innovation projects, attempting to build digital product models according to war room thinking. Here is an excerpt from a business report from one of these projects, written in 1993:

"Coherent and logically consistent representations of work-centric enterprise knowledge automatically yield reflective views, recursive processes, repetitive working solutions and replicable structures of meta-data. Knowledge from other layers and representations on other media does not possess these properties. Implementing the war room concepts, applying visual modeling languages, automatically give us these intrinsic properties. This in turn yields powerful development, integration, management and reuse capabilities. Most other knowledge domains needed for business operations, such as abstracted process flows, or single views or diagrams of any domain, do not exhibit these properties. Any aspect and view must be derived from the core operational POPS knowledge, to be coherent, consistent and compliant".

On the basis of 15 years of enterprise modeling experience from leading industries, we conclude as follows: "*The variations in knowledge from one enterprise to another are mostly changes in semantics and complexity of structures, in methods and in property embodiment as parameter structures and values. Complexity of structural layers, visual representations, and type* *hierarchies of the four main enterprise knowledge dimensions, in particular of process and product aspects, contribute to the modeling confusions".*

Therefore, to model agile enterprises with support for coherence, consistency, and reuse, we must be able to separate business, knowledge, and IT architectures and models for designing and configuring these layered knowledge architectures.

These early attempts produced important lessons to learn and helped categorize and describe the challenges. Innovation projects such as Volvo IGP, FORD PW60, the Ericsson A project, and McDonnell-Douglas MD12X made significant contributions to bring forward visual knowledge modeling. In recent years, as the underlying web infrastructure has matured, we have come closer to fulfilling the promises of the AKM approach, including configuring workplaces for executing practical work.

3.2 Experiences from EXTERNAL

An early attempt to realize the AKM approach was made in the EXTERNAL project (Krogstie et al. 2002a). One focus in EXTERNAL was to support the formation and running of smart networked organizations, also known as extended enterprises, by combining the resources from a number of existing organizations in forming a common enterprise. The infrastructure to support smart networked organizations, developed in EXTERNAL, consists of three layers (Karlsen et al. 2001; Krogstie et al. 2002a; Krogstie and Jørgensen 2004). These layers are identified as follows:

- Layer 1, the *information and communication technology* (ICT) layer: defining and describing the execution platform, software architectures, tools, software components, connectivity, and communication
- Layer 2, the *knowledge representation* layer: defining and describing constructs and mechanisms for modeling
- Layer 3, the *work performance and management* layer: modeling and implementing customer solutions, generating work environments as personalized and context-sensitive user interfaces available through portals, and performing work

3.2.1 The ICT Layer

The ICT infrastructure is an integration of the enterprise and process modeling tools brought into the EXTERNAL project by the partners:

- METIS (Lillehagen 1999): a general purpose enterprise modeling and visualization tool, allowing model builders to define tailored metamodels and views
- XCHIPS (Haake and Wang 1997): a cooperative hypermedia tool integrated with process support and synchronous collaboration
- SimVision (Kuntz et al. 1998): a project simulator used to analyze resource allocation, highlighting potential sources of delay and backlogs
- WORKWARE (Jørgensen 2001, 2004): a web-based emergent workflow management system with to-do-lists, document sharing, process enactment, and awareness mechanisms
- FrameSolutions (Kallåk et al. 1998): a commercially available framework for building automated workflow applications

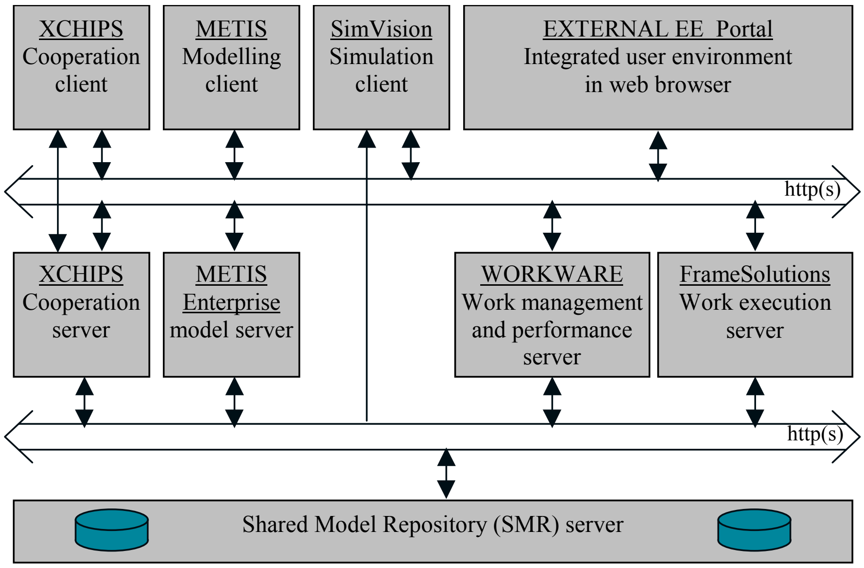

Fig. 3.2. The EXTERNAL infrastructure, ICT layer architecture

The ICT layer is depicted in Fig. 3.2 indicating the clients, servers, and database layers. In addition to the tools described earlier, we have included a portal as a common front-end to automated processes governed by FrameSolutions and to emergent processes supported in WORKWARE.

3.2.2 The Knowledge Representation Layer

The knowledge representation layer defines how models, metamodels, and metadata are represented, used, and managed. A version of Action Port Modeling (APM) (Carlsen 1998; Jørgensen 2004) constitutes the core of EXTERNAL's modeling language (EEML). The kernel concepts are shown in Fig. 3.3 as a simplified logical metamodel. The process logic is mainly expressed through nested structures of *tasks* and *decision points*. The sequencing of the tasks is expressed by the *flow* relation. *Roles* are used to connect resources of various kinds (people, organizations, information, and tools) to the tasks. Modeling smart networked organizations in EEML thus results in models that capture extensive sets of relationships between the organizations, people, processes, and resources. This is particularly useful considering the dynamic nature of networked organizations. For new partners joining the network, the rich enterprise models provide a valuable source of knowledge on how to *behave* in the network.

