Dustdar, S. (2005). Architecture and Design of an Internet-enabled Integrated Workflow and Groupware system. *Business Process Management Journal*, Special Issue on Internet-enabled Workflow, 11:3, pp. 275 - 290.

Architecture and Design of an Internet-enabled Integrated Workflow and Groupware System

Schahram Dustdar[1]
Distributed Systems Group
Information Systems Institute
Vienna University of Technology
1040 Vienna, Austria
dustdar@infosys.tuwien.ac.at

Abstract

In the last decade, bureaucratic organizational hierarchies increasingly have been replaced with flatter organizational forms, bringing together people from different disciplines to form virtual project teams within and between organizations. Distributed virtual teams often are self-configuring networks of mobile and "fixed" people, devices, and applications. The contribution of this paper is to present theoretical and practical foundations of an integrated Internet-enabled workflow and groupware system. This is achieved by first discussing our contribution in context to related work (collaborative technology) and their underlying metaphors. We show that for an integrated workflow and groupware system, the metaphor used has to be considered and chosen carefully, since in an Internet context, most process participants are members of (many) virtual teams and business processes are often cross-organizational. Furthermore, we discuss design requirements for such an integrated system and provide one example systems implementation (Caramba), which we consider helpful in understanding how virtual teams use combined workflow and groupware systems on the Internet. Finally, we conclude this paper by discussing lessons learned and by providing an outlook on our future work.

Keywords: Internet-enabled Workflow, Process-Awareness, Project teams, Context, Groupware

1. Introduction

Organizations increasingly define their business processes as projects executed by "virtual project teams", where project members from within the organization cooperate with outside experts and therefore build a "team", which in many cases operates as a highly distributed team. The project members work on business processes (e.g. Product Development) but often project members realize their work as a project and not necessarily as part of a larger business process. Their work often leads to artifacts (documents, presentations, products, etc.), which need to be shared among project members. Naturally, as with all high value business processes, those are subject to frequent changes (i.e. exceptions to the rule). Business processes have well defined inputs and outputs and serve a meaningful purpose either inside or between organizations. Business processes in general, and their corresponding workflows in particular, exist as logical models, which can be represented by weighted directed graphs. When business process models are executed they have specific instances (workcases). A business process consists of a sequence of activities. An activity is a distinct process step and may be performed either by a human agent or by a machine.

A *workflow management system* (WfMS) enacts the real world business process for each process instance (Aalst and Hee, 2002; Bussler, 1999; Schäl, 1996; Craven and Mahling, 1995). Any activity may consist of one or more tasks. A set of tasks to be worked on by a user (human agent or machine) is called a work list. The work list itself is managed by the WfMS. The WfMC (2003) calls an individual task on the work list a work item. Several

---

[1] The author is also co-founder and chief scientist of Caramba Labs Software AG, http://www.CarambaLabs.com

process-modeling and enactment systems have been built in industry as well as in academia in order to provide a sustainable infrastructure for business process support and workflow enactment. However, the obtained results are mixed. A recent research survey investigating the usage of Groupware and Workflow Management Systems provided some evidence that Groupware helped 57% of the surveyed firms and Workflow Management Systems were used by 20% of the organizations for their work (see Dustdar, 2003 for more details). Most current WfMS focus on automating structured (modeled) intra-organizational business processes (e.g. Aalst and Hee, 2002; Dayal et al. 2001).

Groupware (Ellis, 1998; Ellis 2000) on the other hand, typically does not contain any knowledge or representation of the *goals* or underlying business *processes* of the group. In recent years there have been a number of attempts to merge workflow-, groupware-, and business process modeling technologies. Industrial research labs and product teams (Bolcer, 2000; Bentley 1997; Caramba Labs, 2003; Casati et al. 2001) have made significant steps forward. Future collaboration systems aiming at supporting virtual project teams focus on integrating workflow and groupware (e.g. Nagypal et al., 2001), covering inter-organizational activities and processes including product value-chains on the Internet (Chen, 2001; Puustjärvi and Laine, 2001) regardless of location (mobility) and regardless of devices used.

The **key contribution** of this paper is to present theoretical and practical foundations of an integrated Internet-enabled workflow and groupware system. This is achieved by first discussing our contribution in context to related work (collaborative technology) and their underlying metaphors. We show that for an integrated workflow and groupware system, the *metaphor* used has to be considered and chosen carefully, since in an Internet context, most process participants are members of (many) virtual teams and business processes are often cross-organizational. Furthermore, we discuss *design requirements* for such an integrated system and provide one example *systems implementation* (Caramba), which we consider helpful in understanding how virtual teams use combined workflow and groupware systems on the Internet. Finally, we conclude this paper by discussing lessons learned and by providing an outlook on our future work.

2. Motivation for Internet-enabled Integrated Workflow and Groupware

Consider the following simplified virtual team example scenario: An IT team (systems integrator) consists of some 10 team members from two companies who are responsible for IT systems consulting and implementation for many customers, such as customer company XY. The virtual team consists of a project manager and one or more team members. The sales director (Mr. Zander) would like to stay informed on all work activities and status information at all times. The team members have to work on customers' premises many hours a week and the sales director is traveling frequently as well. This example case is shown in Figures 5 and 6. The presented scenario, although simplified, shares many characteristics with other virtual teams:

Team members

- (a) require *status information* on all work activities performed by other team members for a joint project
- (b) are increasingly *mobile*,
- (c) *jointly work on artifacts*,
- (d) require knowledge about the *multiple relationships* (associations or links) *between the artifacts* and the context in which they were created, shared, and distributed (i.e. who, what, when, in which context).
- (e) Team leaders require *on-demand access on project status and artifacts* as well as *critical communications* between team members and between team members and customers.

