LIFE CYCLE SUPPORT FOR COLLABORATIVE ENGINEERING AND OPERATION OF VIRTUAL ENTERPRISES

Weigang Wang, Jörg M. Haake, Jessica Rubart
IPSI – Information Publish and Systems Institute
GMD – Germany Research Center for Information Technology
Dolivostr. 15, 64293 Darmstadt
+49 6151 869917
{wwang, haake, rubart}@darmstadt.gmd.de

ABSTRACT
Current information technology for virtual enterprise modeling, computational simulation, process enactment and cooperation support is fragmented. Our integration strategy is based on a dynamic process-centric enterprise model that is presented and manipulated as component-based cooperative hypermedia. The resulting system, XCHIPS, is accessible on the Web and provides life-cycle cooperative engineering support for visual modeling, computational simulation, and enactment of process-centric enterprise models. A use case is presented to highlight the needs and to show how such needs are addressed with our approach.

Keywords
Virtual enterprise modeling, cooperative hypermedia, process support, software components

1. INTRODUCTION
Current trends of globalization and increased competition require new forms of business organization and support. Virtual enterprises, i.e., networked organizations characterized by dynamic, flexible and temporary cooperation between diverse business partners are emerging to exploit such fast-changing market trends.

1.1 The Needs
In a virtual enterprise, business processes are often carried out by a team in a collaborative way. As a consequence of the dynamic nature of virtual enterprises, teams in virtual organizations are increasingly distributed and constantly changing. In addition, the business processes to be executed are changing in our global economy. Together, these trends require that teams in a virtual enterprise

(1) create and learn a process-centric enterprise model collaboratively,
(2) perform computational simulation to the models for process improvement,
(3) execute the business processes defined in the enterprise model and carry out the work following the processes collaboratively, and
(4) adapt processes to the needs of the team and the actual situation.

Thus, a system is needed to provide cooperative engineering and operation support for the life cycle of an extended enterprise: from cooperative modeling, to computational simulation, and to execution of a process-centric enterprise model. Moreover, it should allow team workers to move back and forth in these phases to manipulate a shared enterprise model so as to adapt the model to their changing needs.

1.2 The problem
Information Technology for virtual enterprises has been consistently recognized as both a key challenge and an emerging opportunity by many enterprises that are striving to expand their market share, revenues, products, and services. For example, Information Technology for Virtual Enterprises was ranked first in an unofficial survey conducted by Manufacturers Compatibility Committee in 1998 [6]. However, current information technology for virtual enterprise engineering is fragmented. Process and situated knowledge is captured in disjoint process models. Tools for enterprise modeling, process computational simulation, process enactment, and cooperation support are often developed in separate environments using varying technologies. Most projects for virtual enterprise engineering seem to develop a static system framework with hard coded static templates for creating views of a virtual enterprise. To succeed a virtual enterprise engineering approach has to converge fragmented methodologies, technologies and support integrated dynamic enterprise models.

1.3 Our Approach
Our integration strategy to overcome the above problems is based on a dynamic holistic process-centric enterprise model that is presented and manipulated as graphical hypermedia structures. Our technical approach is to integrate workflow, computational simulation, and task-specific computation support into component-based cooperative hypermedia. The approach can be outlined as follows:

• Adopt a visual enterprise modeling approach for creating dynamic and holistic enterprise models;
• Use advanced hypermedia structures, i.e., extended links and composite nodes, as is required for graphical enterprise modeling;
Integrate process support into the advanced hypermedia structure as is required for supporting collaborative process execution;

Use the authoring and cooperation capabilities of a cooperative hypermedia system to support cooperative modeling, changing, and execution of business processes;

Integrate a computational simulation tool for process improvement;

Use groupware components and external applications for task-specific computation support;

Allow new object types and new groupware tools that manipulate these new object types to be incorporated dynamically at run time.

Provide a graphical hypermedia based schema editor for creating and adapting domain-specific schemas;

Provide a graphical hypermedia browser for creating, modification, simulating, and execution of enterprise models based on selected schemas;

Make the cooperative hypermedia-based process support accessible on the Web.

The remainder of the paper is organized as follows: In section 2, related work is discussed. Section 3 presents our approach to address the identified problems and needs. In this section, our hypermedia-based representation of dynamic holistic enterprise models is presented. It also explains how dynamic aspects and computational support for process enactment, process simulation, and specific task are integrated into the hypermedia-based system. In section 4, implementation details are presented. In section 5, a use case is presented to highlight the identified needs and to show how such needs are addressed with our approach. The paper concludes with a summary and plans for future work.