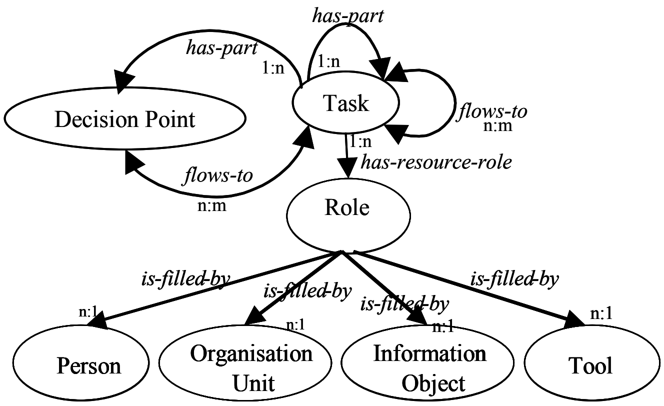

Fig. 3.3. Simplified logical metamodel of EEML

Moreover, the interactive nature of the models, meaning that the users are free to refine them during execution, increases their potential as sources of experience and knowledge. As such, they can be used to *document* details on how the work was actually done and not only on how it was once planned.

From a knowledge management perspective, process models are carriers of work-centric knowledge, that is, knowledge of how to do things, but through the possibility in EEML of attaching information resources to the tasks at any level, such a model also imposes a structure upon the set of information resources relevant for the work described by the process model. To a large extent, the process models themselves form the basis for information management.

The notation of the main concepts within the language is illustrated in Fig. 3.4, which shows a conceptual metamodel of EEML. In addition to the core concepts of tasks, decision points (including milestones), roles, and resources, it illustrates support of goal and competency modeling.

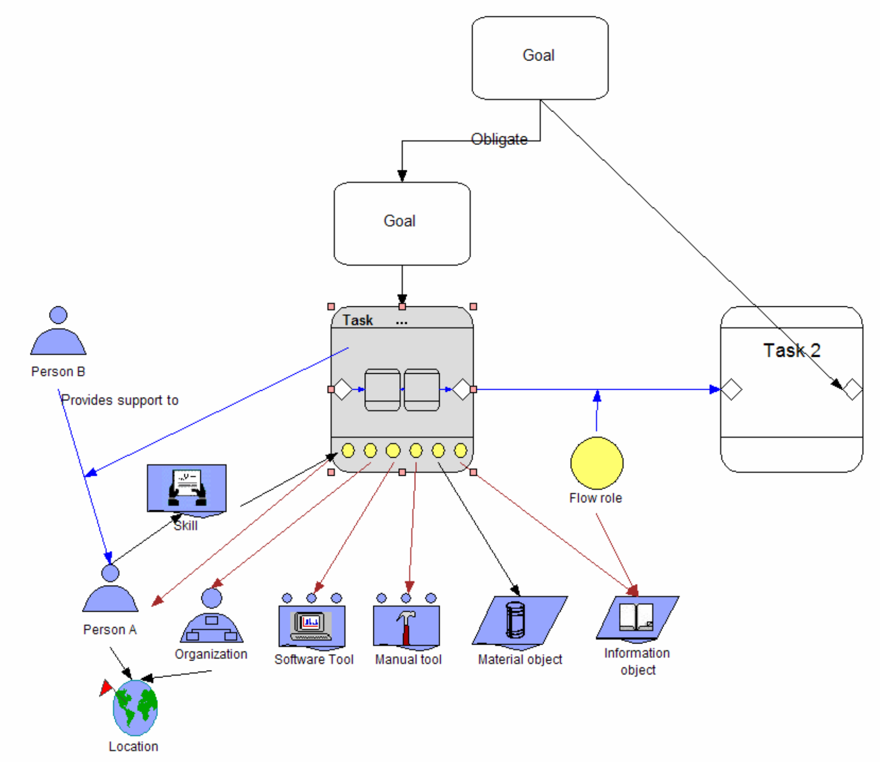

Fig. 3.4. Conceptual metamodel of EEML

3.2.3 The Work Performance and Management Layer

Users access their solutions through portals. A project portal for a networked organization must have support for methodology adaptation and for communication, coordination, and collaboration in teams. Project work management, reporting, and other services must be offered, and finally project work must be performed with possibilities for repetition, providing security and privacy to knowledge workers.

In the EXTERNAL infrastructure, the web-based portal registers and qualifies users, and invokes other tools through the WORKWARE tool set. Modeled tasks are executed through the invocation of tools and applications from the web-based user environment comprising the portal and WORKWARE. WORKWARE sets up the context for each task, giving access to the knowledge and resources needed to perform the task. The actual work performance is done by invoking appropriate services. The task performers may access desktop tools, organizational information systems, web services, or automated processes (in FrameSolutions) through this user environment.

User environments are generated dynamically based on the definition of tasks using EEML. Forms and components for interacting with different model objects are selected and composed based on user interface policies. These policies are also modeled objects. This enables user interface customization and personalization.

The dynamically generated work management interface includes services for work performance, and also for process modeling and metamodeling. The *worktop* (what is later renamed MGWP – Model Generated Work Place) is the main component in this interface. Each task has its own worktop. In addition to the services for performing and managing the task, it contains links to all knowledge in the process models that is relevant for the task. Since the worktop is dynamically generated, subject to personal preferences, the skill levels of task performers can be taken into account, e.g., to provide more detailed guidelines for people who have not previously worked on such tasks. Similarly, customized worktops for project management can support the project management team. The contents may include an overview of the project, adopted management principles, applicable methodologies, project work breakdown structure, results, plans and tasks, technologies and resources, status reporting, and calculations.

The EXTERNAL infrastructure was applied in a number of projects as reported earlier in Jørgensen (2004). These cases constitute a representative selection of knowledge-intensive virtual enterprises. One was a business-consulting firm interacting with its customers. The second was a network of small software companies. The third was an international research project (EXTERNAL itself). Interaction between users and developers ensured an ongoing practical validation. This process started already during the development of WORKWARE in the AIS project (Jørgensen and Carlsen 1999), which was a predecessor to EXTERNAL.

