Distributed Knowledge Management in Virtual Organizations: the 'Social' Experience Factory

Demosthenes Akoumianakis
Faculty of Applied Technologie, Technological Education Institution of Crete, Greece,
da@epp.teicrete.gr

Abstract: This paper considers knowledge management functions as carried out by distributed virtual teams involved in the compilation of information-based products using dedicated and domain-specific computer-mediated practices and tools. We are concerned with two primary tasks, namely depositing shared assets and assembling information-based artefacts by appropriating the benefits of virtual networking. Moreover, these tasks are considered from the perspective of the Social Experience Factory (SEF) – a platform enabling rich collaborative interactions between geographically dispersed members of communities of practice. The SEF incorporates domain-specific workflows and several model-based tools to facilitate systematic accumulation and reuse of collaborative artefacts. An account of these is provided by discussing current implementation in the context of a pilot application.

Keywords: social experience factory, knowledge sharing, distributed collective practices

1. Introduction

Knowledge management is a term with a broad connotation used to describe any process or practice related to the acquisition, capture, construction, sharing, and use of knowledge, wherever it resides, to enhance learning and performance in organisations (Borghoff, 1998). Typically, knowledge management focuses on 'managing' what organisations know (Davenport and Prusak 2000) but also what they should know (Lueg, 2003). Recently, on-line communities of practice (Wenger, 1998, Wenger and Snyder, 2000) and organisational memory information systems (Hackbarth and Grover, 1999) have established powerful mechanisms for both spreading codified knowledge as well as accumulating new experiences. Nevertheless, managing what the organization needs to (but does not yet) know remains a challenge and turns out to be a difficult undertaking. Market research indicates that companies do invest on monitoring on-line discussions aiming to find out what is being said about a company and its products using tools such as eWatch, CyberAlert and IntelliSeek. The key question, however, is how organisations translate these findings into new knowledge and experience.

Virtual communities offer an alternative model for improving knowledge-based assets and competence building by fostering a social view on learning and knowledge creation (Brown and Duguid, 2000; Erickson and Kellogg, 2001). The underlying assumption is that knowledge is deeply embedded in the collaborative artefacts as well as the technological practices and social context of the community which creates and manages it. Moreover, there are several genres of software tools that support social construction of knowledge (Erickson and Kellogg, 2001) in communities of practice. Examples include tools for information sharing such as electronic mailing lists, or listservs, MOOs, tools for memory management (Ackerman, 1998; Ackerman and Palen, 1996), collaboratories (Olson and Olson, 2000), and tools for idea exploration (Erickson et al., 1999).

This paper aims to shed light to the 'social' nature and collaborative practices of knowledge management in the context of an electronic village of local interest. An electronic village of local interest denotes an advanced virtual community emphasizing and promoting local activities of community members in a designated domain of discourse (i.e., tourism, learning, construction, etc). The virtual nature of the community necessitates that community practices are encapsulated into computer-mediated tools and workflows to allow incremental and collaborative construction of artefacts, thus new knowledge. In this context, the normative perspective of the paper is to describe the baseline of the 'social' experience factory (SEF) – a platform enabling rich collaborative interactions between members of virtual groups / communities. Our interest is to investigate generic and domain-specific functions supported by the SEF in order to facilitate engagement, participation and negotiation on behalf of the community members. Equally important is the analysis of the type of interactions leading to knowledge construction and new codified experiences. To this effect, we will make reference to recent results of an on-going research and development project, namely eKoNEΣ (see acknowledgement), which has established and operates an experimental version of the SEF in an electronic village of local interest with a thematic focus on regional tourism.

Reference this paper as:
Akoumianakis, D. "Distributed Knowledge Management in Virtual Organizations: the 'Social' Experience Factory." *The Electronic Journal of Knowledge Management* Volume 6 Issue 1 2008 pp. 13 - 32, available online at www.ejkm.com

The paper is structured as follows. The next section outlines current trends in knowledge management and positions the present work against prevailing models, and in particular, the concept of communities of practice. Then, we present the basic archetype of a 'social' experience factory for carrying out collaborative activities in distributed collaborative settings. The emphasis is on two distinct components of the SEF, namely the lifecycle stages characterizing virtual coalitions / squads and the experience organisation intended to facilitate knowledge and experience management. The following section discusses operational aspects of the SEF in the context of a pilot application in regional tourism. For purposes of illustrating basic concepts the paper refers to a case study involving the construction of a vacation package. The analysis and discussion section reports on recent experiences and contrasts the SEF against alternative models. The paper is concluded with a summary and an account of on-going and future work.

2. Related work

The term knowledge-management has various connotations in the literature. It is frequently interpreted as technological infrastructure allowing information to flow through intranets and/or other types of technology (Morey et al., 2000). This approach to knowledge-management is labelled *information-centred* – where access to information is the key provision. Another body of research emphasises the social aspects of knowledge management, recognising the need for learning to take place. This is termed *learner-centred* knowledge management and seeks to engage the participants in a learning process. In the business world, the learner-centred approach is frequently related to competitiveness, as it is felt by many that the faster an organisation can learn, the more successful it will be; hence the drive for businesses to become *learning organisations* (Argyris and Schon, 1978; Hidding and Catterall, 1999; Senge, 1990). Each approach has developed a variety of models to provide prescriptive insight to the type of knowledge managed, the process of knowledge management and the resulting benefits. Detailed reviews of these models (see Despres and Chauvel, 2000; Jennex and Olfman, 2004) are beyond the scope of this article.

From the various theoretical models, developed over the years to facilitate and support knowledge management, the concept of *communities of practice* (Lave and Wegner, 1991) is the most relevant to the present work. Communities of practice are phenomena said to: "galvanise knowledge-sharing, knowledge and change". They are defined as: "groups of people bound together by shared experience and passion for joint enterprise" (Wenger and Wegner, 2000). This can be described as cross-functional teams – brought together to capture and spread ideas and know-how. In terms of "classical" theory, communities of practice focus on articulating tacit knowledge (Takeuchi and Nonaka, 1995) adopting the learner-centric approach to knowledge management. The available literature on communities of practice, although reach on theoretical thinking, exhibits at least two shortcomings. The first is that the vast majority of the reported studies concentrate on community management – i.e., discovering, building and maintaining community – dismissing or under servicing the elements of practice. Moreover, very few from the existing pool of studies claim and/or provide convincing evidence that the systems built/studied provided a 'place' to actually engage in the practice that the community is about. This is further supported by recent empirical evidence on the use of collaborative technologies (i.e., discussion forums, shared databases, repositories and workflow) by organizations (Merono-Cerdan et al., 2008). A second shortcoming is that existing studies on communities of practice analyze community management in single organizations, either public or private (Juriado and Gustafsson, 2007). The more demanding problem of community formation across organizational boundaries – either through inter-organisational partnerships or external communities of practices – is seldom addressed (Dewhurst and Cegarra Navarro, 2004).

The above justify the need for investigating alternative operational knowledge management models in community settings which concentrate on managing 'collective' practices in social inter-organisational partnerships. This is precisely the rationale of the present work. Specifically, our aim is to contribute to the available literature by reviewing, describing and presenting components of a technological frame of reference and supporting tools which allow virtual partnerships to manage diverse resources codified as shared / deposited knowledge and compile / assemble new assets through resource sharing, cooperation and collaboration. The distinct characteristic of the present work is that it builds upon recent literature on communities of practice (Wenger and Snyder, 2000) to formulate the argument that community management – the primary focus in recent writings – is not sufficient to attain distributed 'collective' practices. Indeed, there is a compelling need to design and build *technologies for practice* to allow virtual teams (or communities of practice) to attain true cooperation and collaboration. To this end, we describe the architectural underpinnings of the 'social experience factory' and how it is applied in an engineering domain, namely regional tourism, for building information-based products in community settings.

3. The 'social' experience factory

One of the fundamental premises of knowledge-based product development is to understand and improve quality and productivity (Tiwana, 2000). In doing so, development teams can benefit from empirical evidence and previous project experience. Even for small organizations, large amounts of information can be built up over the years comprising expertise, project data, lessons learned, quality models, etc. For such information to be usable, it needs to be modelled, structured, generalized, and stored in a reusable form in order to allow the effective retrieval of relevant artefacts (Cubranic et al., 2004). A continuous build-up of knowledge requires a suitable organizational structure and appropriate tools. Basili introduced the notion of the experience factory (Basili, 1993) as an institutional concept comprising three distinct components, namely the *software development organization*, the *experience organization* and a *support organization* separate from the other two components. The task of the support organization is to carefully package, document and certify (where applicable) software artefacts. In the original formulation of the experience factory, Basili did not prescribe a particular role for technology or the type of tools needed to support the operation of an experience factory. However, in subsequent publications several examples of codified and packaged experiences have been described as well as the ingredients of the underlying technological set-up (Basili et al., 2001; Seaman et al., 2003).

3.1 Objectives of the SEF

In our recent work, we are experimenting with a model for knowledge and experience management, which is motivated by the experience factory, although it fosters an alternative perspective with regards to both the building components (constituents) and the activities being undertaken. We refer to this model as the 'social' experience factory and it aims to address a number of specific objectives, summarised as follows:

- The SEF seeks to provide the basic model for appropriating the benefits of virtual networking in information-based industries in which products are non-material (intangible) and knowledge is central to gaining competitive advantage.
- The SEF should operate as a 'virtual' software factory (Aaen et al., 1997) tuned to managing and reusing shared assets, tools and components. This requires an orientation towards implementing assembly lines rather than traditional production lines.
- The SEF is proposed as a domain-independent archetype of a virtual organisation with an explicit focus on collaborative practising; in this context domain-specific elements and practices are realised by dedicated tools such as domain-specific design languages, models building components, visual manifestation of artefacts and sound (XML-based) protocols.

To realise these objectives the SEF is organised in distinct and separate constituents, as shown in Figure 1. The rationale for this separation of functions is to be found in the type, range and nature of tasks allocated to each constituent. As shown in Figure 1, the SEF distinguishes between two constituents, namely an activity-specific work environment referred to as 'squad organisation' and a separate knowledge construction and experience compilation organization, referred to as the 'experience organization'. The squad organization encapsulates the distinct lifecycle stages followed by collaborating teams as they attain joint goals. On the other hand, the experience organisation encapsulates two sub-components the knowledge construction environment and the experience codification. The important issue to be highlighted is that in contrast to the squad organisation, which is flexible and independent of organisational model, the experience organization assumes a centralized institutional setting with designated roles and functions. Specifically, there is a moderating role responsible for (a) organizing, leading, mentoring and facilitating the group's virtual activities (b) extracting information from, updating and mining the shared experience data store and (c) codifying successful practices and experience by generalizing, adapting, recording, publishing and sharing artefacts. There is also a domain-specific component in the knowledge construction environment, which designates the distinct workflow stages (i.e., initiation, elaboration, deployment and tailoring) characterising the fidelity of the artefacts produced.