Context is composed of information on the "who, when, how, and why". In order to illustrate the lack of context, consider an "Explorer"-like view on a file system. This view allows the person to see documents (artifacts) stored inside folders. The names of such folders might reflect project names themselves. The mentioned view on these documents does not contain further contextual information on what a person (yourself, or others) actually have to do (did) with it (e.g. create another document, send an e-mail to customer, call partner organization, etc.).

For example, if the person in the above example needs to see who actually received a document stored in any given (project) folder, he is required to manually retrieve his e-mail box in order to find this information.

Groupware systems, in most cases, follow a shared workspace metaphor (see next section) utilizing shared folders (e.g. web-based folders) and do not provide information on the *relationships between artifacts* (e.g. a document) and the *associated activity of a business process* (e.g. activity "presentation for customer XY") or on the *status* of other team members' work activities. However this relationship information is of paramount importance for knowledge-intense business processes of virtual project teams in order to provide contextual information on knowledge artifacts for processes such as new product development, which cannot be modeled using a traditional workflow management system, due to the frequent ad-hoc nature and frequent exceptions that would occur.

The context of our proposed solution
To fully understand the context of collaborative technologies relevant for an Internet-enabled Workflow and Groupware integration, it is important to first analyze the dimensions of current systems. Our conceptual framework analyzes collaborative technologies along two orthogonal dimensions: Knowledge Usage and Knowledge Context as shown in Figure 1. Each axis has a continuum of characteristic features.

Figure 1. Conceptual Technology Framework

Knowledge Usage is about the "paradigm" in which knowledge is used. In its simplest form knowledge is only retrieved. The next stage allows (in addition to retrieval) the sharing of knowledge, for example by using a shared editor for synchronous joint editing. The following stage (includes the steps before) and enables users to create workspaces by organizing knowledge artifacts using files and folder hierarchies. The distribution stage enables knowledge workers (in addition to the previous features) to distribute knowledge artifacts (objects) by using push/pull mechanisms. Finally, the link stage allows retrieval, sharing, workspaces, distribution, and in addition allows links between all knowledge artifacts.

The second dimension *Knowledge Context* reveals contextual information on knowledge artifacts. Generally we can say that the higher contextual information is the more process awareness is stored together with artifacts. In its simplest stage it allows auditing of artifacts. For example users may view timestamp information on creation and routing of artifacts. The second stage enables organizational modeling, i.e. to define persons, roles, departments, and other organizational constructs required to design organizational structure. Organizational models allow organizations to define a set of access rights and rules for artifacts. The third stage additionally enables process modeling. Process tracking enables administrators to view the progress of business processes and the progress of activities as the building blocks of processes. Finally Reporting and Analysis supports analysis of all previously explained stages, and statistical comparisons between them. We find it useful to relate technologies on the market today to those two dimensions in order to elaborate our proposed approach.

Groupware systems usually provide very low knowledge context information but provide relatively high knowledge usage capabilities, since they enable users to retrieve, share, organize their work in workspaces, and to distribute artifacts. An example of a system also supporting linkage of knowledge artifacts will be presented in section 6. Document Management systems are increasingly integrated with WfMS as recent mergers demonstrate (e.g. Lotus Notes/OneStone). Project management (PM) software is still mostly viewed as software for individuals (i.e. project managers) and rarely offers collaborative or business process aware solutions. Moreover, in most cases PM software is not integrated with corporate information systems and in fact is only utilized as a graphical modeling tool for outlining tasks. Most Knowledge Management systems on the market today are workspace-centered and provide only very simple forms to model organizational structures (e.g. using roles only, but not skills). To the best of our knowledge, few KM systems provide interfaces to business process modeling and enactment systems (the domain of WfMS) (Dustdar, 2002b). Most KM- systems enable users to retrieve artifacts from repositories (workspace metaphor), but rarely allow distribution and process awareness. There has been a lot of work recently on the subject of knowledge management. Future work on the integration of workflow and groupware benefits from a process-oriented Knowledge Management approach, where Knowledge Management Processes (KMPs) define the interaction between knowledge works in a process-oriented manner and consist of activities that are supported by knowledge management key actions, such as searching, categorizing, and storing information (e.g. Woitsch and Karagiannis, 2002).

3. Related Work

There has been a lot of work on classification models for collaborative systems, however, no “one and agreed upon” taxonomy of analyzing and understanding collaborative systems has been proposed so far. Academia and industry suggest various classification schemes. In industry for example, people frequently use the term e-mail and groupware interchangeably. More generally, there is the tendency to classify categories of collaborative systems by naming a product (e.g. often many use the term Lotus Notes and Groupware interchangeably). Academic research has suggested many different classification models. For a recent extensive survey of collaborative application taxonomies see Bafoutsou and Mentzas (2002). DeSanctis and Gallupe (1987), Ellis, Gibbs and Rein (1991) and Johansen (1988) suggest a two dimensional matrix based on time and place, where they differentiate between systems’ usage at same place/same time (e.g. electronic meeting rooms), same place/different time (e.g. newsgroups), different place/different time (e.g. workflow, e-mail), different place/same time (audio/video conferencing, shared editors). This classification model helps to easily analyze many tools on the market today; however, it fails to provide detailed insights on collaborative work activities themselves as well as their relationship to business processes. Ellis (2000) provides a functionally oriented taxonomy of collaborative systems, which assists in understanding the integration issues of workflow and groupware systems and is shown in Table 1.

Table 1. Collaborative systems taxonomy

Taxonomy	Metaphor	Characteristics
Keepers	Shared workspace, Database	Access control, artifacts versioning, backup, recovery, and concurrency control.
Communicators	Messaging (point-to-point)	Supports explicit communications between participants.
Coordinators	Coordination and Organizational Model	Handles the ordering and synchronization of activities.
Team-Agents	Agent (Application or User-Interface agents)	Provide domain-specific functionalities, such as a meeting scheduler.