2. RELATED WORK

2.1 Enterprise modeling and enterprise models
Humans prefer to abstract only the part of reality which may be relevant to them. Due to this tendency, most of the modeling approaches have separate models for different aspects. This may not allow us to have a ‘big-picture’ of a system and it is difficult to move from one model to the other [18]. The resulting problems include lack of integration, islands of automation, sub-optimization of resources, ad-hoc development, and lack of a well-defined architecture [24]. A single model will not be able to capture all aspects of an enterprise. Therefore, a complete model composed of different views, aspects, or perspectives is needed [18]. A serious challenge is that such an integrated model should be flexible and must be robust to adapt to changing environments without losing focus [11]. There appears to be some consensus that there is a need to develop a generic approach to build enterprise models, which can represent the entire enterprise while simultaneously allowing each function of the enterprise to maintain its uniqueness [11].

An integrated enterprise model can be defined as a holistic model of the enterprise and its aspects [19]. A holistic model does not mean that the model contains "everything", but that the model covers the most important aspects of the enterprise [10, 19]. Generally, the most important aspects of an enterprise are [8, 19]: information aspects (goals and strategies, products, target market and customers), organization aspects (organization chart, projects, people, and their roles), information systems (infrastructure, applications and data model), tasks and processes. The purpose of an enterprise model is to provide a better overview of the enterprise, chart the value chains and thereby contributing to increased efficiency and improved quality.

The most important thing when modeling or studying the different aspects is that they are parts of a holistic model, and that they play a larger role than might seem when looking at them isolated [19]. A natural ingredient in holistic enterprise models is process orientation. The processes are the glue that holds the enterprise together. For instance, work processes can be linked to goals, strategies, requirements, organization structures, individuals, products, tools and infrastructure. Process orientation means that the processes must be used consciously in planning, performance and control of both everyday and strategic tasks [19].

A dynamic enterprise model means that the model is reflective and regularly or preferably continuously updated. The modeling should preferably be supported by visual libraries of model templates, hierarchies of meta-data, and a guiding methodology [10]. The objective of enterprise modeling is to help people understand, communicate, develop solutions to business problems through use of dynamic enterprise models.

2.2 Information technology for virtual enterprises
There is a wide range of information systems aiming at or having to potential for supporting virtual enterprises. These systems include workflow systems, graphical modeling tools, simulation and analysis systems, ERP (enterprise resource planning) systems, and Web-based hypermedia systems.

Workflow systems focus on automated process support. They generally do not manage information objects and their relationships that are needed for creating a holistic view of an enterprise. Web-based workflow systems address some needs of virtual enterprises. However, confined with the capability of HTML and web browsers, they often provide users only with a “ToDo” list and task descriptions [4], or
Enterprise modeling and analysis tools focus on modeling and simulation. Metis is a powerful visual modeling tool for creating holistic enterprise models [10]. Vite is a computational simulation tool, which takes administration issues into account to provide more accurate simulation results [14]. KBSI [9] support modeling and analysis. System Architect 2001 provides a set of diagrams for capture the entire enterprise from various business perspectives. Cooperative modeling and process enactment are not supported by these systems.

ERP systems focus on enterprise resource planning and information management. But most of these systems are costly and used within one organization. When integrated with workflow and project management, they can provide comprehensive support for enterprise engineering and operation. To broaden their availability, current trend for ERP and project management systems is to adapt Web-based interfaces, integrate project management with strategic planning, and incorporate collaboration and communication tools [17].

The CHIPS system developed in our previous work provides hypermedia-based support for cooperative process definition and execution [15, 20]. However, in the past it had not been applied and extended for creating holistic enterprise models. It had not been integrated with a process simulation tool, and it is not accessible on the web.