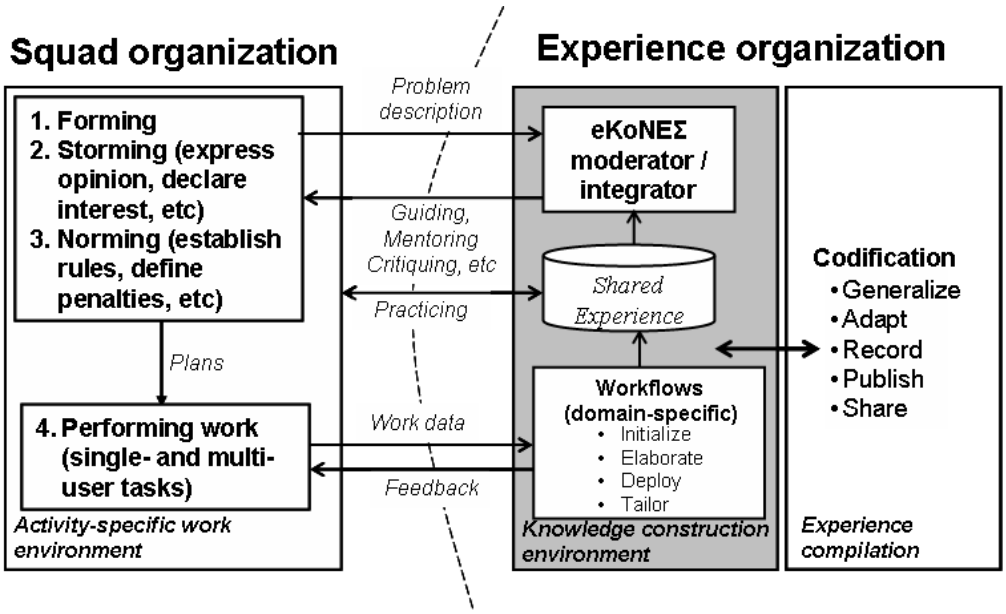


Figure 1: eKoNEΣ as 'social' experience factory (adapted from Basili 1993)

The SEF is functionally organised in such a way so as to support the social interactions taking place between collaborating group members. In this context, social interactions imply exchanges taking place between group members and being dependent on the group's lifecycle stage and level of stability. Such exchanges differ as the group progresses from *formation*, to *storming* (i.e., getting to know each other), *norming* (i.e., resolving conflicts and reaching agreement) and *performing* towards the common goal (Tuckman, 1965). The second reason for the 'social' qualification (of the SEF) is that the above distinct stages in the group's lifecycle are explicitly supported (by dedicated tools) and characterize the design of the SEF. In other words, the SEF assumes that group work entails attainment of distinct goals during the forming, storming, norming and performing stages. Throughout these stages, an experience function / organization compiles experiences by monitoring, analyzing and consolidating persistent outcomes of a group's collaborative exchanges. In the following we provide a detailed account of each component of the SEF as currently supported in the eKoNEΣ pilot in the area of tourism.

3.2 Constituents of the SEF

3.2.1 The squad organization & lifecycle

Squads are cross-neighbourhood coalitions (virtual teams) tasked to attain common goals by aggregating and negotiating primitive resources (i.e., neighbourhood assets). Neighbourhoods are communities of practice with topical/thematic interest. For instance, neighbourhoods in the tourism sector include transport, accommodation, cultural heritage communities, etc. Each neighbourhood sets up own rules of engagement which determine participation and acceptable social behaviour within the neighbourhood. As these neighbourhoods exist virtually, rules are embedded into processes covering *registration and access rights*, *acceptance of new members*, *setting rules for acceptable behaviour*, *security*, *privacy*, *freedom of speech/act* and *moderation*.

Squads are formed to carry out a designated mission, thus they are mission-specific. The mission may vary depending on the domain of application (i.e., tourism, learning or construction). Once formed, squads follow distinct stages to reach their ultimate target (see Figure 1). Initial formation is determined by the mission's requirements (or primitive services required) and the assets of neighbourhood members as declared during electronic registration to neighbourhoods. Each squad comprises one moderator and several participants joining forces to address a problem (i.e., develop a vacation package). The moderator designates the type of input required and establishes a pace of working. In due time, a squad may change in form and structure depending on contextual and circumstantial factors (i.e., a member may be temporarily unavailable or unwilling to commit further resources). This means that at any time, a member can opt out from a squad only through an explicit request for withdrawal.

However, dynamic formation does not ensure stabilization and effective performance. Instead, empirical evidence suggests that group stabilization is strongly correlated with the group's ability to effectively move

from the initial forming and storming stages into norming and performing. In other words, the group's level of stabilization increases as the group progressively moves from forming (i.e., trying out activities, expression of opinions), to storming (i.e., resolving conflicts) and into norming (i.e., enfolding group coherence, setting group objectives) and performing (i.e., carrying out activities towards the group's mission). The SEF provides explicit tools for moderators to manage squads as they move from formation to performance. These tools are transparent to squad members, while they make use of data posted / exchanged through the SEF's shared collaborative message board.

The forming stage

Typically missions relate to developing information-based products with specific characteristics. For instance, in the in the context of eKoNEΣ, missions constitute efforts for creating new vacation packages. The need for a new product (i.e., a vacation package) may be motivated either by a customer request or other circumstantial factors, such as a scheduled event, which may act as trigger for a new package. The mission is always specified by a moderator. Squad formation follows on the grounds of matching mission-specific requirements against members' deposited resources. Mission-specific requirements depict demand for certain neighbourhood services (i.e., in eKoNEΣ such neighbourhood activities may include transport, accommodation, cultural heritage, etc.). All registered partners offering such services are prospective members of the squad, but their ultimate participation in the squad is subject to their explicit commitment (or withdrawal). Commitment or withdrawal from a squad is an asynchronous notification task which involves a member's response to the moderator's invitation. This is indicated in Figure 2 which summarizes the asynchronous tasks performed by the involved actors in the forming stage. Specifically, the customer's request is typically manifested as a post to the eKoNEΣ forum through the portal. This is translated by the moderator to a mission by declaring a tentative package name, description, duration and indication of neighbourhood activities required. The milestone at this stage is an explicit mission as instance of a designated family of missions codified in the experience base.

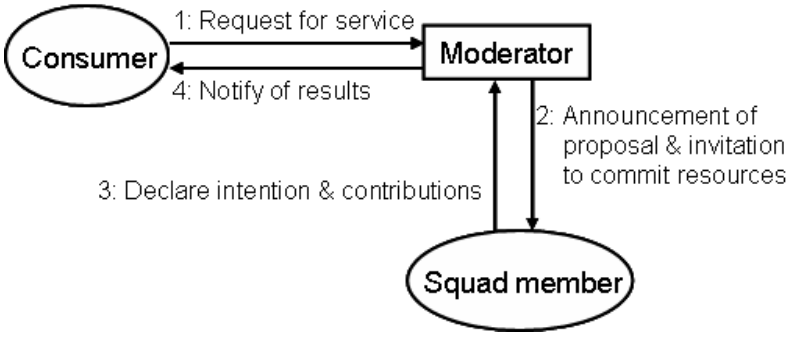


Figure 2: Asynchronous interactions during squad formation

The storming stage

Following initial squad formation, members of the squad engage in a variety of exchanges in the storming phase. These exchanges allow members to gain detailed insight into the squad's mission. This is achieved by setting objectives, exchanging opinion, posing issues for consideration and by advancing proposals. All these are persistent exchanges visible to the moderator. A typical interaction cycle in this stage is depicted in Figure 3. The moderator creates issues for discussion, which are manifested as threads in the squad's message board. These issues are extracted from the template of the designated mission family, but they may also be created in due course. Squad members are invited to contribute by adding issues for discussion, offering alternatives and raising concerns. These are all persistent posts to the message board, packaged as XML statements and having a semantic indicator which designates the type of contribution (i.e., add issue, proposal for new issue).

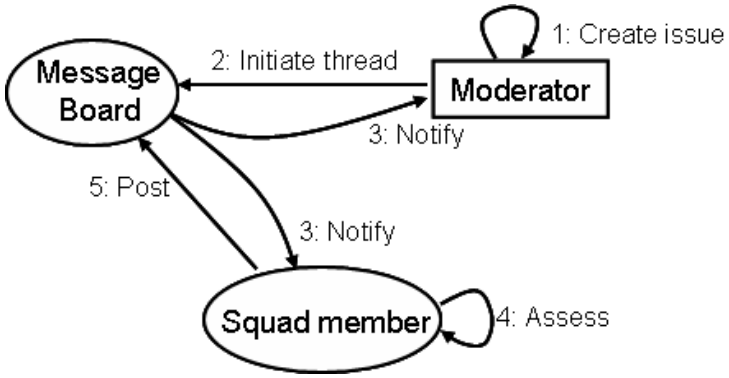


Figure 3: Asynchronous exchanges in the storming stage

The norming stage

The norming stage is where the squad members establish a common agenda for reaching the ultimate goal. The agenda is defined in relation to the issues identified and the alternatives offered in the storming session. An agenda is considered complete when there is no pending issue. Consequently, the emphasis in the norming stage is not on what is to be done but rather how it is to be done. The squad's moderator acts primarily as a facilitator rather than a mentor. The exchanges involve choices from a set of proposals or alternatives populated during the storming phase, while all issues raised during storming must be resolved. Once again these exchanges take the form of asynchronous XML posts to the message board, similar to those encountered in the storming stage, but this time their scope covers specific options for designated issues. The moderator can review the state of affairs at any time by considering the issues which have been resolved, those pending as well as the behaviour of squad members as expressed by their votes. Figure 4 summarizes an interaction cycle in the norming stage. Once again, the moderator initiates exchanges by modifying the status of issues and highlighting the alternatives. For each issue squad members are invited to argue for or against an alternative through voting. Issue resolution is by majority vote.