The classification system of Ellis (2000) provides a framework to understand the characteristics of collaborative systems and their technical implementations. The first category (Keepers) provides those functionalities related to storage and access to shared data (persistency). The metaphor used for systems based on this category is a “shared workspace”. A shared workspace is basically a central repository where all team members put (upload) shared artifacts (in most cases documents) and share those among the team members. Technical characteristics of “keepers” include database features, access control, versioning, and backup/recovery control. Popular systems examples include *BSCW* (Bentley et al., 1997), IBM/Lotus *TeamRoom* (IBM, 2002) and the Peer-to-Peer workspace system *Groove* (Groove, 2002). The second category (Communicators) groups all functionality related to explicit communications among team members. Basically this boils down to messaging systems (e-mail). Its fundamental nature is a point-to-point interaction model, where team members are identified only by their name (e-mail address) and not by other means (e.g. by skills, roles or other constructs, as in some advanced workflow systems). The third category (Coordinators) is related to ordering and synchronization of individual activities that make up a whole process. Examples of Coordinator systems include workflow management systems. Finally, the fourth category (Team-Agents,) refers to (semi-)intelligent software components that perform domain-specific functions and thereby help the group dynamics. An example for this category is a meeting scheduler agent. Most systems in this category are not off-the-shelf standard software. Both evaluation models presented above provide guidance to virtual teams on how to evaluate products based on the frameworks. Current systems for virtual teamwork have their strength in one or two categories of Ellis’ framework. Most systems on the market today provide features for *Keepers* and *Communicators* support or are solely *Coordinator* systems (e.g. Workflow Management Systems) or are *Team-Agents*. To the best of our knowledge there is no system integrating at least three of the above categories in one system. In the following section we evaluate current collaborative systems categories for their usage in virtual teams and summarize their shortcomings in respect to the requirement for virtual teamwork.

Evaluation of Collaborative Systems

Cooperative tasks in virtual teams are increasing, and as a consequence the use of collaborative systems is becoming more pervasive. In recent years it has increasingly become difficult to categorize systems according to the frameworks discussed above, since systems boundaries have become increasingly fuzzy and due to recent requirements for virtual teamwork. Traditional systems in the area of interest to virtual teamwork are groupware, project management (PM) and workflow management systems (WfMS). The mentioned system categories are based on different “metaphors”. Groupware systems mainly can be categorized along two lines (metaphors), namely the *communications* or *workspace* metaphor.

Communications-oriented groupware supports unstructured work activities using communications as the underlying interaction pattern. One very popular instance of communications-oriented groupware is e-mail. When e-mail is used as the main medium for virtual teams (as in most cases), data and associated information (such as attachments) remain on central mail servers and/or personal inboxes without any *context* information in which those email communications were used (involved business processes, performed activities, created artifacts as described above). Enterprise groupware systems are generally focused on enterprise-wide messaging and discussion databases and do not support organizational components and structures such as people and their associated roles, groups, task, skills, etc. This leads to "organizationally unaware" systems treating all messages alike (semantically) and without any awareness of underlying business processes, which are essential for efficient collaboration in project teams.

Workspace-oriented groupware, on the other hand, allows team members to upload/download artifacts using files and folders to organize their work. Groupware, as indicated above, usually does not implement an underlying organizational model (i.e. providing information on the structure of a team such as team members and their roles, skills, tasks, and responsibilities). The lack of explicit organizational “structuring” is a disadvantage as well as an advantage at the same time. It is disadvantageous because traditional groupware has no “hooks” for integrating business process information, which is important in order to integrate artifacts, resources, and processes. This will be discussed in more depth in the next section. The advantage of the lack of explicit organizational structure information is the fact that such systems may be used in all organizational settings without much prior configuration efforts on the one hand and secondly this leads to increased personal flexibility, as the proliferation of e-mail systems in teamwork demonstrate.

The second category, which we will briefly investigate in this section are *project management systems* (PM). As we have stated above virtual teamwork is in most cases organized as project work. Projects have well defined goals and are defined by their *begin* and *end* dates as well as by the required resources and their tasks (work breakdown structure). It is interesting to note however, that PM systems traditionally support the work of the project manger as the main (and only) user of the PM system. They do not support dynamic interaction (instantiation) of processes. More recently, project management systems combine with information sharing tools (shared workspaces) to provide a persistent storage for artifacts. The enactment of the task by team members, as defined by the project manager, is not supported by PM systems. In other words we can conclude that PM systems are not geared towards virtual teamwork but focused more on the planning aspect. They provide “static” snapshots (usually in the form of GANNT charts) of projects and how they “should” be. There is no support for the work activities performed by the virtual team members.

The purpose of *workflow management systems* is to support the notion of processes within and in some cases between organizations (Aalst and Kumar, 2001; Bolcer, 2000; Bussler, 1999). A business process can be unstructured (ad-hoc), semi-structured, or highly structured (modeled). For example a business process such as “customer order entry” can be modeled using a traditional WfMS. However *highly structured* processes can only be enacted (instantiated) as they were designed. If an exception occurs, a workflow administrator needs to re-model the process before the execution can continue. This limits the usability of WfMS in a world where constant adaptation to new situations is necessary and where teams are increasingly mobile and distributed. An example of an *ad-hoc* process is discussion of a project’s design review. A *semi-structured* process consists of groups of activities, which are modeled; however in contrast to a structured (modeled) process it may also consist of activities, which are not pre-defined. A process is semi-structured, when there might be one or more activities between already modeled activities such as *assign process*, which are not known beforehand and therefore cannot be modeled in advance. Most WfMS distinguish models of a business process (build time) and their enactment (run time). For a comprehensive study of Workflow products and their characteristics see Aalst and Hee (2002). For a detailed discussion of WfMS-shortcomings, e.g. unclear semantics of Workflow control patterns see Aalst et al. (2003). The often required modeling of workflows before enactment, as well as their unclear modeling semantics, frequently lead to substantial inflexibility (Ellis, 1995) for virtual teams. In (cross-organizational) virtual teams "exceptions are the rule", therefore modeling a process (project) is often not possible for creative, innovative virtual teams of knowledge workers such as in product development or consulting teams, as our small motivating scenario presented above, indicates.