3.2.4 Case 1: The EXTERNAL Project

EXTERNAL took its own medicine as an early experimentation arena for the AKM technology. Experiences were fed back into the development process for the benefit of the industrial cases. The project plan was articulated in early prototype versions of EEML, and later imported to the work execution environments (WORKWARE and XCHIPS). Because of resource limitations and the instability of an evolving infrastructure, it was decided to put particular emphasis on supporting two typical process examples rather than the whole project:

- *Periodic progress reporting*: A mandatory, routine administrative procedure, where reports are written for each work package each quarter, and then collected and sent to the customer (in this case the EU Commission) twice a year.
- *Joint project planning*: A knowledge-intensive activity, elaborating work package plans. Often, planning takes place after reporting to accommodate deviations and provide more detailed plans for the next period.

In addition to these planned case studies, which were carefully evaluated, ad hoc utilization of EXTERNAL tools also took place in the project. The following subsections summarize lessons learned from these cases, focusing on the aspects most relevant for evaluating the interactive modeling approach, the language, and tool support from WORKWARE. More details are available in Haake et al. (2002).

Periodic Progress Reporting

The main activity in this case is quarterly progress reporting (QPR). For each of the nine work packages (WP), the WP manager writes a separate report. The report template and actual report are modeled as information resources to these work items. The project manager is responsible for coordinating and following up the reporting process. In the model, an optional meeting is included for coordination purposes. Though this process is quite simple, it shows that the interaction perspective helps to limit the complexity of the model. For instance, we need no flows from the start of the main process to the concurrent subitems. The lack of input flows means that no constraints prevent the items from starting. Another simplification is evident in the location of the task *Evaluate need for meeting*. This is something that all nine WP managers must do. In systems that only allow one person per task, you would thus need nine items. Here, all WP managers are allocated to one collaborative item. This allocation is made indirectly through the actor roles on each of the *Write WP progress report* items, so if one of the managers delegates the task to someone else, that person is automatically involved in the meeting as well. (In this example, indirect resource allocation does not follow the work breakdown structure.) There are more examples where resource allocation captures dependencies that need not be duplicated to the process dimension. One is the use of information resources to represent the report document parts produced by each WP manager. The architecture of multiple model interactions between the document manager and the workflow engine thus simplifies the models.

The tasks of the project manager (PM) are supported by the services of the infrastructure, and thus need not be articulated in detail. Work lists provide overviews of the current state of the process, helping the PM to see which WP managers have not yet written their report. Through the *Mail to all* service on the worktop, the project manager sends reminders to the WP managers when it is time to write a new report, and again when the deadline approaches. The PM role was reassigned five times throughout the project, so there was a need for explicit coordination routines that the new manager could reuse.

Progress reporting is a routine, administrative procedure that recurs throughout the project at regular time intervals. This model was thus reused a number of times. When the process was first articulated, the support for reuse was limited to copy-and-paste in METIS. A lot of the initial learning and alignment of reporting practices across organizations and countries was already captured in the first version. However, an updated procedure was implemented 1 year later, taking into account experiences with working together as well as increased understanding of the capabilities of the model-driven infrastructure. The new version improvements included the following:

- Changed work breakdown structure to make individual responsibilities clearer.
- Resource allocation was made more explicit to handle reassignment better.
- Added output flows so that *Write QPR* automatically finishes when all of its subitems have been completed.
- Added previous reports for each WP as resource and template for the *Write WP progress report* work items.

Another occasion of end user innovation in the reporting case involved metamodeling. The process in general and the management work in particular are time-driven. The participants decided to model *timers*, a decision connector subclass not part of the EEML at that time, but one that they needed to handle exceptions (delays) and coordination. As WORKWARE did not support timers, the project manager had to remember to do these things, but she was reminded by the presence of the objects in the model. The timers were thus manually activated. For process knowledge management and IS evolution, this information was highly relevant, as it pointed to requirements that were not expressed in the specification documents (Strømseng et al. 2000) but emerged during use. An AKM active knowledge arena has improved such process knowledge management in a number of ways, including the following:

- Incorporating local modifications and metamodeling (adding a trigger-time property to the decisions) supported by the instance-oriented meta-metamodel
- Allowing propagation of dynamic changes so that updated definitions are used in all instances
- Implementing parameterization of model properties. Most of these tasks' names refer to WPs or the current time period, and could easily be generated from parameterization rules
- A specialized, semiautomated reuse metaprocess called *Create new periodic progress report* could be included as a service in WORKWARE. Based on some property values from the user, e.g., the name and deadline of the current period, as well as the current project plan, the reporting process model was automatically generated

Joint Project Planning

Project planning was selected as the second case from the EXTERNAL project because its characteristics complemented the reporting case. Planning is a more knowledge-intensive ad hoc activity, and it utilizes modeling tools for work performance. While the emphasis of reporting was activation and reuse, planning primarily concerns model articulation. It was also expected that the need for coordination between different work packages would require the collaborative modeling services of XCHIPS. The first implementation included the *plan* (a process model) as well as the *planning process* (metaprocess), but not the operation of the plans.

The planning process was modeled in EEML and enacted in XCHIPS. XCHIPS supports closer (in time) collaboration than WORKWARE. When two people work on the same item, they immediately see the effects of each other's actions. The interface provides real-time awareness of who is currently working, and shows the current status of the tasks by color coding. The use case report contains an example of how these features were utilized for defining a template (Haake et al. 2002):

"Once the joint planning (JPL) process model was finished, one designer created a work package model template in the METIS modeling environment and made the template available by using the shared repository[...] Subsequently, she put a link to the template into the JPL process model. Now, another designer used that template to create a sample work package model, by using modeling services. This model was reviewed by the first designer and improved in a number of iterations. The final example model was made available in the shared repository and linked to from the JPL process model. This mixture of largely asynchronous work and some synchronous discussions was greatly facilitated by the shared repository, collaboration, and modeling."