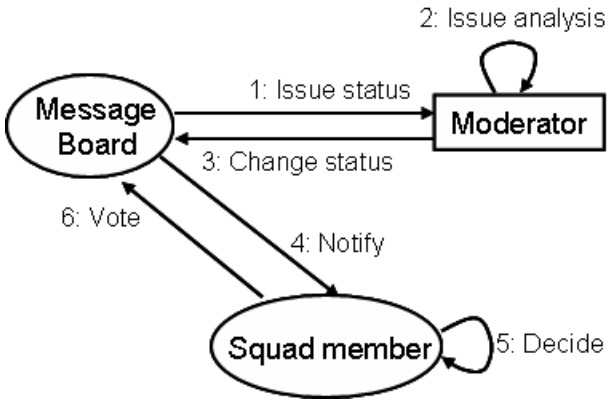


Figure 4: Asynchronous interactions in the norming stage

The performing stage

The performing stage is where each squad member undertakes the 'local' tasks required to facilitate smooth completion of the overall mission. We refer to these tasks as 'local' to highlight the fact that they embody or reflect upon local practices, not necessarily common to other members of the group. As such the nature of these tasks is highly individualistic (i.e., they are performed by members atomically and in a manner which is transparent to the rest of the squad), while the tasks' scope and execution is bound to the organizational boundaries of each squad member alone. The details of each member's work, the tools used to perform this work and the artefacts produced are indifferent to the rest of the squad. In summary, the only binding condition for squad members in the performing stage is to obey to the norms and rules defined jointly by all squad members in the norming stage. As for the social responsibility of each member, this amounts to feeding through to the squad an indication of the task's accomplishment.

3.2.2 The experience organization

The experience management organization of the SEF is broadly defined in terms of three sub-constituents namely a distinct role (i.e., eKoNEΣ moderator), a collection of domain-specific workflows and the persistent experience data store. As indicated in Figure 1 these constitute components of the knowledge construction

environment which mediates and interacts both with the squad operational settings (i.e., activity-specific work environment) and the experience compilation component.

SEF moderators

In the SEF, the moderator or administrator is a key role that is mandatory for the effective operation of a squad. This role involves active engagement in a range of social interactions and knowledge-based tasks. Social interaction entails monitoring, guiding, facilitating, mentoring and critiquing squads as they move from formation to performing. On the other hand, the knowledge-based tasks involve manipulation of the 'soft' components of the experience organization (i.e., visual models, templates, evidence, etc). Accordingly, the moderator's work may be seen as a complex undertaking with a dual responsibility. The first responsibility is acting as a competence centre or an experience broker mediating between the virtual assets of an eKoNEΣ electronic village and the active squads. In this capacity the administrator offers advice on problem solving strategy, tools, and best practices, based on existing experiences. The second responsibility of the moderator is acting as a silent critic to mine the data generated by a squad as it works to accomplish its set targets and to codify these data in the form of persistent new knowledge. These responsibilities are further detailed in the next section where operational details of the SEF are described.

Domain-specific workflows

In the SEF sharing, negotiation and construction of knowledge is not an ad-hoc process. It combines information flows exchanged / produced in the course of executing a small set of domain-specific workflows. These workflows are initiation, elaboration, deployment and tailoring (see Figure 1). Before describing each of those in detail it is important to highlight two issues. The first is that these workflows provide insight to a mission from an artefact-oriented perspective. In other words, if a mission is to create a vacation package, then the workflows depict the stages the vacation package will go through from inception to execution. The second issue relates to the temporal overlap between the workflows and the squad lifecycle stages. This is illustrated in Figure 5. As shown the forming stage continues throughout the workflows to allow flexibility. Thus, a member may withdraw from a squad at any time, while new members may join a squad at a later stage if the need arises. In both cases, withdrawal and / or commitment require explicit notification of the moderator and the rest of the squad. The storming and norming stages continue throughout the elaboration and deployment workflows. Finally performing overlaps with deployment and tailoring.

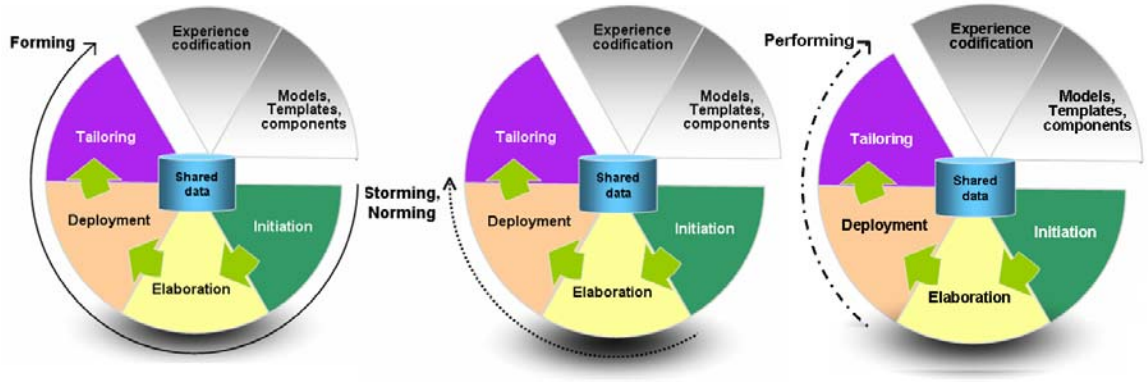


Figure 5: Overlap between workflows and squad lifecycle stages

The package initiation workflow is the responsibility of the moderator. This involves definition of an abstract package by assignment of a name, indication of resources required (i.e., neighbourhood activities) and package duration (start and end date). In effect, this task amounts to creating a new instance under the abstract package family. This instance will incrementally be transformed to a concrete offering. Once the instance of the package family is defined a corresponding squad is initially formed as a coalition of all members offering the resources required by the package.

Package elaboration requires a stable squad which is measured by the commitments posted to the shared message board. During elaboration, squad members seek to populate the designated package with all possible or alternative offerings. Their contributions cover specific parameters of the package, such as pricing of services, accommodating customers' preferences and declaring commitment to offer services. These exchanges take the form of 'request-post' replies and result in updates in the package's model or the introduction of pending issues requiring agreement. At the end of the elaboration phase, a new package has been populated and is available for review. In case of conflicts between the squad members or unresolved issues, the moderator launches a virtual meeting in the form of a synchronous session. This is an innovative component of the current version of the software as it supports typical groupware functions (i.e., object sharing, floor control) as well as role-based access to and various collaborative practices over the shared objects. Notably, throughout such exchanges the object of collaboration (i.e., a graphical version of the

package) remains fully synchronized, using a powerful object replication model. A typical synchronous scenario will be described when describing operational details of the SEF in the next section.

In the deployment stage the package has been agreed and becomes an active resource available to interested parties and prospective customers through the portal. This entails selection and authoring of one or more template layouts so as to facilitate package multi-platform presentation (e.g., desktop using Java or HTML, PDA or a cellular phone). In case an existing template layout does not suffice, then a new one can be created and stored as a reusable component in the experience data store. A dedicated LifeRay portlet has been developed to provide the container for deployed packages and to allow package navigation in a portlet context. Moreover, through the asynchronous notification mechanism built on top of LifeRay, all end users who have registered their interest in the package are informed and prompted to consider making a personalized reservation.

Package personalization / tailoring is the stage where end users (i.e., prospective customers) are exposed to the package and adapt the package so as to reflect own preferences. Package adaptation entails making choices from the variety of alternatives encapsulated in the deployed package. For instance, a user may select a particular type of accommodation, transportation or food and beverage from the range supported by the package. In reality this involves choice of specific squad member offering the service. Since the package is fully populated, end users can access it through a variety of devices including desktop computers, mobile devices or other network attachable terminals using the suitable templates. It is also worth mentioning that during tailoring users can engage in a variety of social interactions commonly found in on-line communities. For instance, prospective buyers of a package are presented with the feedback provided by persons who have already bought a similar package in the past. Also when tailoring a package, customers are presented with information on patterns of tailoring which have emerged. Finally, customers are also encouraged to provide ratings and write reviews for services offered and packages obtained.

The experience data store

At the core of the SEF's experience management organization is a domain-specific ontology, which serves as the main knowledge and experience-modelling repository. In the context of our current work, we are using Protégé (http://protege.stanford.edu/) to build the ontology for the eKoNEΣ-Tourism electronic village. The design philosophy of the ontology is as follows. eKoNEΣ members are registered in neighbourhoods such as residence, transportation, entertainment, cultural heritage, etc. Each neighbourhood maintains its own social policies and rules of engagement. Each category is specialized into sub-classes representing structure of a neighbourhood with representatives instances and member offerings. Shared resources deposited by members are of two types namely `primitiveServices` and `packages`. A `primitiveService` is a neighbourhood specific activity (i.e., accommodation). `Packages` are built by assembling instances of `primitiveServices` and are negotiated by squads. They represent resources, which do not pre-exist but rather are compiled by members to facilitate an articulated demand. However, the process of assembling them and negotiating their details is distinct and totally different than conventional practices. Specifically, an instance of `Package` is derived from the archetype of a package family, in a similar fashion as a product inherits properties of a product line. Thus assembling a package involves incremental tailoring of properties of a family of packages. Each package is owned by the squad contributing to the package. Moreover, all deliberations made by squad members leading to the package are persistent and can be traced.

4. The SEF in operation: ekoneσ-Tourism

The SEF, as presented above, has been used to support computer-based collaboration in a pilot electronic village with a thematic focus on regional tourism (eKoNEΣ-Tourism) as well as in other small-scale case studies seeking to assemble 'collective' information-based products. The software platform developed integrates various components to facilitate management of squad lifecycles and domain-specific package workflows. To illustrate the concept, this section presents details of the SEF's operation for vacation package assembly by reusing shared resources and codified experience in collaborative settings. The specifics of a package are not really important for our discussion, as the SEF's models can be tailored to support several different families of packages. Moreover, we will not describe the portal, its augmented functionality i.e., custom portlets, electronic partner registration system, etc., or the interoperation of the portal and the tools described below, as these are technical details beyond the scope of the present paper. Nevertheless, the reader may find such details in recent publications (Akoumianakis et al., 2008) or visit the current version of eKoNEΣ-Tourism (http://www.e-kones.teiher.gr/web/Village/Home) to obtain insight to non-protected content and functions. Instead, our interest is to highlight the steps involved in assembling packages and the knowledge management tools supporting generic functions of the SEF.