To our knowledge, traditional groupware and workflow management systems do not support the requirements outlined above (see e.g. Bafoutsou and Mentza, 2002). Most groupware systems follow a “workspace” metaphor, which allows users to upload/download artifacts using files and folder to organize their work. When e-mail is used as the main medium for project teams (as in most cases), data and associated information (such as attachments) remain on central mail servers and/or personal inboxes without any *context* information in which those email communications were used (involved business processes, performed activities, created artifacts as described above). Enterprise groupware systems are generally focused on enterprise-wide messaging and discussion databases and do not support organizational components and structures such as people and their associated roles, groups, task, skills etc. This leads to "organizationally unaware" systems treating all messages alike (semantically) and without any awareness of underlying business processes, which are essential for efficient collaboration in project teams. To summarize: The requirements for virtual teams cannot simply be met by loosely coupling traditional workflow and groupware systems, which are based on different metaphors and goals. To summarize, we suggest that an integrated Workflow and Groupware system needs to

(a) provide organizational constructs (persons, roles, skills, groups, tasks) in order to flexibly model an organizational structure and responsibilities of virtual teams;

(b) provide constructs for modeling generic tasks and associated document-templates or applications in order to enact them for particular team members;
(c) provide the means to graphically model a control flow for business processes at a high level of abstraction (granularity);
(d) allow cross-organizational process enactment with ad-hoc processes (without modelling) and analysis of interaction patterns between team members;
(e) allow integration (and communications-references) of database repositories as an important resource for artifact management

Our goal for the next section is to distil the above requirements of (cross-organizational) virtual teams into a systems-architecture and present our design goals and to outline architectural considerations of *Caramba* (Hausleitner and Dustdar, 1999; Caramba Labs 2003)

4. Design goals and architecture of Caramba

The presented system was initially a research prototype with development beginning in 1997. It evolved into a commercial product, which was launched in 2001. Caramba manages all involved processes in knowledge work for project teams: from creating ideas, via invoking enterprise applications to support this work, up to coordinating and making these processes visible and reusable both within, and between organizations. The main design goal encompasses the integration of workflow and groupware features outlined in previous sections. Therefore, support for meta-modeling the organizational (team) structure as well as the ability to integrate "business objects" (e.g. DBMS-tables such as product databases) is essential. Secondly, traceable and continuous support regarding the relationships between people, artifacts, and business processes is of paramount importance. Thirdly, different levels of integration with other communications systems (e.g. SMTP-server) and information systems (e.g. Web Server) should be possible. Finally, the system should allow outside project partners to be integrated with the virtual team as tightly as possible, allowing access to all information provided by a CarambaSpace, if security policies allow. Various access mechanisms such as using a web-browser, Java client application, or mobile device have to be provided. The software (middleware and client) is written in Java based on Java SDK 1.3 for enhanced GUIs using the Java Foundation Classes (JFC) to support drag and drop. Software architectures typically include the description of *components*, *connectors*, and *configurations*. For this it is important to decompose a system into a well-defined set of components that have clear responsibilities. Since architectures for project teams have to integrate with various information systems installed in organizations, we decided to strive for a middleware style rather than a classical client-server style. Figure 2 provides a high-level overview of supported setup scenarios. The left part shows a simple application scenario with one project team (and therefore one CarambaSpace) using an embedded Java-DBMS. Caramba services are hosted on one server. Clients may access the project's team-space using Java client applications or via a built in HTML/XML portal. In cases where tight integration with corporate information systems and databases is required, the administrator utilizes the Caramba meta-modeler and relation wizard to integrate corporate DBMS and other resources such as SMTP-Mail servers or company Web Servers (see right hand side of Figure 2).

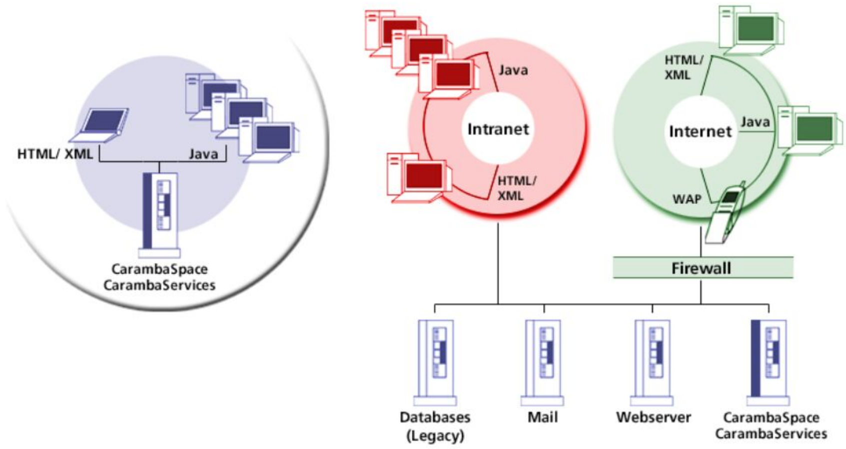

Figure 2. Integration scenarios

The following descriptions will point out the respective architectural style used in a particular layer or component. The Caramba software architecture (Dustdar, 2002a) is composed of multiple layers: middleware, client suite, and a persistence store, as depicted in Figure 3.

