e-Portfolio, an auto-adaptable grid service with components using Peer-to-Peer agents

Jérôme Lacouture and Saber Mansour

LIUPPA, Université de Pau et des Pays de l’Adour
3 av, Jean Darrigrand 64115 Bayonne Cedex, France
{lacoutur, mansour}@iutbayonne.univ-pau.fr

Abstract. The new trends for long-life training and social activities emphasize on qualification of competence. Our objective is to improve Vocational and Educational Training and specially the process of informal learning by mixing these requirements and new technologies like Grid Computing. The European Learning Grid Infrastructure -ELeGI-project proposes to provide an architecture and some services to promote formal and informal learning on the Grid. In this context, the e-Qualification process is our solution to feature informal acquisitions during training sessions on the Grid. To each individual is associated an e-Portfolio service where are reported his activities, knowledge, competencies and objectives. In this paper, we insist on the dynamic character of the training and we propose some ways for the adaptation and composition of dynamic grid services between users. Our approach is situated at the convergence of components and agents paradigms, components producing adaptable grid services and agents dynamically composing them. Our flexible approach is not limited to the area of learning. It is also applicable to others similar grid applications.

Keywords: Virtual Communities, e-Qualification, e-Portfolio, Adaptable Components, P2P Agents, ARS model.

1 Introduction

Nowadays, all European Countries are looking for improving their education systems by using new technologies; that’s why new long-life training and social activities trends offer an alternate and promising avenue of research. Mixing these requirements and new technologies like Grid Computing [FKNT02], the European Learning Grid Infrastructure -ELeGI- project proposes to provide an architecture and some services to promote formal and informal learning on the Grid [GRS03]. Other projects like ICEAGE

\footnote{http://www.iceage-eu.org/v2/index.cfm} have taken advantage of the combination of grid and education. Grid technologies could be useful in terms of security, QoS, decentralization, large-scale services and allowing the production of stateful and transient educational services.

In this context, we introduce a solution to feature informal acquisitions during training sessions on the Grid: the e-Qualification process. This process is
based on the design of "Communities of Practice" [Wen00] and "Communities of Competencies" [Smi05] used as a real environment that promote collaborative learning. According to ScotFEICT (ICT in Scottish Further Education), "E-qualification" is a collective name for a suite of qualifications, which is related to online learning (e-learning) and online assessment (e-assessment). However, E-qualification in ELEGI is not a new e-learning scenario and differs from e-assessment (and from the above definition).

The major objective of the e-Qualification process is to bring a solution to the two following research questions: "How do we optimally design an operational e-Qualification process for competence-based training environment?" [KK01] [BRSJ03] and "how are students actually assessed when they learn in informal learning mode during training sessions on simulated environments?" [GMNY05].

This leads to significant improvements in the methodology and results of training and addresses the following aspects of training: procedure assessment considering human factors, trainee's assessment and self-training, human factors in task oriented environment, awareness of the maintenance technician, improved planning of tasks and training in a Virtual Environment.

The procedure is to let students work in an experimental condition and monitor their behaviors and work in the electronic environment. So, we propose to found the e-Qualification process on collaborative learning (to promote informal learning) and on the reciprocity strategy (each learner influences the other, and vice versa). That implies the introduction of a major tool: the e-Portfolio. Indeed, each learner needs to be guided and followed throughout life. Thus, in ELEGI, we introduce the e-Portfolio as a real shared workspace, which has the following characteristics:

- CV e-portfolio, which is the learner's profile.
- Working e-Portfolio, which represents the collaborative dimension of the e-Portfolio to interact with others.
- e-Portfolio of evidence, which is an additional dimension to increase the quality of work sharing by qualifying learners against a scale of competencies.

This new extended view (e-Qualification) makes training calculable, with informal learning activities and it brought up additional facilities to: Everywhere, Anytime, Any mode (formal/informal) for individual as well as groups (virtual communities).

Resolving e-Qualification problem in the open and dynamic environment that is the Grid leads us to think about other grid problems. In fact, we have to provide solutions to form and manage virtual organizations in such environment. We have also to offer an architecture making it easy to develop, deploy and compose grid services. We resolved these problems, with the combination of Agent and Component technology, for the e-Qualification case but the solutions we propose can be easily reused in other similar grid applications.

The goal of the paper is to show that components and peer-to-peer agents could be a good solution to design adaptable and autonomous entities. This proposition is illustrated in the context of e-learning on the Grid. In this paper we present how users equipped with e-Portfolios (grid service) can be dy-
namically qualified during their training process. The section two deals with the e-Portfolios role. The section three broadly presents the e-Qualification process where users evolve dynamically integrating virtual communities thanks to their service of e-Portfolio. In section four, we propose a component approach to produce adaptable e-Portfolio services. And in section five and six, we compose dynamically services, via Peer-to-Peer agents grouping users to share competencies.