Table 1 summarises a tentative scenario (in the form of partitioned narrative) describing activities taking place across three distinct constituents, namely the customers' context, the community context and the practitioners' context. We have intentionally oversimplified some of these activities to depict a logical sequence of steps without necessarily striving for the maximum of analytical insight. Indeed some of the activities such as 'raise/respond to issues', 'update model/parameters', etc., are demanding in terms of technological set-up, justifying the need for synchronous groupware, persistent exchanges, mining social interactions, etc., but this is further developed later on. In the following sections, we will briefly describe key technological components facilitating each context and its tasks.

Table 1: Partitioned narrative

The customers' context	The community context	The practitioners' context
1: Customer request for service	2: Create package	
	3: Announce new package	
	4: Invite participation	5: Confirm / reject invitation
		6: Contribute to package
	7: Update model	
	8: Raise issue	9: Respond to issues raised
	10: Request offer / bit	
		11: Update parameters
		12: Request clarification
	13: Clarification of issues	
	14: Consolidate issues	
	15: Publish package	
16: Tailor package / request changes		

4.1 Families of packages

The package family is the basic abstraction acting as a factory for concrete packages (i.e., instances of the family). The SEF maintains in a persistent data store a reusable description of an abstract package family in the form of packaged experience. Selection of a package family by a moderator signifies the scope of the package to be developed. This scope is defined in terms of designated neighbourhoods, corresponding services, choice of template and template resources (i.e., images, textual descriptions, etc). Figure 6 describes a relevant extract of a package family class model and the corresponding XML segments. Both these constitute elements of pre-packaged experience codified in the SEF. For purposes of simplicity we have intentionally omitted details of the package family description which are not needed for the present discussion.

As shown, the package is considered as a hierarchical structure comprising activities taking place within a day. Such containment hierarchies can be extended to depict alternative application domains; an issue addressed in section 5. Activities represent instances of neighbourhood services and can be interrelated. The transition from a package family to a concrete offering (i.e., package instance) involves collaborative agreement on all elements of the model depicted in Figure 6. Such transition is achieved as the package being assembled proceeds from initiation and elaboration to deployment and tailoring. In practice, this is an incremental process taking place in the squad's virtual work room. The squad virtual workroom is implemented as a distributed Java application downloadable from the portal upon successful electronic registration to neighbourhoods. The application is designed so as to provide a uniform interactive embodiment of a virtual work room and the corresponding collective practices involved in package assembly. Moreover, it separates all practice-related aspects required for package assembly from other communication-oriented tasks which take place through portal components (i.e., the community forums, directories, etc).

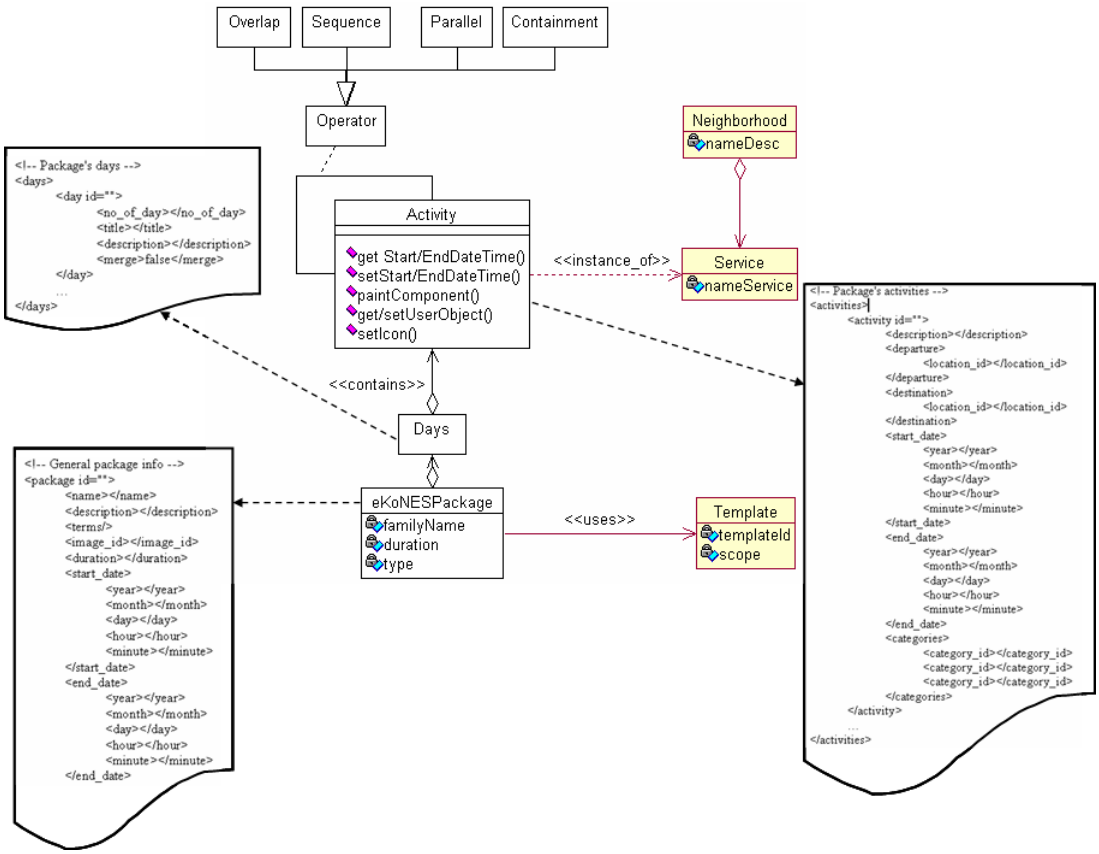


Figure 6: Class model of a package family

4.2 Distributed collective practices for populating a package family

Collective practices related to a package are executed using tools of the squad virtual work room which offers a synchronization point for all members of a squad and a shared virtual space for collaboration. The tools offered and the contents of a room are adapted depending on the role of the entrant. This type of adaptation is recognized and initiated by the system and covers both adaptation to local computing environment (i.e., language) as well as adaptation of the content of the room and its interactive manifestation. Thus, squad members have access to active packages to which they have committed resources, the room's shared message board and the synchronous collaborative session management tool. On the other hand, moderators have access to additional tools allowing them to carry out some of the practices involved in the package workflows.

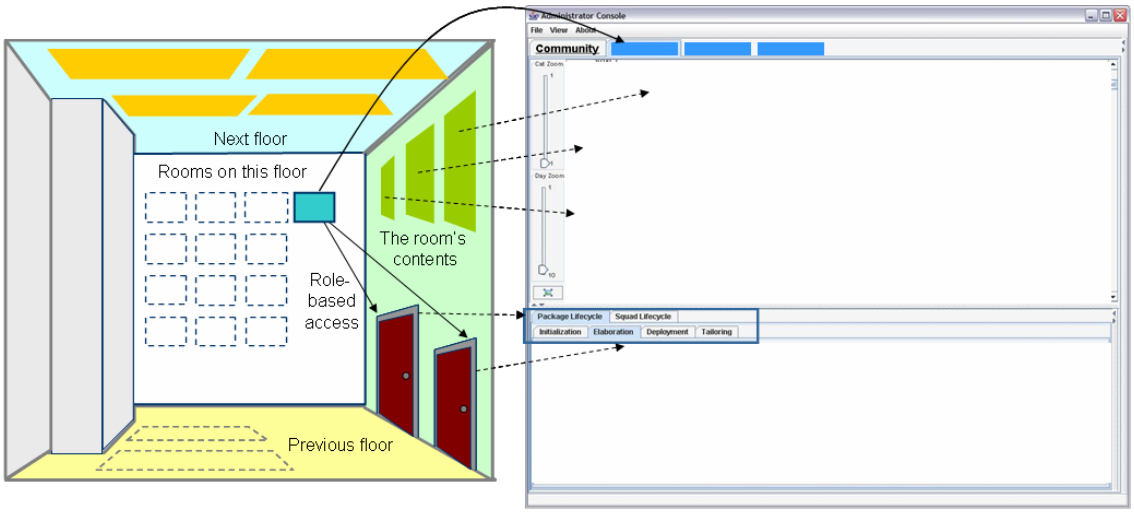


Figure 7: Elements of the squad virtual work room

Figure 7 describes mappings between components of the real world room metaphor to its symbolic embodiment in the graphical user interface. As shown, a package has its own work room represented by a separate tab. Squad members may be concurrently involved in several packages. A room has two distinct entry points representing the moderator's view and the squad members' view of the room. The moderator's view encapsulates dedicated tools for managing package workflows and squad lifecycle stages. The squad members' view is simpler and includes a visual representation of the package in a designated workflow and the tasks the squad member has to perform.