The template produced here is typical. It includes a basic structure for objects, with separate folders for tasks, inputs, outputs, organizations and people, as well as a project document archive. Some elements, e.g., parts of the archive and the organizational structure, are shared among the work packages. The inputs to one WP in many cases are the outputs of another.

This example shows how (meta)process support can facilitate knowledge management. XCHIPS was also used for enacting the process of *defining new projects* in this version of the infrastructure, invoking METIS to let users define the first plan of the project and then forwarding it to WORKWARE. However, real-time collaboration met technical difficulties with firewalls and limited bandwidth across the Internet. Consequently, for version 2 of the infrastructure, a web-based solution replaced XCHIPS for project definition.

Evaluation Results

The QPR and JPL cases were subject to a formal evaluation where 10 people answered a questionnaire (Chrysostalis et al. 2003; Krogstie et al. 2002b; Lillehagen et al. 2002b; Scagno 2002). The same questions were asked after the first period, when none of the EXTERNAL tools had been used, and then again after the second period, during which the infrastructure had been in use. For the reporting case, the time spent, perceived quality of results, and the need for outside help or documents showed great improvement (Scagno 2002). Part of this improvement could be due to learning that would occur anyway from the first to the second cycle. However, a baseline survey of the similar process of *Summary Cost Statements* showed less improvement than QPR.

For the planning case, opinions were more mixed. Some of the respondents felt that quality and effectiveness had improved, while others claimed the opposite. A clear majority however thought that the plans had become more accurate. When asked what the most important problem was in planning, half of the respondents originally said lack of collaboration. After having tried the tools, however, all but one chose *identifying dangerous delays*. It was also reported that initial experience shows that the current infrastructure and tools are too rigid (Haake et al. 2002). While the numbers from this survey clearly are too small to draw statistically significant conclusions, the relative results of the two cases, and also for different criteria, are interesting. The opinions by the participants were more clearly articulated (both positive and negative) after tools were applied. Apparently, real-time cooperation was not as important as we thought, while simple enactment support seemed more useful. For the further experimentation, it was thus decided to add more work performance orientation to the planning case as well. Experiences from this are reported later.

Action Lists – Emergent Project Planning

The first implementation of the JPL process took a top-down perspective, where managers were responsible for planning the work inside their work package. Such plans, however, seldom are detailed enough to cover all the tasks that are to be performed. Consequently, the EXTERNAL project also had a web-based action list located at the project web server. This solution had a number of limitations, typical of publish-oriented web environments:

- Only the project manager could change the list, update status, add new actions, etc.
- The actions lacked context and were often hard to comprehend.
- The actions were not explicitly connected to project plans.
- Actions were not linked to a work environment, documents, or tools.
- Although the list could be sorted on different attributes and filtered according to certain criteria (e.g., one list for each person), it was not possible to add new criteria.

Consequently, the action lists were not actively used by many of the project participants. During the spring of 2002, it was thus decided to replace them with the EXTERNAL infrastructure. WORKWARE had the central role in this application, managing the actions as tasks. It took just a few hours of work to customize a WORKWARE installation for action lists. It organized actions according to these criteria:

- Status, e.g., most lists contain only ready and/or ongoing actions
- Delay

- Work packages
- Teams that are responsible for coordinating interrelated tasks across work packages
- Persons and roles, separating the actions which the current user is responsible for from the ones where she is just a participant
- Follow-up lists, containing all tasks that the current user is customer of.

The increased access to edit actions should make the list more up-to-date. Although the structure for the actions was not connected to a full project plan, teams and work packages provided increased context for the work. Explicit assignment of follow-up responsibility and the ability to look in the event log to see who created the action made each item easier to understand. The old, static action lists contained 288 actions after 2½ years of operation, while WORKWARE contained 131 after just 2 months, even though it was installed during the summer holidays. It thus seems safe to claim that the second application was experienced as an improvement. After the action lists had been available in WORKWARE for a while, however, usage frequency dropped significantly. This happened although consensus was articulated that the application was useful and should be used. A number of factors may have contributed to this decline:

- Lack of project management commitment and contractual obligations to use the system.
- No clearly defined roles were modeled with a consistent set of user-composed coordination services and views.
- Since WORKWARE allows everyone to define new tasks and themselves mark them as finished, the project manager no longer had to perform these tasks. Although this relieved him of some duties, it also gave him less responsibility for following up all actions. For instance, at project meetings, nobody was assigned responsibility of recording new actions.
- A number of major deliverables were completed, e.g., final versions of the tools, infrastructure, and methodology. Several of the most eager users thus no longer participated actively in the project.
- There were technological limitations, e.g., cumbersome document upload. User interfaces and enactment policies for tasks, in general, were perhaps too complicated for simple actions.
- Instability and poor performance of servers may also have discouraged some users. Performance suffered when the action model grew large.
- For a number of situations, e-mails remained the simplest and most used coordination tool.
- In spite of its web and e-mail integration, some users saw WORKWARE as yet another tool added to an already complex user environment.

During the main period of use, however, it was noted on a number of occasions that people sent out e-mails referring to tasks, and pointing to documents uploaded to WORKWARE. This did not occur with the previous application.

This case shows how quickly and easily WORKWARE could be customized to a particular usage need by defining an overall process model (in this case, the WP structure), a menu structure, and some specialized work lists and services. After people started to use the application, further customization was made based on their experiences. The case also shows how bottom-up emergent process articulation can complement top-down project planning and give the organization a more accurate picture of what is really going on in the project.

3.2.5 Case 2: The Business Consulting Project Cycle

The business consulting case involved primary users outside of the EXTERNAL project. The company in question was supported by process modeling experts from one of the EXTERNAL partners. The company had already defined a procedure for how their projects should be executed. This procedure was available on the corporate Intranet, in the form of textual descriptions and informal visualizations. One of the first tasks for the EXTERNAL consultants was thus to model this procedure, known as the *project cycle*, in the EEML language. Local requirements were then collected, and a customized version of the EXTERNAL infrastructure was installed. The users in this case were novices with respect to process modeling and groupware systems, so they selected WORKWARE as their primary tool.