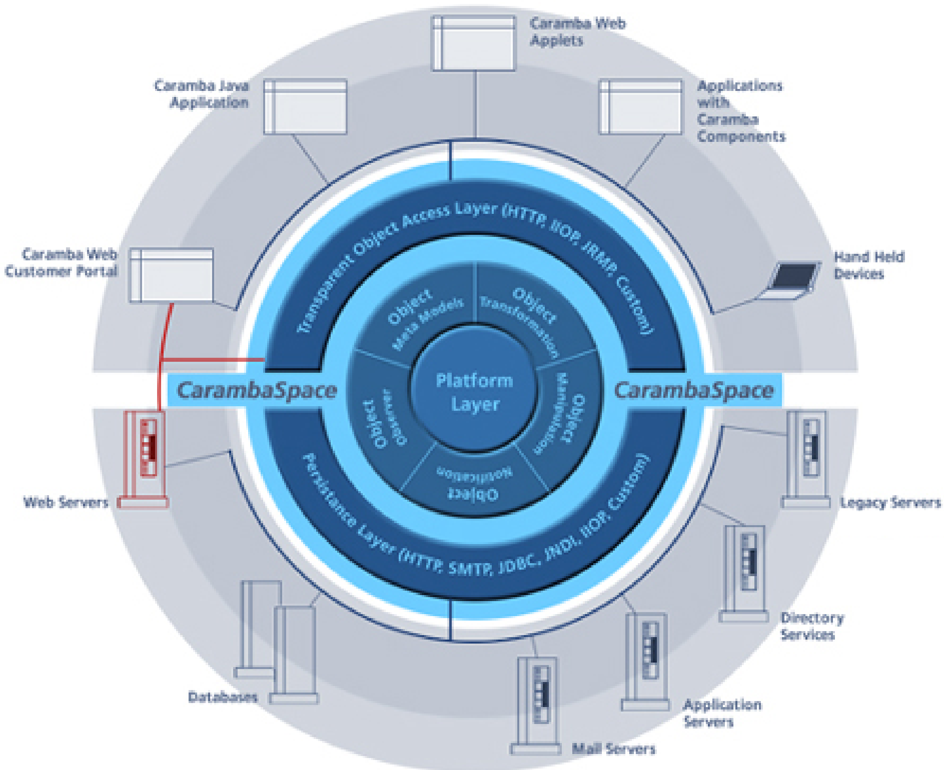

Figure 3. Conceptual Systems Architecture

Objects (artifacts) and services (e.g. notifications) are accessed through the ***Transparent Access Layer*** (TAL) from the CarambaSpace platform (middleware). Depending on access mechanisms and the requested services (e.g. via Java client with RMI protocol or via Web browser with http), Caramba provides a unique way to handle requests using a metamodel framework to describe content and separating presentation, logic, and data. This model permits high flexibility, enables customization, and extensions as well as the adoption of new devices or technologies.

The goal of this layer is to offer transparent access to a CarambaSpace. The CarambaSpace is defined by an address (host) on the Internet and the port address where there Caramba system is running. The TAL utilizes various services to transform, describe, manipulate, and observe objects. All objects managed through a CarambaSpace are well described using a meta-model description framework. Objects can be customized in their structure (e.g. adding columns to tables, adding relations to objects) and their presentation by adopting their meta-model description. Any changes are dynamically reflected by client components. Based on the meta-model description framework, Caramba enables various options to customize data and content as well as to integrate data from different resources (e.g. corporate databases). This layer also provides facilities for fine-grained object notification services and the implementation of customized services based on object observers. The middleware however, does not manage states and persistence of objects itself. Objects are stored, manipulated, and retrieved via the **Persistence Layer** (PEL). Caramba leverages and adopts standard Java based technologies (e.g. JDBC, JNDI, HTTP, etc.) to access and integrate data. This architecture provides the required flexibility virtual project teams where team members mostly connect using a web browser while still require access to all relevant context information provided by Caramba. Examples of such context information include: *who* performed *which* activities (including sub-activities and time/cost information); *which* resources were utilized for these activities and *how* the activities are related to the overall business process. Figure 4 provides an overview of the client components, their connection to the CarambaSpace as well as the connection between a CarambaSpace and the persistent data store (e.g. embedded Database or relational DBMS).

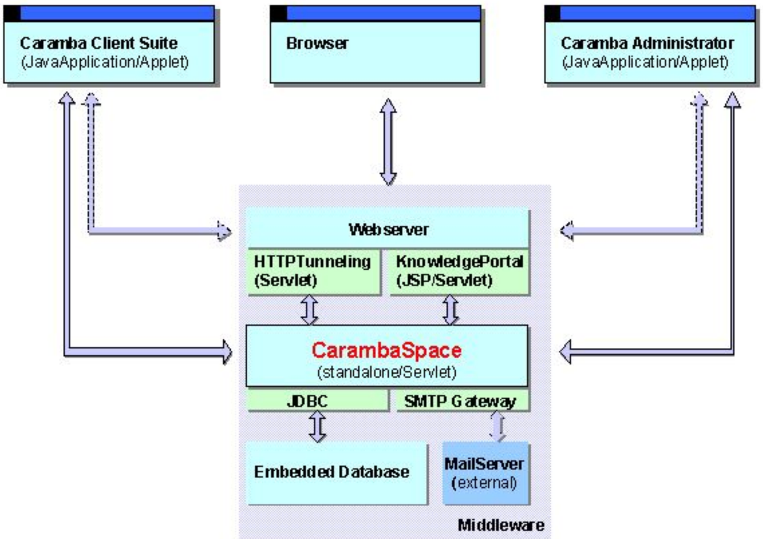

Figure 4. Caramba Configuration Overview

Caramba supports various configurations of virtual teams and is “organizationally aware”. It supports “organizational constructs” such as: Persons, Roles, Groups, Skills, Units, and Organizations. The utilization of organizational objects enables various team configurations, where members (Persons) may be part of Groups, Units, and Organizations, and possess various Skills. Using those objects a team leader can model a virtual team structure consisting of team members, their configuration (what roles and skills each members has; to which groups each member belongs; which document-templates and applications each member is allowed to invoke).