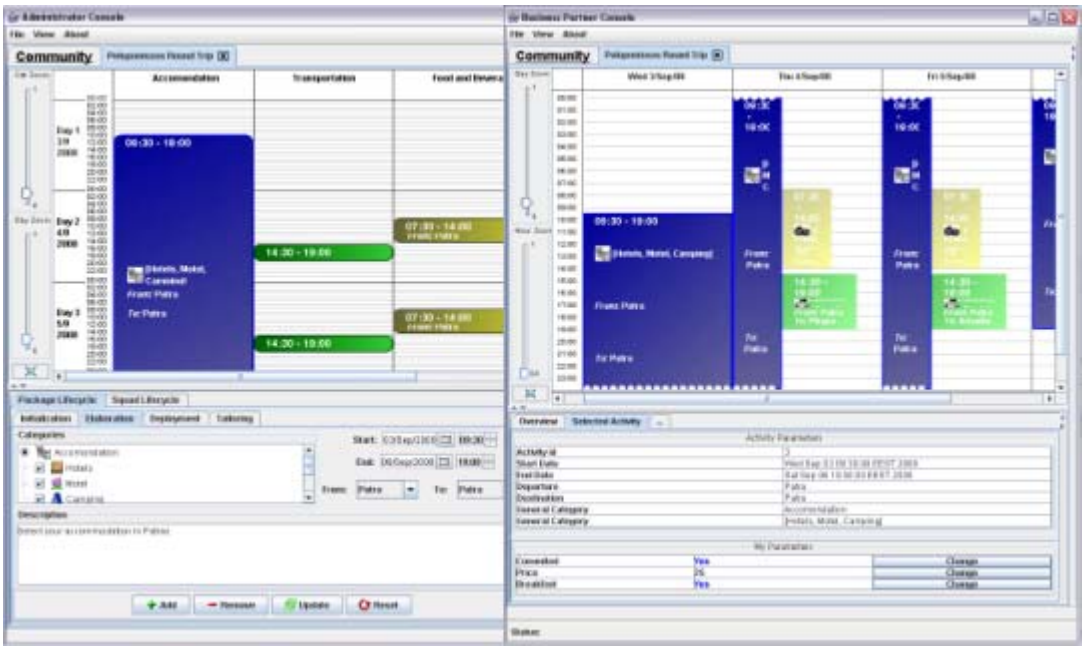


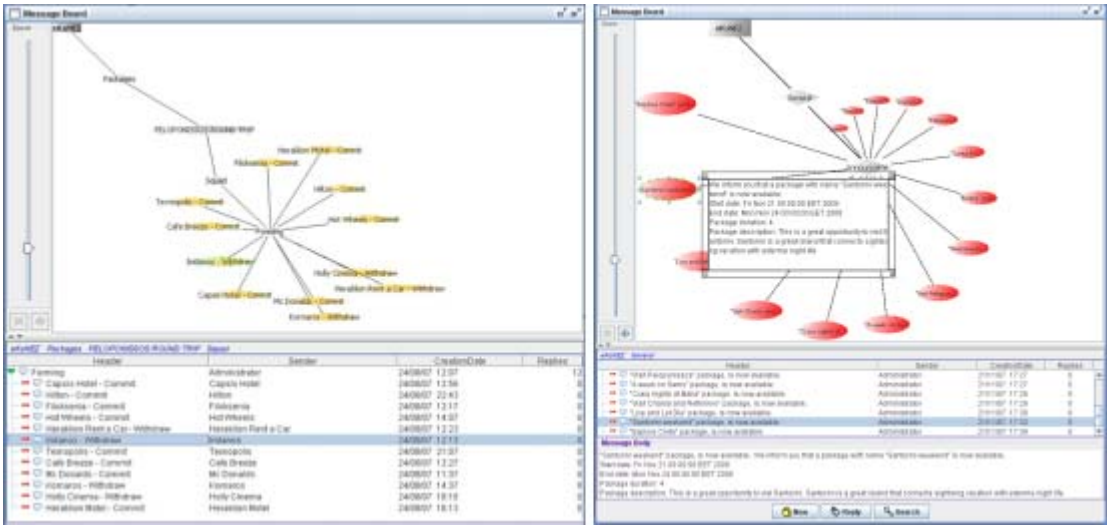
Figure 8: Package elaboration stage

Figure 8 presents respective instances of the moderator's and the squad members' user interfaces showing a package in the elaboration workflow. Figure 8a presents the moderator's tools for constructing an initial proposal for the package. Activities are defined using the dialogue at the lower part of the screen and asserted into the activity panel, which occupies the package workroom (upper part of the screen). For each day of the package, the moderator assigns the activities to take place. Columns of the activity panel represent the neighbourhoods contributing to the package, while rows of the activity panel list all activities scheduled for a particular day. All activities in a designated day are represented as selectable objects differentiated by colour depending on their type. The colouring scheme is also stored as packaged experience (i.e., XML document) and can be easily modified.

In the squad members' view (see Figure 8b) the package layout is different. In this view the same package is presented in a TV-program like metaphor with each column representing activities of a single day. As shown, activities are allowed to have full or partial temporal overlap. In this view the only selectable objects are those representing resources owned by the current squad member. Moreover, upon selecting an activity various semantic actions are available allowing squad members to express opinion, request clarification, accept or decline proposals, etc. These actions are realized as asynchronous posts of XML messages to the collaborative message board.

Figure 9 presents two different instances of the message board. Figure 9a depicts message exchanges by members of a specific squad tasked to develop a designated package namely "Peloponissos Round Trip'. In this case, posts to the message board are organized by squad lifecycle stage (i.e., forming). The selected stage (root node) appears at the bottom of a waterfall of nodes (i.e., eKoNEΣ → Packages → Specific package → Squad → stage) depicting pathway, while posts within this stage are organized in a circular fashion around the selected root node. The user can select either a child node and review the post or a node in the pathway and accordingly update the visual layout of the message board. In this manner the user can obtain quick access to large amount of data regarding a specific squad. An alternative layout is presented in Figure 9b which presents asynchronous communication at the level of the electronic village. The specific example presents announcements made through the eKoNEΣ forum with time indication. This time the circular layout of child nodes is replaced by a spiral layout in which distant posts are located farther away from the root. Both layouts have been built as extensions of the 2D visualization library JGraph

(www.jgraph.com). In both cases the visual layout is fully synchronized with the tree-table view at the bottom, allowing alternate access to a specific node.



(a) Package-specific instance (b) Village-wide instance

Figure 9: The collaborative message board

Conflicts are usually handled in synchronous collaborative sessions (virtual meetings) were all squad members can take part and express opinion. To this end a Java-based collaboration toolkit has been developed supporting role-based view of collaborative artefacts, object replication, and floor control for manipulating shared objects. A synchronous session is announced and launched by the squad moderator, who also defines the collaboration agenda (i.e., what part of the package is to be shared and replicated during the session). This is achieved by designating the components (or neighbourhoods) of a package and the package stage (i.e., elaboration). The package components indicate the parts of the model to be extracted and replicated during a synchronous session, while the choice of package stage defines the view in which the replicated object is to be presented. As each package stage emphasizes different aspects of a package, the corresponding views follow different interaction metaphors.

During synchronous collaborative sessions, a floor manager administers contributions to the shared model. The role of the floor manager is runtime permission assignments and participant notification of changes in the state of the model. Runtime permission assignment entails assessment of who has permission to act in the collaborative workspace at any time (see in Figure 10). Thus modifying or adding new content in a synchronous collaborative context follows some rules that clarify and assure that there is a logical coherence in the actions of the participants. Every time a new collaborative session is issued, a new instance of a session floor manager class is assigned to apply the desired policy. Once floor access is granted to a participant, all replicas of the shared model at the registered clients are locked. Manipulation of the shared object by the floor owner is transparent. In other words all participants are concurrently notified of the changes introduced to the shared object by the floor owner. This allows a kind of feed-through whereby actions in the shared object are always performed on the latest version of the model. Figure 11 summarizes this feed-through mechanism.

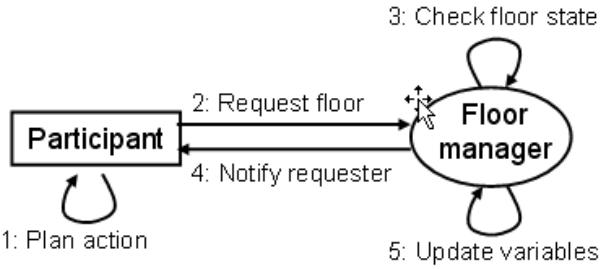


Figure 10: Floor control & management

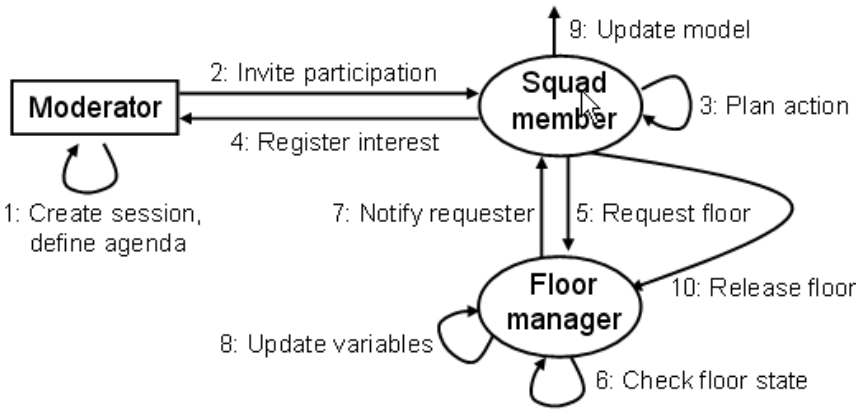


Figure 11: Feed-through

An illustration of a synchronous interactive session is depicted in Figure 12. As shown, the package's data are fully separated from its view. This allows the current version of the package to be assembled and presented differently to the moderator (Figure 12a) and the squad members (Figure 12b). The dotted lines represent alternative manifestation of the corresponding activities in the two distinct layouts. It is also worth mentioning the role-adapted view of activities. Specifically, activity objects in the moderator's view carry an awareness indicator (green bullets at the top of each object) showing the registered collaborating partners. On the other hand, activity objects in the partner's view have a different visual layout and contain additional information (as nested interaction elements). It is also important to mention that the two views although different are fully synchronized within the same collaborative session so as to preserve consistency of the workspace across the different views.

Collaborative practices (i.e., manipulation of the replicated object) in a synchronous session are moderated by a floor manager (see top right hand side dialogue). The floor is granted to a partner following an explicit request. The floor control policy is first-in-first-out, which implies that partner requests for the floor are placed in a queue. Once access to the floor is granted to a partner all other registered replicas of the object are locked. Locking is a mechanism which changes the degree of transparency of the 'locked' replicas objects allowing visual access to the object but no interaction. At any time, the floor holder can interact with the parts of the replicated object (i.e., the visual manifestation of the package) he is authorized to access. For instance, a partner in the accommodation neighbourhood can only manipulate the visual objects representing this neighbourhood. This implies that partners enter the collaborative session with designated access authorities corresponding to their deposited contributions. Only the moderator has full access to the entire replicated object. Allowable modifications to a selected object are automatically propagated to all registered participants. The collaborative actions allowed include inquiries about the object's current status (i.e., total number of contributors), annotating a selected object (i.e., for commenting, questioning or expressing opinion), updating parameters of a selected object (i.e., start and end time by resizing a selected object in a direct manipulation fashion). In addition to the above the moderator can introduce and remove objects.