2 How could e-Portfolio assist human capital management?

According to [RW05], the term portfolio as used in the UK generally describes a collection (or archive) of reflective writing and associated evidence. This is the main reason why the e-Portfolio used to e-Qualification is substantially different from the various forms described above:

- Virtual socialization skills, *Personalized Ontology & Learner Profile*
- Virtual collaboration skills, *Shared work Space*
- Technical skills & *vocational competencies*

In this paper we provide a wide range of improvements and new uses extending the relationship between three e-Portfolio facets for individuals distributed as networked shared workspaces on the Grid. It proposes an information model that might lead to defining future specifications for e-Portfolios tightly coupled with "on demand" ontology on the Learning Grid Architecture (ELEGi). When group negotiated added values (contracts) are included in individual e-Portfolios (signature), a new qualification must be given to the learner, which may be achieved through an assignation into a Community of Competence where groups members have the same level of qualification. According to this knowledge management architecture (service oriented), the lack of feedback on the informal learning process is partially but effectively made up by the traceability of the evolutif pedagogical contract managed by Peer-to-Peer (P2P) agents inside a community of Practice, which allows open dialogs between learners at work [GMNY05].

3 The e-Qualification process

In this section, we present the operational steps of the e-Qualification process:

Building of a Community of Practice (CoP): a selection of trainees (trainee profile), that are going to make the same exercise, is done by the instructor and last session results are stored in the trainees e-Portfolio.

Briefing: the instructor describes the exercises and assigns each trainee’s objectives (stored in the trainee’s e-portfolio).

Training: each trainee exercises on the simulator, system shows guiding messages. The trainee gets a continuous evaluation sheet of parameters. The
Fig. 1. Contracts’ matchmaking (competence extraction)

instructor receives continuous reports. The trainee communicates verbally or electronically with other trainees, peer-to-peer agents gather users by contracts during the exercise. The results and the parameters are stored in the trainee’s e-Portfolio. Figure 1 shows how a trainee (user A) with an e-Portfolio can improve its knowledge sharing competencies in contracts. P2P agents observing the evolution of the trainee manage interactions. Playing roles in contracts, user A obtained new knowledge until he performs his initial goals.

Debriefing: Based on the update of each e-Portfolio, debriefing is managed in peer-to-peer mode.

During the simulation, the e-Portfolio is continuously updated with the trainee’s actions and results. Informal added notes can be integrated to the e-Portfolio during collaborative exchanges. The perpetual evolution of competencies, and contracts negotiated with unpredictable needs, relative to each trainee objectives and desire, underline the need to adapt and compose e-portfolios dynamically.

4 e-Portfolio, an adaptable grid service

Indeed, one of the major skills of current and future distributed systems, and particularly with the semantic grid, is their ability to integrate and co-ordinate available services in on-demand way. The composition of “open” services becomes a central research area [LP06]. The dynamicity is an ever prevailing constraint in the area of learning, and especially for informal learning activities. Debriefing phases between learners add a problem of predictability. Therefore, on-demand solutions have to be proposed.

The idea to resort to the composing paradigm is justified by the results obtained, on the one hand, in the field of the re-use, and on the other hand, on the field of the modelling and the deployment of distributed applications (like grid services).

In the literature, the ”composing” term is largely associated to the re-use. The reusable components are represented using a model of components. Two
essential principles must ideally underlie these models \cite{CS99}: the principle of abstraction and the principle of variability.

The abstraction consists in explicitly distinguish in a component, reusable knowledge (the realization of the component) and knowledge useful to its realization (specification of the component). The realization is the "hidden" part. The specification is the "visible" part of the component. This principle is necessary to build effective infrastructures of re-use. Indeed, the specification part of a component can play an essential role in the exploitation of re-use’s architecture, in particular at the time of the search and the integration of components.

The principle of variability result in distinguishing in the specification of a component a fixed part and a variable part. These two parts are visible in the specification of the component; they introduce the generic character of the component. At the re-use time, it is by fixing the variable part that we choose a particular realization and that we adapt the component according to specificities of the system under development.

4.1 Adaptable components

We integrate these considerations extending a model of component developed in our laboratory for the static integration of components. The model of components called "Ugatze" \cite{Sey04}, adapted to the re-use of autonomous, heterogeneous and distributed software components, falls under the field of the Model Driven Engineering (MDE). This model rests on two essential concepts: the interface of the component and the interaction between components.

The context of the last works led us to handle components defined as abstractions of existing software entities \cite{LP06}. Concretely, they can be programs, applications, services or entities whose "granularity" is important. (for instance: e-Portfolios grid service).