Reuse of Project Templates

In addition to the process model, the template also includes an organizational model with typical project roles, as well as the firm's tools, information repositories, and document templates. The template contains optional items, which are only needed for certain types of projects, e.g., those with a budget larger than a certain amount. These options are currently modeled as normal decisions. However, since many decisions can be made at project startup, modeling them as reuse decisions would simplify the local models. Many of these decisions are controlled by properties of the project, so the potential for automated reuse decisions is substantial.

It is interesting to note that the project cycle mainly defines the administrative work. The actual performance of the project is to be included inside the item *project work*, a subitem of *project execution* at level 3 in the work breakdown structure. This pattern can be expected in a model that represents management perspectives rather than work perspectives, a typical bias in process modeling. It also reflects the fact that administrative procedures are easy to define and reuse without change across all projects, while the core work is dependent on the situated work environment. Therefore, if knowledge management and process improvement are truly to create a competitive advantage, bottom-up core work must be modeled as well.

Security and Access Control

Improved security and multilevel access control was an absolute requirement from the business consulting company. This was the main reason why access control was prioritized for implementation in WORKWARE. A typical project in this company requires these default access rights:

- Only internal participants should be allowed to read and update all documents.
- Employees not working on a project may not have access to project information.
- Only the project manager should be allowed to grant access rights.
- Participants and customers from other organizations should be allowed to read and change documents and plans within their part of the project, but not the others. In some cases, different customers in the same project should not even know about each other. Different customers may have partially conflicting agendas, leading to less than full disclosure of information.

The access and interaction controller of WORKWARE allowed these policies to be articulated at the general level and reused across projects.

Experiences and Evaluation Results

On the basis of his previous experience with Internet tools, the pilot user in this case regarded WORKWARE primarily as a document repository. The concepts of enactment, work management, and status reporting were not useful to him because in the first project, he was the only participant. Consequently, the system was regarded as too complex and cumbersome to use. This initial reaction indicates that simpler user interface components and enactment policies should be the default for novice users. Though some simplifications were made as part of the customization process for this case, they were insufficient. The EXTERNAL process modelers were able to reconstruct the project cycle template using the available constructs in EEML. In some cases, however, limitations of the tools and errors in the documentation prevented them from achieving what they wanted. One example was the modeling of template actor roles. The documentation, for the version at that time based on atomic semantics, stated that resource roles could only be modeled inside tasks, whereas they wanted to model the roles independently. When this confusion was cleared and the semantic holism of the modeling language was described, the template was adjusted.

3.2.6 Case 3: IT Consulting in an SME Network

The final case study in EXTERNAL aimed to support a network of small- and medium-sized IT companies located in different countries, mainly in eastern and southern Europe. Many of these companies are owned by the same group and have cooperated in a number of projects. Three cases with different characteristics were selected (Giotopoulos et al. 2001):

1. Proposal submission for government funding based on a simple and well-defined procedure
2. Software development subcontracting based on a case of medium complexity
3. Management of a Leonardo DaVinci project based on creative and unstructured activities

An overview of the characteristics of these scenarios is presented in Table 3.1.

Table 3.1. Characteristics of different SME network scenarios

Property	Proposal submission	Software subcontracting	Project management
Main objective	Flexibility	Maintainability, reliability	Reliability, adaptability
Duration	Single unit	Long-term alliance	Temporal
Topology	Fixed structure	Dynamic	Mixed
Participation	Single alliance	Multiple alliances	Multiple alliances
Coordination	Tree structure	Tree structure	Star structure
Visibility	Single level	Multiple levels	Multiple levels
Collaboration	Activity coordination	Distributed process management	Joint resource management, cosupervision

Process and Model Diversity

It is interesting to see how these differences manifest themselves in the process models. Table 3.2 shows the number of primary objects of each category in the models of the three cases in this study. For the first two cases, we clearly see that the increased complexity of the cases is reflected in the size of the models. The project management case, however, has a rather simple model. The reason for this is partly that more work has been devoted to studying the two simpler cases, but it may also reflect that project management is harder to articulate than administrative work. For case 3, just the management activities were articulated and not the core work.

Following the history of these cases, it was interesting to note that software subcontracting, the most elaborate case, was originally modeled as a copy of the project cycle from the business consulting use case described earlier. This template was generic enough to be transported to another country and application domain. The fact that the participants in the SME networks had limited previous experience with process modeling also helps to explain why they would rather start with a template than from scratch. Over a couple of months, however, the software subcontracting model evolved, and new items were added to all levels of the work breakdown structure, and existing items were renamed to fit the local terminology. Here, we saw the process of template *appropriation* in practice.

The project management case was modeled as two separate processes, one for the work before the project actually started, and another for the management activities to be carried out during the project work. This modularization makes it easier to reuse the latter process, as management is an ongoing activity that recurs many times throughout the lifecycle of the project.

Table 3.2. Statistics for models of different SME network scenarios

Property	Funding proposal	Software subcontracting	Project management
Number of work items	25	80	10
Depth of work breakdown	3	4	1
Number of actor roles	4	25	9
Number of object/tool roles	0	19	18

3.2.7 Final Evaluation Results

One year after the survey discussed earlier, all the three EXTERNAL cases were subjected to a joint evaluation (Chrysostalis et al. 2003). A questionnaire was sent by e-mail to 19 users, including managers and project participants. They were asked to rate how much they agreed to statements (both positive and negative) on a 7-point Likert scale, and in-depth interviews of some of the participants were carried out. Frequency of use, user-friendliness, and the usefulness of provided functionality were assessed. In general, inexperienced users responded neutrally to all categories of questions. People who had used the tools were typically slightly positive, giving average ratings between 4.8 and 5.6, where 4 is neutral and 7 is maximum.