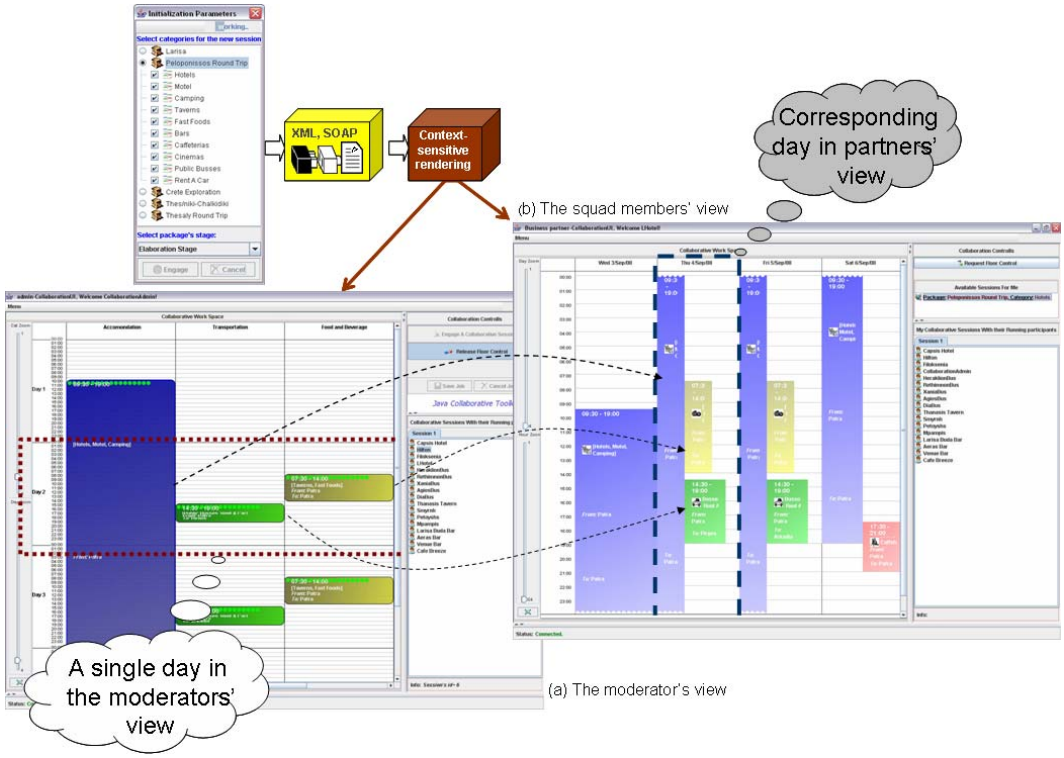


Figure 12: Package views in synchronous collaborative sessions

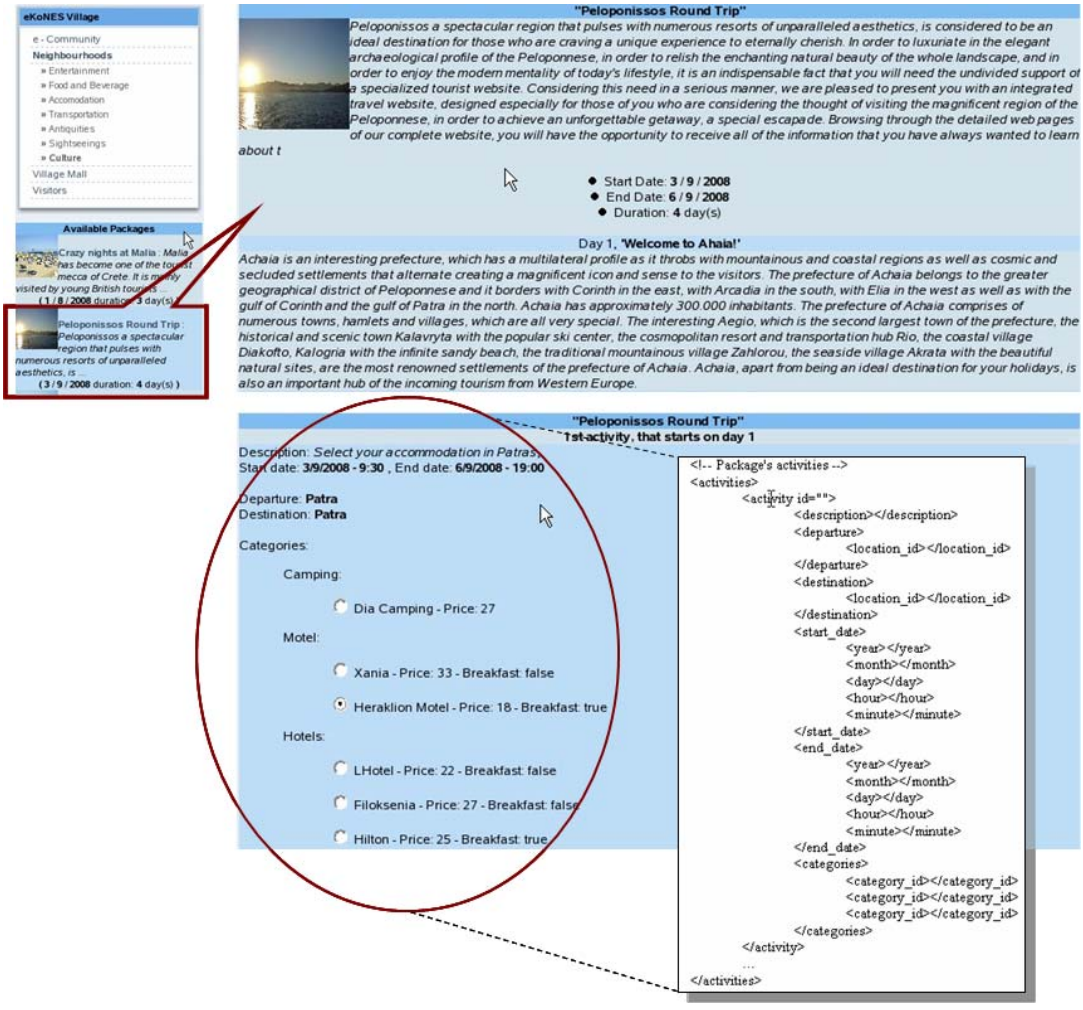


Figure 13: The package in deployment and tailoring workflows

Following resolution of conflicts, the package enters the deployment stage where it is transformed into a concrete offering with clear illustration of package options, alternatives and offers per activity. Package publication entails selection and authoring of a designated template layout and input of the required information), which updates the package's XML file. There may be more than one template layouts assigned to a package. Moreover, in case an existing template layout does not suffice, a new one can be created, by developing a suitable XSLT. Once the details of the package are agreed and finalized, the administrator publishes the package as a new resource through the portal. This signals an automatic update of the corresponding portlet in the portal, which assembles the components of the package automatically and publishes it. Figure 13 presents graphically this scenario. As shown the HTML file generated includes clear indication of the tailoring that the user can undertake to reflect a customer's detailed requirements and preferences.

5. Analysis and Discussion

The SEF promotes a knowledge management model which is built around one generic and one domain-specific component. The generic component is the squad organization, comprising distinct stages in the lifecycle of social groups, while the experience organization is by intention domain-specific. Nevertheless, there are elements and components of the experience organization which may easily be reused, extended and applied to other domains. In this section our aim is to discuss three issues related to knowledge management using the SEF. The first relates to the type, range and scope of packaged experiences codified in the SEF. The second issue addressed articulation of the codified experience to construct new knowledge. Finally, the third issue relates to reusing components of the SEF's infrastructure to address domains other than tourism.

5.1 Packaged experience in the SEF

To gain insight to knowledge management using the SEF, it is important to briefly describe what is codified experience and how this is turned into new knowledge. In general, codified experience in the SEF takes several forms and in all cases it constitutes a persistent asset. Firstly, collective experience is codified as package families – a notion corresponding to the concept of product lines in the software factories literature (Greenfield & Short, 2004). A package family packages the commonalities and variants that characterize instances within the family. For example, a common feature in all packages is the notion of an activity, which models an abstract service component offered by a neighbourhood. Then, a package can be considered as an aggregation of activities taking place within a day or other aggregating concept. On the other hand, activities differ in terms of type, duration, execution mode (i.e., sequence / parallel execution), interdependencies (i.e., a transport activity assumes a transportation medium), etc. XML offers a powerful representational medium for both manipulating and populating elements of a package family, as well as assembling packages within the scope of the family.

Another type of packaged experience is to be found in the toolkit libraries used to visualize elements of a package family. In this context, separating content form presentation is important for a system seeking to provide support for various families of packages. The SEF as currently implemented for eKoNEΣ, supports a small set of advanced interaction platform administration mechanisms, namely toolkit augmentation, expansion and integration, which allow the construction of domain-oriented and metaphor-specific interactive vocabularies (Akoumianakis et al., 2008). Example applications of these strategies have already been presented in the previous section, while applications in domains other than tourism vacation packages are described later on in this section. It is important to notice that, as these strategies are programming-intensive, it is useful to introduce augmented, expanded or integrated objects as parameterized, reusable and extensible software components allowing for different package families alternative metaphoric representations, all generated by using a suitable variation of these strategies.

The above types of packaged experience collectively constitute the SEF's domain–oriented design language which defines the intentional properties of artefacts within its scope as well as how they are manifested and assembled. In other words, the SEF's design language acts as a mediating mechanism facilitating the mapping of functions in a source domain to symbols in a target domain, and vice versa. In its current version this language is characterised by (a) distinct conceptual or ontological domain (see Figure 6); (b) visual manifestation of elements within this ontological domain (see Figure 8, Figure 12) and (c) computational manifestation defining the framework for rendering the language's statements in a designated presentation vocabulary such as a portlet (see Figure 13).

5.2 Knowledge management using the SEF

The next relevant question is how packaged experience is capitalised upon, refined, extended and translated to new knowledge. To address this question it is useful to briefly examine the types of new knowledge being facilitated. First of all, new packages constitute one type of compiled and consolidated knowledge, which is embedded into artefacts. As already mentioned this type of knowledge is recorded as XML and rendered as needed (i.e., assembled in a portlet, translated to HTML, etc). Another type of knowledge is informal and interpreted by assessing customers' attitudes towards the packages as well as the squads' exchanges in the course of creating the packages. Assessing customers' purchasing behaviour with respect to available packages (i.e., profiles of customers, type of tailoring requests, etc) reveals patterns in the target consumer base, which in turn, may be valuable in determining type of packages needed / appreciated by different customer groups (i.e., elderly, young people, business travellers) as well as corresponding marketing strategies. On the other hand, assessing the squad's exchanges, as they progress through their designated lifecycle stages, reveals patterns of behaviour such as partner clustering and cliques, members constantly disagreeing or withdrawing from squads, etc.