Components

Caramba’s administrator components allow customization and extension of Caramba. The administrator toolset, as summarized in Table 2, comprises a set of components, which allow modification of the default meta-model (e.g. organizational model of project teams) and integration of company-wide information systems such as SMTP or Web server.

Table 2. Administration components

Server Components	**Feature**
Meta-Model Administrator	Manage meta-models (customization)
JDBC Wizard	Integrate data from various datatabases
Relation Wizard	Describe relations between objects
Service Administrator	Administrate Caramba services
Mail Integration	Specify email integration and forwarding
Web Server	Web Server (can be exchanged for custom web servers e.g. Apache, Netscape, IIS, etc.)

Caramba offers a suite of software components for end-users to support collaboration, coordination, and cooperation for distributed project teams. Table 3 provides an overview of the client components.

Table 3. Client components

Client Components	**Feature**
ActivityCenter	Manage work items and coordinate, cooperate, communicate with others
ObjectCenter	Access, browse, and link objects
ProcessModeler (for the Administrator)	Model and view business processes
ActivityAnalyzer (for the Team leader)	Analyze and track work and project progress
Notification Center	Register and manage object notifications
Knowledge Portal	Access real time project data through the web

The *ActivityCenter* is the main “collaboration hub” for project team members. Here, project team members, work on their work items, route them to colleagues, and track the work item history if required. The interaction model used is “communications oriented”. In Figure 5 one can see the concepts of *Sender, Addressee and Recipient*, which allow a Sender (Person) to send a work item to any organizational construct (i.e. Addressee, such as Group, Role, Skill, Task, Organization). The construct Addressee is automatically mapped to one or many recipient(s) by the system. This means that Caramba users actively route work items to other team members. Additionally, team members integrate artifacts into the system and link them with their activities. The Activity Center allows those Persons associated with the begin activity of a process model to instantiate (enact) a workflow model. Caramba allows deviations from the process model without the need to re-model the process model, simply by routing the activities to the desired organizational object. The system provides detailed views on who sent (coordinated) activities to whom and when. This provides traceability for all project team members regardless of their location and the system they use to access a CarambaSpace (e.g. Java Client suite, Web browser, Mobile Phone). The ActivityCenter allows *continuous traceability* of business processes to team members, as depicted in Figure 5. Each team member may see all work activities (i.e. the interaction pattern and communications between team members), their status (e.g. New, Done, Read, Active), and the dates when respective activities began (Started), or ended (Stopped) as well as the deadlines activities.

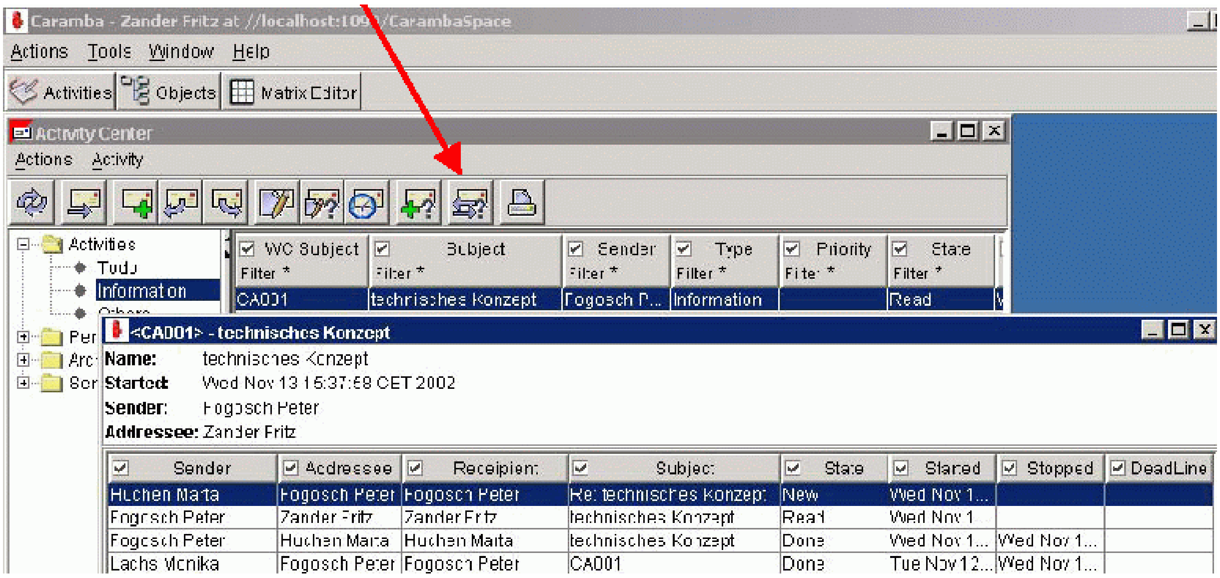

Figure 5. ActivityCenter

The second important end-user component, *ObjectCenter* provides a mechanism to *link* (process) activities with artifacts, as discussed previously. Based on the meta-model discussed above, Caramba provides a set of *organizational objects*: Persons, Roles, Groups, Skills, Units, Organization, Tasks, and Documents (i.e. Templates). Utilizing these *organizational constructs* an administrator is able to model any organizational structure, such as hierarchical, flat, or matrix. Each object class consists of attributes describing the object. The *object class Persons* contains attributes about the Person such as name, address etc. The object class *Roles* allows definition of organizational roles such as “Head of IT”. The object class *Group* defines project settings such as “Product Team IT-Solutions”. *Skills* enable definition of required skill sets such as “Certified Java Developer”. *Units* describe permanent departments such as “Marketing”. The ObjectCenter provides means (by drag & drop) to link the rows of object classes with each other. The ObjectCenter enables (a) project team members to view relationships between *who* (organizational constructs) is performing *which* activities (Tasks) and using *what* (artifacts, documents). In order to fulfill the design goal supporting the relationship between people, artifacts, and processes, Caramba supports modeling of business processes and their enactment by implementing a workflow engine (Process Modeler), utilizing the information presented in the section above (ObjectCenter), using tasks and their associated organizational constructs in directed graphs. Figure 6 shows an integrated “projects” database, which allows team members to “attach” projects to their interactions and to provide status information on the chosen “Business Object”.