In summary, current information technology for virtual enterprise is fragmented. Most systems are built using separated model, rather than an integrated model that cover all the important aspects of a virtual enterprise. There is no cooperative engineering support for the whole life cycle of a virtual enterprise. In this scheme, a model is developed by progressively specify the five views: information, activity, resource, process, and organization views. The Information View defines objects interested in a particular domain and their relationship (business logic and/or ontology). Based on the information view, the Activity View defines the functions performed by the enterprise (what is done). The resources and capabilities managed by the enterprise are defined in a Resource View. The resource view defines two basis aspects: the resources necessary to accomplish activities (and thereby acts as a capability model) and how those resources are organized by the organization. To form a resource view, actors and the process they perform are coupled. The Process View defines a time-sequenced set of processes (how it is done). Finally, the Organization View is used to define how the enterprise organizes itself and the rules governing how it manages itself and its processes. Many of the entities (typed objects) defined in the other views also participate in the organization view. However, several entities have been defined specifically to support the organization view, including planning objects such as goals, plans, policies, and measures. The roles and position agents in the enterprise play are identified. These are then linked to each other, activities, and planning objects through a set of management and organization links.

### 3.2 Hypermedia representation for holistic and dynamic enterprise models

Our hypermedia-based representation has three levels of abstractions: hypermedia meta model, hypermedia schema, and hypermedia document. In the next sections, models in these three levels are described. Then, the tailortability and extensibility of these models are presented.

**Hypermedia meta model**

The hypermedia meta model contains base node and link types that are used to derive domain-specific node and link types for creating the above mentioned five views. Our hypermedia meta model includes four categories of base object types: atom node, composite node, reference node and independent link. An independent link specifies a relationship between two nodes. A reference node has an attribute referring to an object or a URL. Each object in these four categories has a semantic type and a graphical view. A composite node may contain a collection of other nodes. An atom node is an object that does not contain other nodes. In addition, each base type in these categories may have its own attributes and methods. The graphical view for a node contains an image, a name and other optional widgets whose views dependent on the values of specified attributes of the node. The graphical view for an independent link is a colored, labeled line which may have arrow head(s), may be curved or straight in shape, and may contain the node and link types that are used to derive domain-specific node and link types.
be in different line styles (such as dashed or dotted) and thickness.

Hypermmedia schemas for domain-specific enterprise models

A schema contains elements and relationships between the elements. For different kind of enterprises, such as enterprises in Banking sector and Software R&D sector, there are many domain-specific object types and relationships to be captured in different schemas. A hypermedia schema contains a set of domain-specific node and link types for creating domain-specific enterprise models. The constraints in a hypermedia schema are reflected in the containment relationship of composite nodes, and allowable linking relationships between typed nodes. These domain-specific node and link types may be base types or derived types. A derived type is created by customizing a base type or another derived type. For graphical modeling, it is important to define intuitive graphical views (mainly the icon images) for these types.

Hypermmedia templates and documents for individual enterprise models

Both hypermedia templates and hypermedia documents are instances of hypermedia schemas. A hypermedia document is a composite node filled with contents (i.e., nodes plus their links, and recursively for the contained nodes). A hypermedia template is a special hypermedia document filled with well-designed contents. By cloning a template, a new richly linked hypermedia document can be created without the need to create lots of nodes and links manually. Actually, in our approach, any hypermedia document can serve as a template by cloning it using a copy-as-template function. A new hypermedia document can also be created from blank content by selecting a schema.

Dynamic aspects: tailorableity and extensibility

Tailoring and extending often happen hand in hand. In the reverse top-down order of meta model, schema, and instance levels, when tailoring at one level is short of certain constructs (i.e., object types), an extension at its higher level is required (see [20] for details).

Tailoring at the instance level is to modify object instances under the constraints of their types (or their schemas). When a new type is needed for a hypermedia document, its schema has to be extended (i.e., add a new type to it) so as to affect all the documents created using the schema. At instance level, for flexibility we allow document instances to deviate from their schemas by using object types from a global object type palette. This defers the decision for schema modification or creation to a later stage when a mature pattern emerges.

Tailoring at the schema level is to add, remove or modify node and links types. When a new type is needed, it can be created by customizing a base type or another existing type. If no base type or existing type is suitable to customize, an extension has to be made at the meta model level (i.e., to add a new base type).

The meta model level extension can be done by dynamically loading and registering a new object type (i.e., a customized instance) from a selected object class at run time (supported by Java and DyCE, see implementation part). When no class is suitable, an existing class has to be modified (for instance, to add a new method) or a new class has to be added (see implementation part).