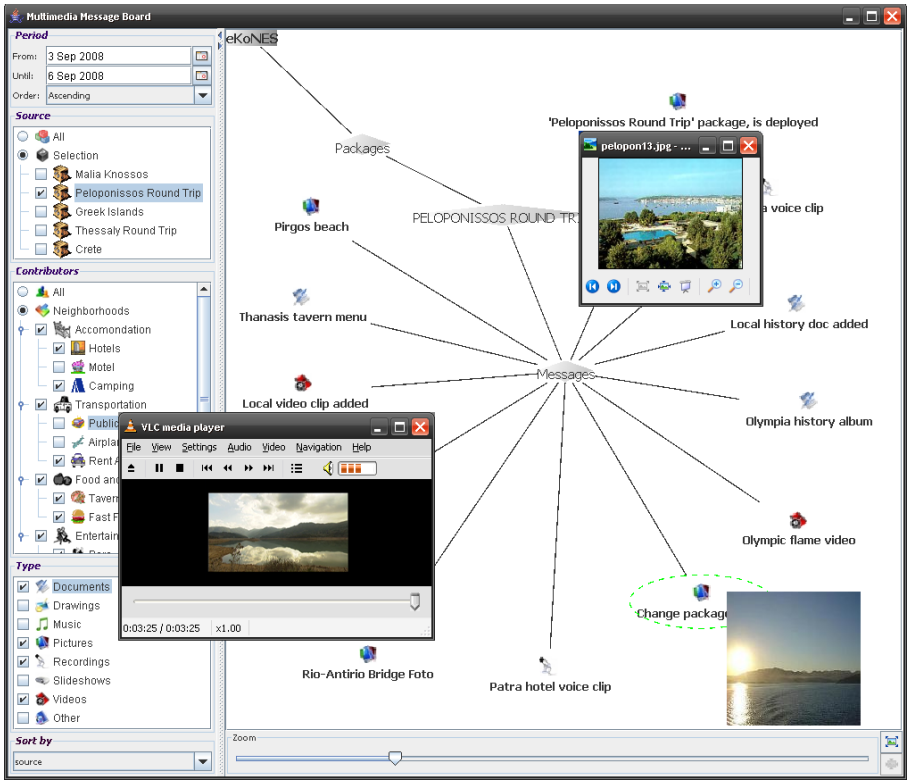


Figure 14: The CommonsBoard graphical user interface

The SEF's current implementation makes provisions for extracting such information by querying and exploring a type of 'virtual' memory. Figure 14 provides an illustrative example of exploration-based access to a squad's archive. This tool allows knowledge management at various levels. Specifically, the query interface (left-hand side component) allows users to specify queries by manipulating graphical components rather than using a conventional query language. In turn, the query results are mapped to a customized interactive display (right hand side of the display), which can be used to reach specific data or refined the original query. In this manner, it is possible to extract the rationale behind a vacation package (and by implication the contributions of the members of the associated squad) by progressively selecting the package and viewing all the messages posted or contributions of a particular type (i.e., documents, videos, images, etc). Another example of interpreted knowledge is the ability to assess behaviour of representatives of a particular neighbourhood in a designated package or all packages within a specific period.

5.3 Reuse and scalability in the SEF

In developing the concept and architecture of the SEF, an important design target was the provision for reuse and scalability of application so as to support virtual practices in thematic domains other than tourism. We have assessed these qualities through small-scale case studies in domains other than tourism (i.e., building a conference plan and scheduling a semester course). As discussed below, both these tasks can be considered as missions undertaken by dedicated squads which progressively move from forming to

storming, norming and performing. Nevertheless the constituent activities of these missions are different from those typically encountered in vacation packages.

5.3.1 Building a conference plan

This case study was motivated by the undertaking to host the 8th International Conference on Computer-based Learning in Science which was hosted by the author's institution of affiliation from 29 June – 6 July, 2007. The case study was performed as a laboratory simulation after the actual event. The corresponding squad comprised representative conference presenters and one moderator. The task was to establish the plan for each day of the conference, indicating parallel sessions, coffee breaks, conference dinners, etc. In a preparatory stage the moderator defined a new neighbourhood namely 'conference' with several representative neighbourhood activities such as keynote speech, paper presentation, workshops, tutorials, coffee break and social event. As part of the package initiation stage, a package family was defined with duration four days corresponding to the actual duration of paper presentations). During package elaboration a proposal for a conference plan was submitted by the moderator and was negotiated in the course of a synchronous collaborative session. As an illustrative example of the outcome, Figure 15 presents the last day of the conference. The remaining days can easily be viewed by scrolling.

The layout of the activity panel is structurally similar to the case of the vacation packages, indicating reuse of the corresponding activity panel and layout manager. Thus neighbourhoods are laid out horizontally representing columns (i.e., the first column collates instances of 'paper presentation', indicating two parallel sessions, the next column lists coffee breaks, etc). In terms of implementation, the new package reuses the expansion pattern to implement the activity panel, while the only component of the SEF requiring modification, although trivial, is the `RadioCheckBoxTree` which this time needs to be populated with activities of the new neighbourhood. As shown in Figure 15 we simply appended these neighbourhood activities into the current list allowing activation through an augmented radio button. For purposes of illustration we have also changed the gradient colours of activities.

5.3.2 Scheduling a semester course

In this case study, the objective was to define the weekly schedule for all courses offered by our department in a semester. Traditionally, this is a time consuming exercise, subject to negotiation between the participants and multiple revisions. Considering the task from the perspective of the SEF and simulating it in a laboratory setting, implies a squad comprising one moderator and representatives of academic stuff of the department. This time neighborhood activities were classified either as lectures or laboratory classes. Initially, the moderator's proposal (i.e., elaboration stage) was identical to the last semester's weekly schedule. This proposal was put forward for negotiation in the context of a synchronous collaborative session Figure 16 presents an extract of the outcome of the exercise as agreed by all participants (i.e., the deployment stage). Worth noticing here is the choice of an alternative activity panel. Specifically, the horizontal alignment represents the types of neighborhood activities, while the vertical alignment represents hours per day. The pop-up dialogue dialogue This instance of the model can then be rendered in HTML and published through the department's web site. The changes required to the eKoNEΣ-SEF were again minimal and amounted to the declaration of the class schedule neighborhood, its relevant activities with the respective gradient colors and the update of the `RadioCheckBoxTree`. The remaining components were fully reused.

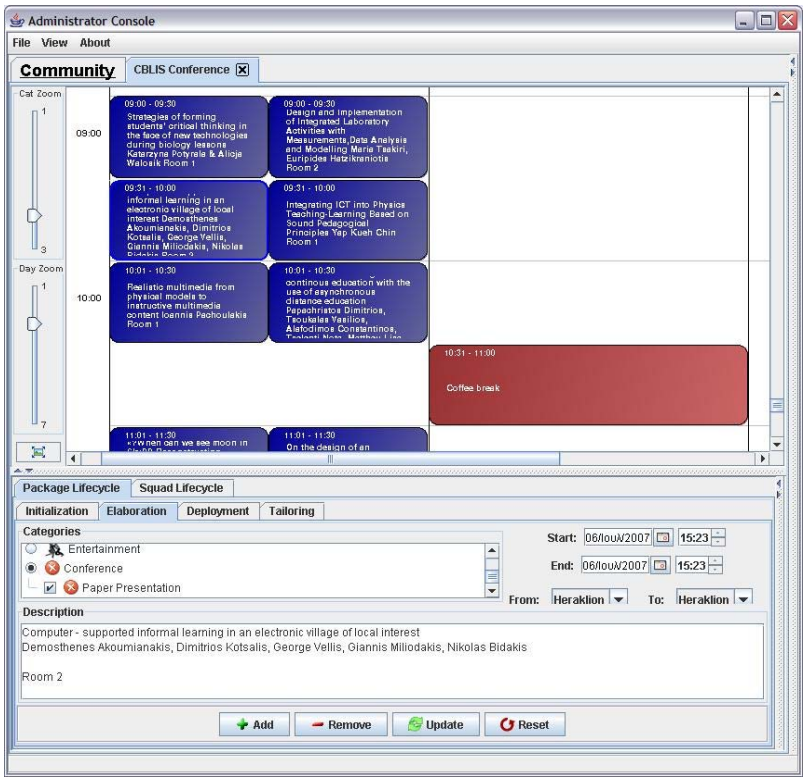


Figure 15: A plan for a conference

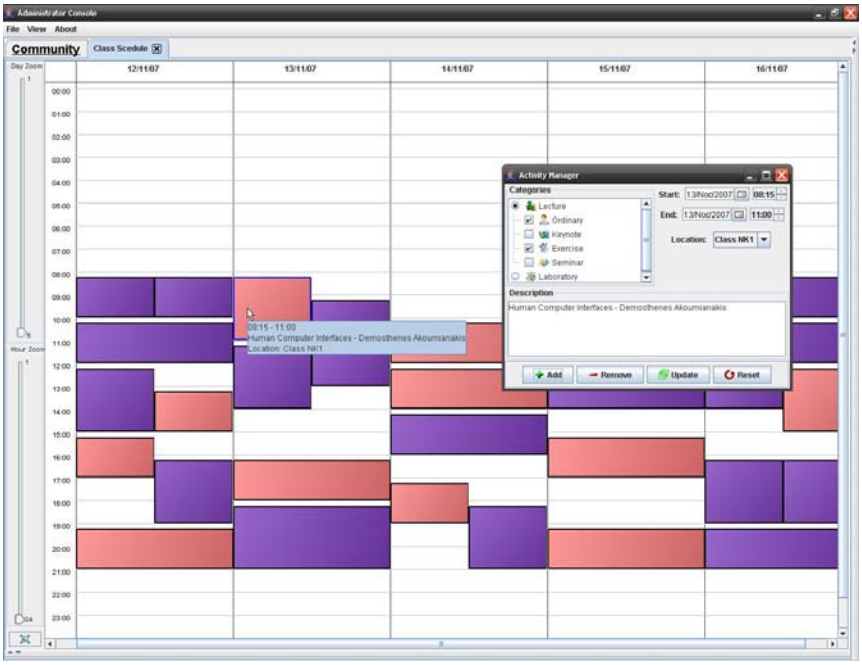


Figure 16: A plan for a semester course

6. Summary and conclusion

In this paper, we have attempted to describe the notion of a social experience factory, how it is substantiated in an eKoNEΣ electronic village of local interest as well as how it scales up to other application domains. The SEF is motivated from Basili's experience factory (Basili, 1993), but is supports slightly different activities and roles. These are informed from sociological research into virtual teams and an analysis of domain-specific work as carried out by virtual communities of practice. The result is a conceptual model and an engineering method for tightly coupling social activities performed in the course of team formation, storming, norming and performing with collaborative workflows such as initiation, elaboration, deployment and tailoring of information-based products.