We qualify these components of "high level of abstraction" and they are characterized by the property of autonomy, autonomy of design, and autonomy with the execution. In our extended model (figure 2) the specification of component is based on the concept of "interface". Our interface is made of "specification points". We propose specification points to combine abstraction and variability principles presented in section 2. We distinguish two types of specification points: interaction points and adaptation points.

Interactions points are produced to integrate the component in the system for its re-use. A classification of interaction points is available in \cite{Sey04}.

Adaptation points are associated to each sub-service of a component to adapt the component. We detail this kind of specification point in the next part.

4.2 Adaptation points

In our model (Figure 1), the variability principle is based on the concepts of "core" and "sub-services". The realization is viewed as a core (fixed part), which we associate sub-services (variable part).
At the specification level, a specification point is associated to each sub-service and we call it "adaptation point". It allows to select a sub-service and to adapt the component to specific needs.

To each component we have to define the adaptation points provided. This step consists in distinguish which sub-services a component provides, or which competencies are sharable. Note that is possible that a component provides an only one sub-service.

To each adaptation point is associated a metadata description. This description comprises two parts: a functional part and a non-functional part (Figure 3). The functional part indicates what is the sub-service’s function. It describes the service itself. The semantic is crucial for this kind of description. A common notation must be used for the description of the different components.

Ontology approaches are often associated to this effort [RAG+05]. Indeed, the referencing of concepts is a largely popular solution to share and interpret the provided descriptions. We opt to use a common ontology (domain ontology) as the most adequate solution.

The non-functional part allows to specify certain properties. Sharing services (competencies, knowledge and so on) have to take into account essential criteria.
like Quality of Service, security, performance or reliability. Thus, non-functional properties allow to choose the best available composition if several components can provide the same services. Of course, non-functional properties depend of the whole system. (It is not an issue addressed in this paper).

4.3 Adaptable e-Portfolio component

A typical scenario of adaptation of an e-Portfolio occurs during informal sharing in communities of practice. Trainees want to improve their knowledge when their objectives are not completed. Other trainees having the required competencies share their competencies joining groups (CoCs).

This e-Portfolio is a personal service describing competencies and the training of a user. Like a kind of curriculum vitae of the learner, it also contains its knowledge and its future exercises. But more than a simple profile it is a real shared workspace tool for a collaborative activity with other learners.

Each e-Portfolio evolves according to the knowledge obtained and exchanged during practiced activities. E-Portfolio is an adaptive component reacting to its owner needs and also reacting to other actors needs for collaborative activities. Consequently, we design each e-Portfolio with our model of component. An e-

![Fig. 4.](image)

Fig. 4. e-Portfolio of a learner, zoom on a adaptation point providing a landing tutoring procedure

PorteFolio has a core and sub-services. For instance, we find sub-services to allow a learner to acquire new knowledge or to help a friend to understand some elements. The Figure 4 gives an example (simplified) of an adaptation point provided to help a pilot to learn a landing procedure. We find the functional part (what the sub-service propose, level of the tutor), and the non-functional part (security and confidence level, the confidence is improved with successful sharing).

Our goal is to distribute users toward groups sharing compatible competencies. In the next section, we present how agents (P2P agents) help for the discovery of services and their organization in groups reasoning on competencies available inside individual’s e-portfolios components.
5 Agents for auto-organized and auto-adapted grid component systems

More and more researchers recognize that agents and components have complementary advantages [Bri05]. On the one hand, component concepts are useful for the design and the deployment of modular and reusable multi-agent systems. And, on the other hand, autonomy, adaptation and coordination are relevant for the components and their composition.

Usually, components are indexed in catalogues to be re-used. The counterpart is that the effort required to re-use them is considerable for the developer, often by a lack of knowledge of the component to be re-used.

We propose to make components more adaptable by integrating an agent approach into our model to make the process of research, adaptation and composition, as transparent as possible.

So, we choose to associate to each component an agent able to react to two main types of events: internal events (to monitor new needs publish by the component itself) or external events (to answer to other components needs).

5.1 Peer-to-Peer Agents supporting Grid Components

In the scope of this paper an agent represents one component. An Agent is continuously controlling the component’s evolution. A Component, representing an e-Portfolio, can, in the same time, offer and need sub-services. Requirements of component are specified in term of composition. Each new sub-service needs to be associated with similar sub-services of other components. Rather than use an existing type for agent, we have chosen to create our own. Our agents (PeerAgents) use the JXTA platform [JXT03] to process in Peer mode. The reasons to us are clear. First of all, our work will be used in the Grid, a very dynamic and decentralized environment. So we are targeting P2P networks rather than the traditional client/server networks where agents types have been applied because they are interesting in terms of decentralization and extensibility, fault tolerance, adaptativity and dynamicity (add/delete peers), and capacity to virtualize resources; secondly, we want to take advantage of the JXTA protocols for simplicity and efficiency, and this approach is more easily accomplished by using the research features these protocols provide which are explicitly directed at P2P networks; In a third place, sometimes when it is appropriate to use existing robust technology, as the capabilities offered by JXTA platform, and integrates it with the work that has preceded. And finally, with this approach we can rapidly prototype and test solutions on P2P networks, finding their strengths and weaknesses.