3.3 Computational support for workflow, simulation, and specific task needs

Our approach to process support is to incorporate task-related attributes, such as state and time, into hypertext nodes, and incorporate control flow and data flow semantics into hypertext links [5, 10]. Thus, a process is represented as a set of hypermedia task nodes connected by process links. More specifically, a few base node and link types are created for the purpose. These base types include an atom task node, a composite task node, a reference task node, and a flow link type.

Our process meta model is a combination of the CHIPS [15] process model and the Vite process model [14]. Vite is a computational simulation tool. Its simulation results are presented in various statistics diagrams, such as PERT/CPM chart and workload Backlog diagram. The Vite model contains object types and attributes for modeling projects, processes, and organizations. To support simulation directly from our hypermedia based graphical process definition; we incorporated all the Vite object types into our process model. For example, the attributes needed for Vite "Activity" and "precede" types are represented in the attribute list of derived types of task nodes and flow links, respectively. Additional derived node and link types are created for the Vite "Start", "Milestone", "Meeting", "Actor" object types and its "supervise", "assign to", "change" (re-work), and "coordination" (information exchange) relationships.

Task-specific computational support for activities in a process is realized using specific software components. The shared data model of a component can be used directly in a hypermedia structure as an atom node or indirectly through a reference task node. When a reference node is opened, a window for the component of its referred object will appear. If the referenced object is a URL, then the content will be displayed in a web browser. There will always be a need for new task-specific software components and we can use them indirectly through task reference nodes without a need to change object types for process definition.
3.4 Methodology and tools for VE engineering and operation

Modeling is a major element in enterprise engineering. Enterprise engineering deals with the analysis, design, implementation, and operation of an enterprise [8]. Methodology provides procedures to guide users to construct and utilize models using the tools supporting the methodology. Our methodology is an extension to the above-described modeling scheme. The extension is made to cover operational support of an enterprise model including computational simulation, process enactment and cooperation support.

The life cycle of a virtual enterprise has four major phases: initiating, engineering, maintaining, and dissolving. These phases and their underlying engineering methodology are outlined as follows:

- In the initiating phase, a business opportunity (i.e., a custom need) and partner organizations that have complementary expertise and resources to exploit the opportunity are identified. Also, a proper modeling template is selected to integrate the relevant parts of partners’ resources and business processes into an initial enterprise model.
- The engineering phase tailor the initial model to meet the specific need of the virtual enterprise. In this phase, detailed model that covers all the aspects of information structures, process structures, resource and organization structures are created.
- The maintenance phase organizes, analyzes, refines and activates the enterprise model to guide the completion of the work in the business processes and customer delivery.
- In the dissolving phase, templates and solution components are stored and can be re-composed, re-engineered and reactivated for new virtual enterprises.

In this way, we can provide cooperative support for the whole life cycle of virtual enterprise engineering and operation.

Our method for creating a schema and template is to:

- identify domain-specific objects and their relationships and capture them in a domain model,
- create a hypermedia template for the identified domain model by creating and using node and link types for the identified domain objects and their relationships.

The object types can be created progressively for information view, to activity view, resource view, process view, and finally the organization view. The sequence is not necessarily in that order. A complex domain model can only be created by using both top-down and bottom-up approaches and back and forth in iteration. The hypermedia schema can be automatically extracted from the hypermedia sample or template [21]. Collaboration and reuse may happen at all the above mentioned steps.

Our method for creating a model instance is to:

- clone a template or another model instance,
- tailor the copy by modifying its hypermedia structure and the attributes of nodes and links in the hypermedia structure,
- if necessary, use additional object types in a global object palette or modify its schema by creating and dynamically adding new types to the schema,
- perform computational simulation for process improvement,
- modify the model based on simulation results,
- repeat above two steps until satisfaction,
- enact the process,
- perform computational simulation at any time,
- modify inactive parts of the running process,
- continue the process execution, and upon completion, archive the process, extract reusable model fragments, and index them for future reuse.

Again, tailoring can be done progressively for each of information view, to activity view, resource view, process view, and finally the organization view. Similarly, the sequence is not necessarily in that order. Collaboration and reuse may happen at all the above mentioned steps.

In XCHIPS, the main tools for engineering and operation of virtual enterprises include:

- a graphical hypermedia schema editor,
- a graphical hypermedia document browser and
- a set of groupware tools for informal coordination and communication.