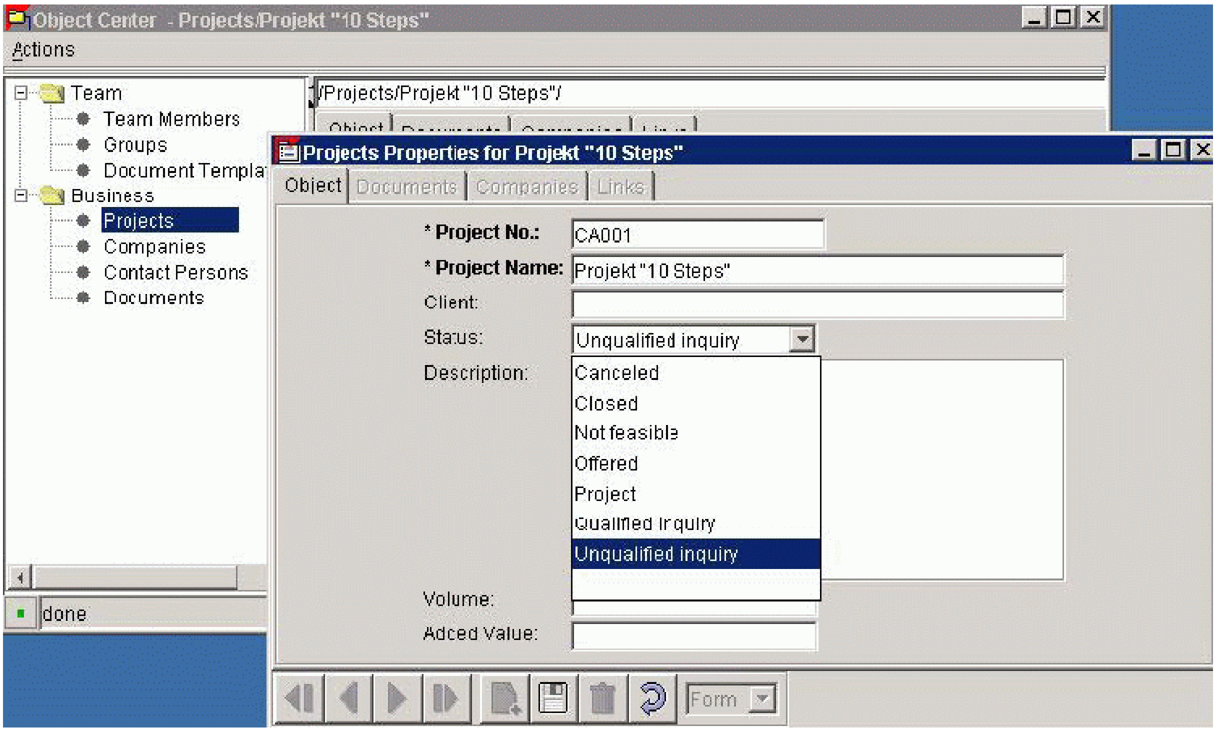

Figure 6. ObjectCenter

5. Lessons learned and Conclusions

Integration of Workflow and Groupware functionalities is essential for virtual teamwork. However, it would be naïve to believe that such an integration is achievable as a pure "software integration" endeavour (i.e. by gluing systems together). Both system categories have their relevance in many business scenarios. Virtual team collaboration on the Internet, a fairly recent organizational reality, requires possibilities from both "ends" of the Workflow and Groupware spectrums. Motivated by increasingly active virtual teams, this paper outlined some foundations of process-aware collaborative work. We provided an analysis of current Workflow and Groupware shortcomings in respect to virtual teamwork, outlined the design goals, architecture and an implementation of a system aiming at supporting virtual teams on the Internet. The system presented is in fact a hybrid between Workflow, Groupware, Project - and Knowledge Management systems. Our approach presented in this paper enables virtual teams to *link artifacts* such as documents (format independent) or database entries to process *activities* performed by human actors.

Based on the experiences we gained from the customers of our presented system, we wish to summarize some shortcomings and lessons learned. Firstly, the *change of working habits* is one of the most difficult barriers for new (collaborative) systems. Most people use e-mail rather extensively, even for actually managing virtual teamwork and interacting on the Internet. Although most people agree that tracking work activities (as depicted in Figure 5) is an essential shortcoming of normal e-mail, it is still difficult for them to change the style of their everyday work. Secondly, the *trade off between modeled business processes and ad-hoc processes* is difficult to handle for virtual teams. Virtual teamwork is, in most cases, rather unstructured (ad-hoc) or semi-structured. However, team leaders as well as members could benefit more, if they would analyse their interaction patterns and distil those experiences into process models for future process modeling and enactment. The third current limitation is strongly connected to *mobility and replication* issues of virtual teams. Our current philosophy and implementation follows an "always on" metaphor. This means that virtual team members connect (via PCs, PDAs, and SmartPhones) to the Caramba Middleware (CarambaSpace) in order to collaborate with fellow members. However, as mobility increases, a "nomadic" working style penetrates virtual team work. Nomadic work requires the option to work "offline" and to replicate the work results later. This is currently not implemented in our system.