The major innovative contributions created by these use cases are as follows:

- Developing the active knowledge model-configured infrastructure, work arenas, and workplaces.
- Developing executable emergent work processes as task patterns, and supporting continuously improving work processes.
- Developing the understanding of the importance of capturing work-centric knowledge elements by inventing common visual solutions modeling language and approach.

3.3 Experiences from ATHENA

In the ATHENA Integrated Project, subproject B5 (ATHENA 2007), interoperability in industry and collaborative business, c-Business, was assessed and validated using six concrete business use case pilots. All six pilots were designed using various infrastructures, knowledge architectures, and methodologies as developed in the project.

The pilots were quite different with respect to approaches, infrastructures and methodologies applied, and service platforms configured and operated. The services therefore do not constitute a homogeneous set of services from one approach for developing and configuring and using Service-Oriented Architectures to build and operate the pilots. There are at least three distinctly different approaches to developing the pilots.

The pilots built to prototype ATHENA components and services were as follows:

- The automotive pilot at CRF, focusing on the testing of car systems

- The aerospace pilot at EADS, focusing on engineering change management of aircraft landing gear
- The furniture pilot at Aidima, focusing on the exchange of information and data among the key stakeholders and the decision support given
- The telecom pilot built at Intracom, focusing on model-configured workplaces for product managers, supporting Product Portfolio Management (PPM) services
- The IV&I pilot, focusing on Inventory Visibility, built by a group of partners coordinated by AIAG/NIST in the United States
- The Outbound Logistics pilot, focusing on part identification scheme interpretation, built by a group of partners coordinated by CAS AG

We have selected to present the telecom pilot at Intracom as a generic use case prototype illustrating the approach, architectures, and methodologies that contributed the most to realize the AKM approach and the visual solutions modeling methodology.

3.3.1 Telecom Pilot

The telecom pilot is about PPM where support for selecting the right product to produce and deliver is in focus. The pilot was developed at Intracom in close cooperation among partners developing model-configured solutions.

The pilot focused on PPM and product data sharing among key actors inside a telecom company. Charged with the task of selecting the right products and product variants to produce products for a dynamic market and customer base, the company must find new ways of managing product design and engineering and supporting customer communications. The pilot was implemented using a model-configured, user-composed platform and services (MUPS) architecture to design a service layer with roles, views, and model-generated workplaces and services, focusing on the needs of the product manager.

The technicalities and architectural details of these technologies and how they are applied to ATHENA results are described in more detail in ATHENA deliverable DB5.3. Here, we limit our discussion of what these technologies contribute to, stating that CBP (Collaborative Business Processes) is a top-down approach to put more coordination into work processes. MDA is a middle-out approach to provide mapping and transformation services to data, messages, and work processes. SOA is a bottom-up technology to reengineer legacy systems and provide generic components. Finally, the MUPS approach combines all the three, and adds role-specific services, reflective views, and collaborative context.

Pilot Purpose and Architecture

The telecom pilot was intended to test the applicability and usefulness of the ATHENA interoperability solutions in a typical industrial scenario, developing and maintaining a variety of high-technology product lines in a sector characterized by highly competitive and rapidly changing market conditions. The use case scenario is based on the process of PPM. Starting from enterprise modeling constructs, web-based workplaces are created, offering navigation and work views supporting all operational tasks as depicted in the enterprise model. Interaction with enterprise information repositories is facilitated via an underlying service-oriented architecture.

PPM is a process involving decisions made at different levels, where the company's active products list is constantly updated and revised. New products are evaluated, selected, and prioritized, and existing ones may be accelerated, killed, or reprioritized. The objective is to allocate resources in a way to maximize sales and profits and minimize risks.

PPM is of significant importance especially to a large enterprise with many business units and complex products. Product design typically requires the collaboration of many different engineering teams inside the same corporate environment, as well as within a business network of collaborative enterprise. PPM involves many business processes (new product development, product management, supply chain management). Many actors in different roles from strategic level (e.g., business unit manager, sector manager) and tactical level (product manager, project manager) to operational level (team leader, engineer) will be involved.

The focus of the pilot was on product management and on supporting the role of the *product manager*. The product manager is assigned to a product or product family and is responsible for developing or overseeing all aspects of the product including product definition, product development, product launch, current product management, and product phase-out.

It is apparent from Fig. 3.5 that PPM and particularly the role of product manager (PM) entails access to information of varying nature, stored in various systems and diverse platforms and implementations. The evolution of such systems in any typical medium-size enterprise has been typically outside a planned framework of interoperable systems, a situation that is changing lately with the introduction of service-oriented platforms and architectures. It is therefore currently difficult for the PM to access those disparate systems and retrieve the information needed in a comprehensive and user-friendly presentation mode.