The schema editor allows users to define hypermedia schemas by crafting an example hypermedia structure. New types of objects are created by first creating an object of a selected prototype from the set of base prototypes and then assigning a new type name. For instance, when a node of a new type is created using the schema editor, the system will ask the user for its name and its semantic type (see [ht97] for more details on the example based hypermedia schema definition).

The document instance browser is used to instantiate, modify, activate simulation, and enact processes. With it, users can navigate through the hypermedia based enterprise model and activate tasks ready for enactment. The object type palette of the browser is dynamically configured from the content schema of the composite node opened in the browser.
The above mentioned cooperative hypermedia browsers are groupware tools by themselves. There are also many other groupware tools can used together with them, such as a Chat component and a Whiteboard component. These groupware tools can be incorporated in the tool bar of the browsers or placed into the content pane of the browsers.

4. IMPLEMENTATION

The XCHIPS system is implemented using Java and DyCE [5, 6] (Dynamic Collaboration Environment). DyCE is a Java-based framework for creating mobile groupware components that are loaded on demand. Additionally, it provides dynamically replicated shared data which is displayed and edited by the shared groupware components, as well as transactional support for access to and modification of this shared data. DyCE provides a common task-based programming model for groupware components which allows components to invoke each other and which also enables users to combine components according to their collaboration needs. The XCHIPS collaborative components developed on DyCE can be registered to a server and transferred across the Internet. These components may work as Java applications or Java applets that are executable across the Internet.

4.1 Shared objects and components

In the XCHIPS system, both the hypermedia-based process structure and its associated information are shared objects represented in mobile groupware components. The XCHIPS browser is one mobile groupware component. Other components that are used by it via DyCE’s task-based programming model are a shared notepad, whiteboard, chat, query, hierarchical navigator, HTML presentation and the XCHIPS schema editor component. The shared objects as well as the groupware components themselves are stored on one or more DyCE servers and are replicated to client applications when needed. The shared data can also be imported from and exported to documents in XML as universal format for structured data in order to be able to easily exchange data with external tools.

4.2 Sessions

The mobile groupware components are used in sessions. A session is made up of a collection of people, groupware components, tasks, and shared objects. Working with the XCHIPS browser component in a session, the groupware components used by it can be activated in the same session (thus becoming accessible to all users within this session) or in a new session (providing a simple transition between coupled cooperative and individual work). Users can invite other users to join their working session by activating the Invite user button at the upper right corner of the session window (see figure 1). Users can also use the query tool to search for all the active sessions and request to join selected sessions.

4.3 Hypermedia structure representation

In XCHIPS, the hypermedia structure is represented by composite nodes and extended XML links. Composite nodes are graphically presented as labeled icons and the extended links are graphically presented as labeled arrow lines between the node icons. The common web embedded links (i.e. the HTML links or simple XML links) can be used to point to the content information of a task or associated reference materials. They are presented as link markers with a special icon image. Composite nodes may contain other nodes, links, and information objects. All the nodes have a "closed" view and an "opened" view. The XCHIPS browser has been developed to present and edit the graphical process structure, its associated content objects and HTML links. The state information of task nodes is color-coded in their views (white for inactive, yellow for enabled, green for active and brown for completed). Process execution can be started by selecting "enable" and then "activate" menu operations on the root composite task node of the process. When a composite task node is "activated", the starting tasks of the process structure nested in the composite task node are "enabled". The actors of the enabled tasks are notified. They can then "activate" their tasks. When the work of a task is finished, its actor can select "complete" menu operation on the task node, such that its following tasks (or nested starting tasks if it is a composite task node) are "enabled". A process is completed when all its subtasks are completed.

4.4 The XCHIPS browser

There are two palettes on the XCHIPS browser: a tool palette on the left-hand side and an object palette on the right-hand side of the browser (see Figure 2). The tool palette contains groupware tools and information organization tools. The groupware tools include a user’s guide browser, an example browser, a paste tool, a search tool, a chat tool, a schema editor, and a whiteboard tool. The information organization tools include hypermedia objects for creating notepads, embedded links and folders. The user’s guide browser calls a DyCE task on an underlying shared HTML presentation object that invokes the HTML presentation component on the user’s guide. Similarly, the other icons also invoke collaborative groupware components. An exception to this is the paste tool. This allows pasting a clone of a shared object (previously copied to a clipboard) into the XCHIPS browser. If the cloned shared object is of no hypermedia type; a reference node of our hypermedia meta model is created that points to the cloned shared object. Notepads reference a shared notepad, folders point to shared composite objects.
that again can contain a hypermedia structure and embedded links can reference arbitrary shared objects.