We believe that supporting the integration of artifacts, resources, and processes is of paramount importance for distributed virtual teams. Our future work includes peer-to-peer modeling and execution of process models, integration of distributed persistent data stores, and providing Caramba-functionalities as web services with the goal to enable loosely coupled collaborative services.

References

Aalst, W.M.P. van der, Hofstede, A.H.M, Kiepuszieweski, B., Barros, A.P. (2003). "Workflow Patterns". Distributed and Parallel Databases, 14(1), pp. 5-51, July 2003.

Aalst, W.M.P. van der, Hee, K. van der (2002). Workflow Management: Models, Methods, and Systems, MIT Press, Cambridge, MA.

Aalst, W.M.P. van der, Kumar, A. (2001). "A reference model for team-enabled workflow management systems", *Data & Knowledge Engineering*, Elsevier, 38 (2001), pp. 335-363.

Bafoutsou, G., Mentzsa, G. (2002). "Review and functional classification of collaborative systems." International Journal of Information Management 22 (2002), pp. 281-305, Elsevier Science.

Bentley, R., Appelt, W., Busbach, U., Hinrichs, E., Kerr, D., Sikkel, K., Trevor, J., and Woetzel, G. (1997). "Basic support for cooperative work on the World Wide Web." International Journal of Human–Computer Studies, 46(6), pp. 827-846.

Bolcer, G.A. (2000). "Magi: An Architecture for mobile and disconnected Workflow", IEEE Internet Computing, 4(3), pp. 46 – 54.

Bussler, C. (1999). "Enterprise-wide Workflow Management", IEEE Concurrency, 7(3), pp. 32-43.

Caramba Labs Software AG (2003) http://www.CarambaLabs.com

Casati, F. et al. (2001). "Developing e-Services for composing e-services", (K.R. Dittrich, A. Geppert, M.C. Norrie, Eds.) Proceedings CaiSE 2001, Computer Science Lecture Notes, Springer Verlag, 2001, Toronto, pp. 171-186.

Chen, Q. et al., (2001). "Peer-to-Peer Collaborative Internet Business Servers", HP-Labs Technical Working Paper HPL-2001-14.

Craven, N., Mahling, D.E. (1995). "Goals and Processes: A Task Basis for Projects and Workflows", Proceedings COOCS International Conference, Milpitas, CA, USA, 1995.

Dayal, U. et al., "Business Process Coordination: State of the Art, Trends, and Open Issues", Proceedings of the 27th VLDB Confererence, Roma, Italy, 2001.

DeSanctis, G. and Gallupe, R.B. (1987). "A foundation study of group decision support systems." Management Science, 23(5), pp. 589-609.

Dustdar, S. (2003). "Towards integration of artifacts, resources, and processes for virtual teams", in Virtual Team: Projects, Protocols, and Process (David Pauleen, Ed.), Idea Group Publishing

Dustdar, S. (2002a). "Mobility of Context for project teams." Proceedings of the International Workshop on Mobile Teamwork at the 22nd International Conference on Distributed Computing Systems (ICDCS 2002), July, Vienna, pp. 391-395, IEEE Computer Society Press.

Dustdar (2002b). "Collaborative Knowledge Flow - improving process-awareness and traceability of work activities". 4th International Conference on Practical Aspects of Knowledge Management (PAKM 2002), December, Vienna, pp. 389-397, Springer LNCS.

Ellis, C.A. (Skip) et al. (1995). "Dynamic Change within Workflow systems". In Proceedings COOCS International Conference, Milpitas, CA, USA.

Ellis, C.A. (Skip) (2000). "An evaluation framework for collaborative systems". University of Colorado at Boulder Technical Report CU-CS-9001-00.

Ellis, C.A. (Skip) (1998). "A Framework and Mathematical Model for Collaboration Technology", In: Coordination Technology for Collaborative Applications – Organizations, Processes, and Agents (Conen and Neumann, Eds.), Springer Verlag, 1998, pp.121-144.

Ellis, C.A., Gibbs, S.J., and Rein, G.L. (1991). "Groupware: Some issues and experiences." Communications of the ACM, 34(1).

Groove (2002), http://www.groove.net

Hausleitner, A., Dustdar, S. (1999). "*Caramba - Ein Java basiertes Multimedia Koordinationssystem*", In: Erfahrungen mit Java. Projekte aus Industrie und Hochschule. Silvano Maffeis, et al. (Eds.), dPunkt-Verlag, Heidelberg 1999.

IBM Corporation, http://www.ibm.com

Johansen, R. (1988). Groupware. Computer-support for business teams. The Free Press. New York.

Nagypal, G. et al. (2001). "Integrating Workflow and Groupware Functionalities for Co-operating Small and Medium Sized Enterprises: a Case Study", Proceeding of Seventh International Workshop on Groupware, pp. 38 - 43. IEEE Computer Society Press, Darmstadt, Germany, September 2001.

Puustjärvi, J., Laine, H. (2001). "Supporting cooperative inter-organizational business transactions", Proceedings DEXA 2001, Computer Science Lecture Notes, Springer Verlag, 2001, pp. 836-845.

Schael, T., (1996) Workflow Management Systems for Process Organizations. New York: Springer 1996.

Workflow Management Coalition (WfMC) (2003), *Workflow Management Specification Glossary*, http://www.wfmc.org

Woitsch, R., Karagiannis, D. (2002). "Process-oriented Knowledge Management Systems Based on KM-Services: The Promote Approach. Proceedings of the International Conference on Practical Aspects of Knowledge Management (PAKM), LNAI 2569, pp. 398-412, 2002, Vienna, Springer-Verlag.