The SEF has now been used in the area of tourism, which is the main pilot application in which the concept is being validated, but also in other engineering domains through small-scale case studies. These case studies serve a two-fold purpose in the context of the present work. Firstly, they contribute to the verification of the basic operational model of the SEF as described in Figure 1, both in terms of squad lifecycle stages and package development workflows. Secondly, they unfold commonalities which can be generalized across application domains, abstracted to form reusable components and codified to become shared experience through the SEF. Generalizing the results of the case studies leads to expansion of the basic ontological elements of the SEF's underlying design language. Abstracting to provide the SEF with reusable components leads to the development of a common template to model general neighbourhood activities and the development of general interaction patterns for manipulating activities through abstract activity panels. This allows us to support multiple views (i.e., alternative instances of the abstract activity panel) for domain-specific packages. Finally, codifying these to become shared experiences through the SEF entails a number of platform-specific implementation tasks such as building XSLTs for visualizing XML models, introducing web services for downloading packages to local hosts, implementing the abstract activity panel using a target (toolkit-based) vocabulary, allowing for its tailoring (i.e., modifying the colouring scheme used to denote activities of various types), etc. These are skill demanding and programming–intensive tasks, which would not be easily managed without the codified experience of the SEF. As for their technical details, they have been reported elsewhere (i.e., Akoumianakis et al., 2007a, Akoumianakis et al., 2007b and Akoumianakis et al., 2008) providing evidence of the SEF's capability to cope with a range of application domains and engineering problems.

The main contributions of this paper can be summarized as follows. First of all, the SEF provides a frame of reference and a guide for building software tools to support knowledge-based virtual communities of practice in their efforts to construct information-based products by assembling components and reusing experience. As such, it is not only concerned with computer-mediated communication, but instead, it seeks to provide an environment for managing knowledge-based assets and codified experiences in collaborative settings. Secondly, the SEF emphasizes the social aspects of collaborative practicing, in the sense that it links explicitly practice-related outcomes to evolutionary stages of a virtual team's lifecycle. In other words, the outcome of a virtual team is intertwined with the team's level of stability. Thus, a mission is complete only when the team has reached the performance stage. Thirdly, the SEF adopts a model-based approach to establish the fabrics for collaboration. This approach integrates several technological tools to allow role-based access to shared artefacts, adaptable interactive manifestation of domain-specific objects and model editing. Finally, the SEF implements a factory-oriented model for assembling resources into new packages. Such packages are information-based services assembled from components rather than constructed from scratch. Moreover, they represent added value both for the end users (prospective customers) and the coalition members (participating organizations), since no single member of the latter could offer the package cost effectively.

References

Aaen, I., Bøtcher, P., Mathiassen, L. (1997) "Software factories", [Online], *Proceedings of the 20th Information Systems Research Seminar in Scandinavia*, Oslo, http://www.cin.ufpe.br/~in953/lectures/papers/Software_Factories_17.pdf [12 Dec 2007].

Ackerman, M. S. (1998) "Augmenting organizational memory: a field study of Answer Garden", *ACM Transactions on Information Systems*, Vol. 16, No 3, pp 203-24.

Ackerman, M. S. and Palen, L. (1996) "The Zephyr Help Instance: promoting ongoing activity in a CSCW system", *ACM Conference on Human Factors in Computing Systems (ACM CHI '96)*, ACM Press, New York, pp 268 – 275.

Akoumianakis, D., Milolidakis, G., Kotsalis, D., Vellis, G. (2007a) "Interaction platform administration strategies - Practice and experience", *11th Panhellenic Conference on Informatics (PCI-2007)*, 18-23 May, Patras, Greece, New Technologies Publishers, Athens, pp 87-96.

Akoumianakis, D., Vidakis, N., Vellis, G., Milolidakis G., Kotsalis, D. (2007b) "Experience-based social and collaborative performance in an electronic village of local interest: The eKoNEΣ framework", *9th International Conference on Enterprise Information Systems (ICEIS'2007)*, 12-16 June, Funchal, Madeira, Portugal, pp 117-122.

Akoumianakis, D., Vidakis N., Vellis G., Milolidakis G., Kotsalis D. (2008) "Interaction scenarios in the 'social' experience factory': Assembling collaborative artefacts through component reuse and social interaction", *Proceedings of the 3rd IASTED International Conference on Human Computer Interaction*, 17-19 March, Innsbruck, Austria.

Argyris, C., Schon, D. (1978) *Organisational Learning - A theory-in-action perspective*, Addsion-Wesly, Massachusetts.

Basili, V. R. (1993) "The Experience Factory and its relationship to other improvement paradigms", in Somerville, I., and Paul, M. (eds.), *4th European Software Engineering Conference (ESEC), Lecture Notes in Computer Science 717*, Springer-Verlag, London, pp 68 – 83.

Basili, V. R., Lindvall, M., Costa, P. (2001) "Implementing the Experience Factory Concepts as a Set of Experience Bases", [Online], *International Conference on Software Engineering and Knowledge Engineering (SEKE '01) – Conference Proceedings*, Buenos Aires, Argentina, http://www.cs.umd.edu/~basili/publications/proceedings/P90.pdf [12 Dec 2007].

Borghoff, U. (1998) *Information technology for knowledge management*, Journal of Universal Computer Science, Vol. 3, pp 835–842.

Brown, J. S., Duguid, P. (2000) *The Social Life of Information*, Harvard Business School Press, Boston, MA.

Cubranic, D., Murphy, C. G., Singer, J., Booth, S. K. (2004): "Learning From Project History: A Case Study for Software Development," *ACM Conference on Computer Supported Cooperative Work (CSCW '04)*, ACM Press, New York, pp 82-91.

Davenport, T., Prusak, L. (1998) *Working Knowledge: How Organizations Manage What They Know*, Harvard Business School Press, Boston, MA (paperback version published in 2000).

Despres, C., Chauvel, D. (2000) A Thematic Analysis of the Thinking in Knowledge Management. In Charles Despres & Daniele Chauvel (Eds.), *Knowledge Horizons: The Present and the Promise of Knowledge Management*. Butterworth-Heinemann

Dewhurst, F.W., Cegarra Navarro, J.G. (2004) *External communities of practice and relational capital*, The Learning Organization: The International Journal of Knowledge and Organizational Learning Management, Vol. 11 Nos 4/5, pp. 322-31.

Erickson, T., Kellogg, W. (2001) "Knowledge Communities: Online Environments for Supporting Knowledge Management and its Social Context", in Ackerman, M., Volkmar, P., and Wulf, V. (Eds.) *Beyond Knowledge Management: Sharing Expertise*, MIT Press, Cambridge, MA, pp 299-325.

Erickson, T., Smith, D.N., Kellogg, W.A., Laff, M.R., Richards, J.T., and Bradner, E. (1999) "Socially translucent systems: Social proxies, persistent conversation, and the design of Babble", *ACM Conference on Human Factors in Computing Systems (ACM CHI'99)*, ACM Press, New York, pp 72-79.

Greenfield, J., Short, K. (2004) *Software Factories - Assembling Applications with Patterns, Frameworks, Models & Tools*, John Wiley & Sons, New York.

Hackbarth, G., Grover, V. (1999) *The knowledge repository: Organization memory information systems*, Information Systems Management, Vol. 16, No. 3, pp 21–30.

Hidding, G., Catterall, S. (1998) *Anatomy of a Learning Organisation*, Knowledge and Process Management, Vol. 5, pp 3-13.

Jennex, E. M., Olfman, L., (2004) *Assessing Knowledge Management Success/Effectiveness Models*, Proceedings of the 37th Hawaii International Conference on System Sciences, IEEE Computer Society.

Juriado, R., Gustafsson, N. (2007) *Emergent communities of practice in temporary inter-organisational partnerships*, The Learning Organization: The International Journal of Knowledge and Organizational Learning Management, Vol. 14 No. 1, pp. 50-61.

Lave, J., Wegner, E. (1991) *Situated Learning – Legitimate peripheral participation*, Cambridge University Press, Cambridge.

Lueg, C. (2003) "Knowledge sharing in online communities and its relevance to knowledge management in the e-business era*", International Journal of Electronic Business*, Vol. 1, No. 2, pp 140 – 151.

Merono-Cerdan, F. A., Soto-Acosta, P., Lopez-Nicolas, C. (2008) *Analyzing collaborative technologies, effect on performance through intranet use orientations*, Journal of Enterprise Information Management, Vol. 21 No. 1, pp. 39-51.

Morey, D., Maybury, M., Thuraisingham, B. (2000) *Knowledge Management – Classic and contemporary works*, MIT Press, London.

Olson, G., Olson, J. (2000) "Distance matters" *Human-Computer Interaction*, Vol. 15 (2-3), pp 139-178.

Seaman, B. C., Mendonca, G. M., Basili, R. V., Kim, Y-M. (2003) "*User Interface Evaluation and Empirically-Based Evolution of a Prototype Experience Management Tool*", IEEE Transactions on Software Engineering, 29(9), IEEE Computer Society, pp 838-850.

Senge, P. (1990) *The Fifth Discipline - The Art of Practice of the Learning Organisation*, Doubleday Currency, New York.

Simon, H. A. (1973) "The structure of ill-structured problems" *Artificial Intelligence*, Vol. 4, No. 3, pp 181-200.

Takeuchi, H., Nonaka, I. (1995) *The knowledge creating company: How Japanese create the dynamics of innovation*, Oxford University Press, New York.

Tiwana, A. (2000) *The Knowledge Management Toolkit: Practical Techniques for Building Knowledge Management Systems*, Prentice Hall PTR.

Tuckman, B. (1965) "Developmental sequence in small groups" [online], *Psychological Bulletin*, Vol. 63, pp 384-389, http://dennislearningcenter.osu.edu/references/GROUP%20DEV%20ARTICLE.doc [12 Dec 2007].

Wenger, E. (1998) *Communities of practice: Learning, meaning, & identity*, Cambridge University Press, Cambridge.

Wenger, E., Snyder, W. M. (2000) *Communities of Practice: The organizational frontier*, Harvard Business Review, Vol. 78, No. 1, pp 139-145.

Winograd, T. (ed.) (1995) *Bringing Design to Software*, Addison Wesley, New York.