The object palette on the right-hand side contains hypermedia objects used to represent workflow processes. These objects currently are “Start”, “Finish”, “Milestone”, “Task”, “Actor”, and “Meeting” nodes, and “precede”, “coordinate”, “change”, “supervise”, “assign to”, and “participate” links, but can be adapted using the example-based XCHIPS schema editor component.

As mentioned above, the XCHIPS browser is a DyCE groupware component working on a shared object. Opening a node via its context menu results in calling a DyCE task on the underlying shared object of the node. Therefore, again a groupware component is invoked and is displayed either in the same session or in a new one, depending on the menu item chosen. The XCHIPS browser works on a shared composite object containing shared objects of the underlying hypermedia schema. Concerning the four categories of our hypermedia meta model, an atom node directly connects with a shared content object, its open operation will activate a component. A composite node has attached a shared composite object for which the XCHIPS browser publishes DyCE tasks so that again an XCHIPS browser is invoked opening a composite node. A reference node can point to a shared object and can be opened, if there are DyCE tasks published on the referenced object. When opening such a node, the related component is invoked; e.g. the whiteboard component in case of a whiteboard shared object. Additionally, a reference node can point to a URL. In this case, opening the node calls the web browser to open the URL. Finally, an independent link is connected to a shared link object that can’t be opened.

4.5 Extensibility

XCHIPS is easily extensible for end-users. This is because DyCE provides dynamic extension of the working environment in the sense of adding groupware components at runtime that are loaded on demand. Additionally, the XCHIPS component allows adding arbitrary shared objects to its hypermedia structure using reference nodes. Groupware components are also used to work on these objects. Opening reference nodes results in invoking the associated groupware component, so that computational support for specific task needs is provided.

Moreover, XCHIPS utilizes the Java language’s support for adding base hypermedia object types that can be used to derive and compose other hypermedia object types at runtime. Hence, this provides extensibility at the meta model level. Furthermore, XCHIPS uses a prototype mechanism for customizing base types. Prototypes are initialized base type objects that can be used as a template for new base type objects. Therefore, prototypes allow extensibility at the schema level and are used by the XCHIPS schema editor.
5. A USE CASE

A successful proposal brings together 5 organization partners from 3 countries into a CEC project (i.e., CEC IST1999-10091 EXTERNAL project). In order to establish an efficient virtual organization to carry out the project, the partners decided to create a shared holistic process-centric model of their virtual organization. One partner (GMD) has a cooperative hypermedia based enterprise-engineering tool (XCHIPS). As the system is widely accessible groupware, the partners used it to create and maintain a shared enterprise model for the project. One of the schemas in XCHIPS was developed by a cooperation between DNV and GMD a year before the EXTERNAL project. The schema is a merge and extension to the CHIPS process model and the Vite project enterprise model. The schema includes object types for creating information, task, resource, process, and organization views of an enterprise model. As a starting point, this schema was used to create our holistic dynamic enterprise model.

The Java application based version of the XCHIPS system was distributed to all the partners in the middle of February 2000. The installation was automatic and took about 5 minutes. A DyCE server running at GMD site has been used for shared access and persistent storage. The shared access to the server is established by running the XCHIPS system and typing in the IP address of the server machine in a login dialog box. A one-hour training session was provided by two GMD team members. Each partner has two team members participated in the training session. One of the GMD team members served as tutor and he invited all these team members into a cooperative session. In addition to the XCHIPS browser, a Chat tool was also opened in the same session window for informal communication and coordination (see Figure 2). The tutor introduced the user interface of the system and performed a guided tour through a sample model. Then, he demonstrated how to create and modify the hypermedia-based model. Finally, the team members worked together and created a toy project model as a practice.

After the training session, the partners worked with the system synchronously and asynchronously from time to time. They first created a portal page containing objects for an information view of the high-level project-related aspects. These aspects include the “purpose”, the “work packages”, the “deliverables”, the “exploitation plan”, the “resources”, the “partners”, and the “processes” of the project (see Figure 2). They then navigate the model by opening composite nodes or by following links from nodes to nodes to see nested structures and to create objects for other views of the model.