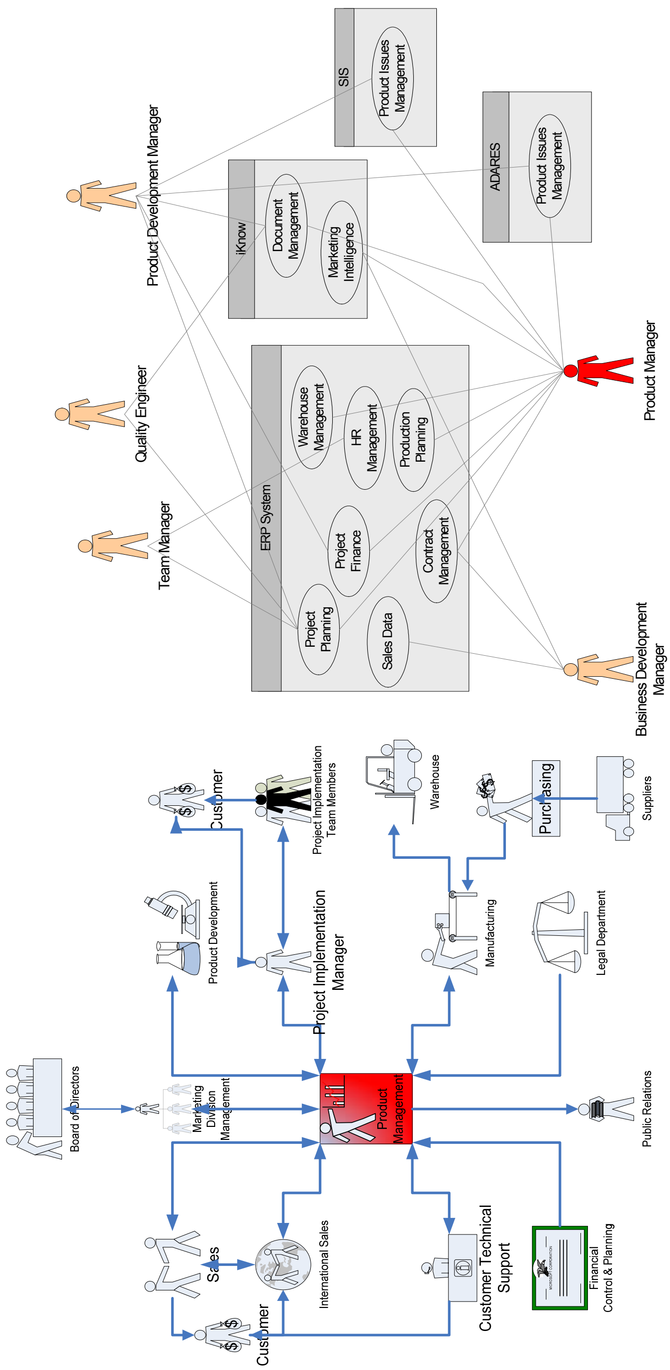

Fig. 3.5. A high-level overview of the as-is situation in the telecom pilot

In the architecture (Fig. 3.6), most of the enterprise modeling takes place in the *business* level/layer. The *process* layer represents model-generated workplaces that result from the enterprise modeling and are presented to the user as interfaces for their tasks. The *services* layer encompasses all services that are used or created to support the workplaces.

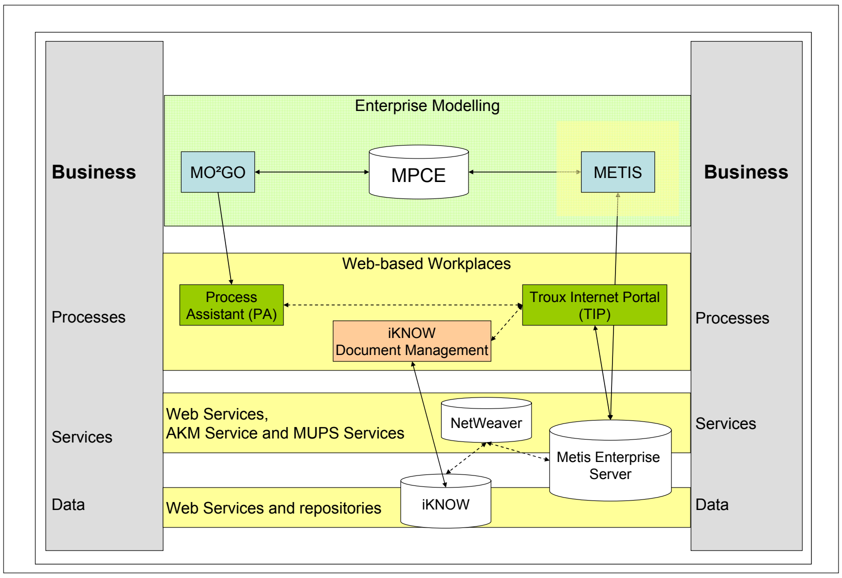

Fig. 3.6. A high-level overview of the architecture of the telecom pilot

The following ATHENA results and tools have been used to support the telecom pilot:

- *POP** – for modeling the different aspects of the enterprise and generating the workplace through the models
- *Import/export of POP** – for modeling different aspects of the enterprise
- *MPCE* for supporting interchange of models of different aspects of the enterprise in different tools
- *Transform ITM and BPM models to MEAF models* for modeling different aspects of the enterprise
- *MEAF ATHENA extensions* to facilitate web services, task management, and user interface modeling
- *MGWP (PA + TIP*) during the generation of the workplaces through the models
- *MOOGO* for process assistant (PA) generation
- *Metis* for Troux Information Portal (TIP)

- *MGWP TIP Services for Web* Services for discovery of web services and linking them to the models
- *Johnson* (+ Lyndon) for design, testing, and deployment of services

How to Use the Services

The services and tools used in the ATHENA telecom pilot are organized as shown before in the architecture figure and are generally presented along the six steps that are shown in Fig. 3.7.

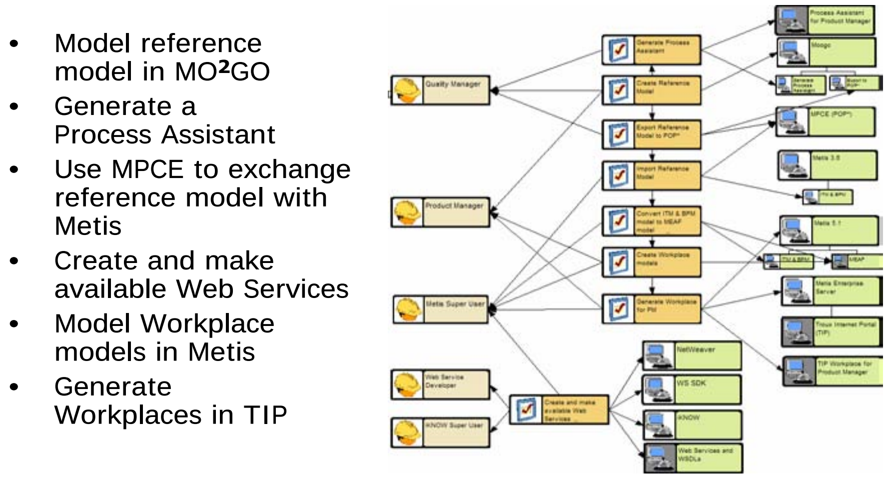

Fig. 3.7. Steps of using ATHENA services in the telecom pilot

Step 1, modeling the Reference Model in MO^2GO is a preliminary modeling step that results in the enterprise reference model in MPCE. The Process Assistant Workplace can be automatically generated in step 2.