In the “process” composite node, they used process view related object types to create a process-centered project plan. More specifically, they created task nodes and flow links between the tasks to form a high-level process structure. Then they scheduled the tasks, created a role-based organization structure, and allocated roles and information resources to each task using “assign to” links (they are filtered out from Figure 1).

Finally, they modified and created more relationships between the graphical node representations. The relationship was created either implicitly using a spatial hypertext metaphor (i.e., by placing related near to each other) or using a navigational hypertext metaphor (i.e., by creating explicit links between them). One team member wanted to create a link to explicitly relate two “reference” nodes, but he noticed that such link is not defined in the schema. Upon his request, such a link type named “refer to” was derived from the base reference node type, and dynamically added to the schema and consequently appeared in the object palette of the XCHIPS browser.
By that point in time, a quite comprehensive model for the project enterprise was created (see Figure 1 for a detailed view on the process structure for work package 4). In order to get an overview of the process structures and to see who is working on which part, one team member used a search tool and the search result is presented in a hierarchical browser (see Figure 4). In order to improve the model, they activated the computational simulation operation on the top-level process structure. A window of the Vite tool appeared which shows the analysis reports in various statistic diagrams, such as Pert/CPM, Actor Backlog, Activity Cost, Activity Verification Quality and Activity Schedule Growth Risk as shown in Figure 3. Based on the simulation results, they modified the process structure and adjusted the resource allocation. They repeated the simulation and modification cycle several times until they satisfied with the simulation results. Then the project leader of the GMD team started the execution of the process.

In the process execution phase, in order to couple with dynamic changing needs, they performed computational simulation from time to time, and modified the resource allocation to and structures of the part of the process that were not started. Up to this point in time, the process execution and such computational simulation, modification, individual or cooperative work for the tasks in the process continue in the XCHIPS system.

6. CONCLUSIONS
This paper presents a compute support for collaborative engineering and operation of virtual enterprises. This computer support integrates workflow, simulation and task-specific computer support into a cooperative hypermedia system. A use case for applying this system is presented in the end to highlight the needs and to show how such needs can be addressed with our approach.

The key to organization integration is a common understanding of shared knowledge, reflected by a dynamic holistic enterprise model. The meta modeling, tailoring and dynamic extension features of our cooperative hypermedia approach have addressed the dynamic aspects that are needed for making a holistic visual modeling approach applicable. Using our approach, schemas and templates for different enterprise sectors can be defined, tailored and dynamically extended for creating domain-specific enterprise models. Under our dynamic extension framework, new node and link types for creating information, resource, and organization views of a holistic enterprise model can be created by customizing base hypermedia object types. New base hypermedia object types can be dynamically added into the system.

Our approach to process support is to integrate task and process related attributes and computational semantics into hypermedia nodes and links. Computational support for specific tasks is provided by groupware components, which can be dynamically loaded into the system and activated by reference task nodes in a process structure. The cooperative modeling and execution support is based on a shared hypermedia space and a set of cooperative tools implemented on a Java-based cooperative application environment (DyCE).

Based on the shared holistic enterprise model, the Vite simulation tool is integrated into the XCHIPS environment. The XCHIPS system can support team members to perform process simulation at both process definition and process execution phases. This makes it possible for team members to simulate a running process and to change the parts that are not yet started for continuous process improvement. Moreover, the simulation results on completed processes can provide performance benchmarks. The nested process structure provided by XCHIPS allows simulation to be performed on different levels of a process structure for the interest of different roles (e.g., project leaders at the top level and work package managers at middle levels). Our cooperative hypermedia based process support combined with the unique simulation capability of Vite provides a novel round-trip support for collaborative engineering and operation of virtual enterprises.

The Java and XML based web integration and the groupware nature may provide an XCHIPS-like system more extensive market penetration and availability than that offered by many non groupware based enterprise engineering tools. The wide accessibility to a shared holistic enterprise model may also cut training time and confusion of enterprise partners and customers.

The XCHIPS system is the tool we bring into and extend in the EXTERNAL project (IST1999-10091) funded by the CEC. EXTERNAL focuses on the engineering and operation of networked organizations, and the management of process...
knowledge. Next, we will extend our approach to process knowledge management, integrate it with other EXTERNAL tools, and evaluate this approach in three real-world use cases.

ACKNOWLEDGMENTS
We thank Frank Lillehagen for stimulation discussions and Daniel Tietze and Shirley Holst for providing helpful comments on a previous version of this document.

REFERENCES