MPCE is used to exchange the reference model with the Metis tool where additional modeling is performed to instantiate constructs of the Instance Model relevant to a user role. In the telecom pilot, this role is the product manager. Before the TIP workplace can be generated with any meaningful usability, web services that interact with the data layer and retrieve information from enterprise repositories must be designed, tested, and made available through metamodels built in Metis. When the web services are created, they are deployed in a service infrastructure accessible by Metis. Metis is used to import the available web services, instantiate MUPS services, and finally generate the TIP workplace, which is the actual GUI the PM uses to perform his work.

Guidelines and Experiences

Present experience with the development of Model-Generated Workplaces in the PPM scenario strongly supports the effectiveness of the approach for creating *role-specific application spaces* that can support day-to-day tasks while adhering to the corporate standards captured in the enterprise model. It has been clearly shown that the generated workplaces can go far beyond a traditional application development approach. Future directions for extending the MGWP concept so that it can evolve into an end-to-end interoperability solution framework may be inspired by the following observations and challenges:

- The actual design and implementation of underlying-information-retrieving web services does not stem from the enterprise models. The services preexist and are currently *linked* to appropriate tasks in the models. This creates a *boundary* or *mismatch* between the model-generated workplace, which is formally created, and the services supporting it. This is not a problem in services supporting document-related tasks, which can be uniformly designed, but it *is* a problem in all other types of data services. The main issue is the need to *match* constructs resulting from two very different processes at the boundary of the enterprise model and the existing (legacy) systems.
- Even more so, TIP User-Composable Services (MUPS) presuppose the existence of a *complete* set of visible services or the ability to transparently combine and orchestrate existing services to support all tasks modeled and supported by TIP. This is not a workplace task, but is a barrier in the effective utilization of such a workplace. Further efforts should identify ways of formally creating a description of this *complete* set of services needed and expected by the workplace.
- Deriving views for each role, which is the last step in the modeling procedure, is roughly defined in ATHENA. There is no current best practice, guideline, or mechanism that identifies how to derive the role-specific view from a model *automatically*. A preferred approach is probably the customization of appropriated services (e.g., web services) based on the enterprise model on demand. Subsequently, the model has to contain all necessary information to customize the right service. This means a high level of granularity and increased model complexity. Some solution to manage this issue has to be developed.
- Model-configured solutions can facilitate tangible knowledge sharing across roles and disciplines involved in PPM, only if actual product structures are captured in the models, so that workplaces can be adapted to them. Recent extensions to the Metis/TIP web service plug-in are

intended to support importing of large data structures from XML business documents (e.g., Web service results) so that they can be mapped to model concepts and imported automatically. The workplaces can also update legacy system data by replicating the tasks that invoke updating web services for each component, element, or parameter in the product models.

3.3.2 Conclusions

As a general conclusion of the usefulness of the ATHENA results, one pilot builder stated as follows: "The functionality provided by the services often is not comprehensive enough. In future work, the prototyping services should be enhanced step by step to provide a full set of functionality for practical applications."

Another pilot builder stated: "What you do not design for you will never get, implying that interoperability must be designed for."

However, the conclusion from all six use cases was that the approaches to interoperability developed in ATHENA have proven to be good approaches and working methods to reengineer the systems that are deployed and operational. The challenge is rather to use the AKM and SOA approaches to find new ways for designing future systems.

There will always be a demand for reengineering services, as demonstrated by these pilots, to extend and adapt operational platforms and services. The IT system providers will produce new approaches and methodologies to allow a wide variety of adaptable and extendable customer services for industrial users and partners to build their own operational workplaces. However, for markets and application areas where stakeholders and users are not easily involved in operations or maybe not even available we will have other approaches and methodologies for developing systems and operational solutions.

The pilots belong to three distinct types of operational industrial solutions:

1. Global horizontal peer-to-peer systems: characterized by repeatable business objects and clusters of services, where interoperability can be achieved by *bottom-up* resolution, model-configured services, model-mapped semantics, standards, and logistics alignment, prototyping and supported by the results of the ATHENA project
2. Collaborative business process systems: characterized by a service-oriented architecture middle layer, integrating services across various legacy systems, where interoperability is achieved by *top-down* model-driven architectures (MDA), standards, and services

3. Collaborative product and process design (CPPD): characterized by process flows being designed as task patterns, and related to add intelligence to product structures, where interoperability is achieved middle-out, through model-configured, user-composed layered service architectures

Most global, multinational corporations have a need for all of the three approaches. The common glue is a set of web services for project portfolio management, services management, and document management, all integrated in an Active Knowledge Architecture (AKA) and made accessible through a service-oriented architecture.

Future systems will be designed by employing more services to involve stakeholders and support user interaction, so we will be *designing for interoperability*, supporting continuous team learning, and life-cycle knowledge harnessing for reuse. Active business knowledge will drive system development and configuration of workplaces and services.

Certainly trying to implement interoperability in deployed traditional IT systems that have been in use for some time is not easy and will not give agile solutions or support design and services evolution. Current operational IT systems, sold off the shelf, suffer from a dramatic loss of stakeholder contextual knowledge in their life cycle from specifications to delivery and to reengineering or demolition. Service provisioning is a step in the right direction, but support for users to configure, adapt, and manage their own services and workplaces is needed. As seen from the experiences from EXTERNAL, the user interface to support such service must be highly usable.

3.4 Summary

In this chapter, we have recapped experiences and lessons learned, particularly from 1998 to 2006, through the application of early versions of the AKM technology in the EXTERNAL and ATHENA projects. The intent has been to shed light on the evolution from off-line modeling of smart infrastructures, including MDA to inline modeling for building AKAs, automatically configuring workplaces and visual arenas for work execution.

With the introduction of the IRTV visual modeling language, refinements of the AKA structures and contents, and implementation of design methodology components the first AKM arena for AKA-configured workplaces was ready for delivery.