The responsibility of the agent is to find the existing agents having roughly description of the new sub-service.

5.2 The ARS model

The ARS model is a conceptual model based on three major concepts: Agents, Roles and Signature. The model is designed to easily manage open groups with
fluid boundaries and used in dynamic and rapidly changing environment. In such environment there is no guaranty that a member is online at a given time. Moreover members can move frequently from one group to another, the whole organisation of agents is not predefined, neither are the rules that manage the groups. The decentralisation and dynamic structure seems to be, in such conditions, the only solutions to avoid having ”systems” with predefined groups and to efficiently manage them. The Grid environment is an ideal area where ARS could be considered as a suitable solution to solve the problems of dynamic and openness that many other organisational models in MAS don’t solve. The ARS model is not designed to a specific kind of application and aims to solve all problems where dynamicity is a main concept and where systems are open and their structure can’t be designed before execution time. ARS is flexible enough to support applications where requirements are very different from classic group management in MAS field to composition of services in the Component.

We do not intend to address these aspects and the way ARS resolve all these problems in this paper. Rather, our goal is to show that the combination of agent technology (Madkit) [GF00] with the JXTA P2P core protocols will help to eliminate some of the barriers to component deployment in a largely distributed and very dynamic environment.

5.3 The ARS concepts

Agent: An agent, as defined by Ferber [Fer01], is a physical or virtual entity which is capable of acting in an environment, communicating directly with other agents. It is driven by a set of tendencies (in the form of individual objectives or of a satisfaction/survival function which it tries to optimize). In the ARS model we impose that agents should behave in a Peer-to-Peer mode. They have to advertise their capabilities, accept to be equal to other agents and to offer sub-services they can provide and answer when possible to queries of other agents. All the agents of the same group are considered as equal and have especially a common task to do: managing their group. We design our solution on the JXTA technology that provides P2P facilities (creation, search, connection, advertisement, discovery, and so on) adopted by agents implemented with the Madkit platform. An agent has a description of its capabilities that will be published as a JXTA Advertisement, so that it can easily be discovered by other members of the network.

Role: Agents play roles in order to reach goals. Agent plays one or more roles in one or several groups. Each role has a set of rules that specify conditions allowing agent to play it and verify continuously that agents respect rule’s obligations. To play a specific role, an agent should satisfy its conditions and engage itself to respect the rules managing the role. From the agent’s point of view the role can be in the following states:

- Asked: When the agent is waiting for an answer to its request to play the role.
- Accepted: This state corresponds to the case where the agent’s request to play the role is accepted (by the signature).
– Refused: When the agent receives a negative answer for its request the state became: refused.
– Failed: A role is in a failed state when the agent doesn’t respect one of the obligations.

**Signature**: We can define it as follows: the signature (figure 5) is the mesh that binds a lot of agents, makes them grouped and allows them to interact and to be organized in a decentralized way. It contains rules that manage it and that should be respected by all the members of the group. It has a set of attributes and stored data. The signature stores also the list of goals that an agent can achieve by joining the group. To reach these goals members should play roles providing these goals in the group. Respecting the rules of the group and is specified via social contracts. Moreover, being a member of a group is a commitment towards all the members to respect all the rules of the signature. When a member plays a role it engages itself to respects its obligations and this is also a commitment. The signature could also be seen as a mirror by which

agents observe each others. In fact constraints on signature’ fields that members should respect offer a view on the approximate state of members. For instance, if we consider the case of landing tutoring procedure the attributes (Flaps, Navigation, Landing speed and order) can have a tolerance values (for flaps : 10 and minimum is 8 and maximum is 11) this is an implicit information about members. This reflection allows an indirect interaction between agents, and consequently between components. In fact, the signature contains rules and evolves continuously when components evolve. When one of the signature’s parameters changes in some sub-services of the component, the signature can change consequently. The evolution of signature due to these changes or to the entry of a new member should be managed. Thanks to the mechanism of inference, rules will decide whether the received signature is acceptable or not. The agent will then return back to the requester the answer resulting from the rules inference. All the requests treated by any member will have the same answer since the rules used are the same. From the management point of view all the members are peers,

![Fig. 5. The signature structure](image)
there is no "super" member. The signature of a group is published through the P2P network via JXTA advertisement. An agent uses their capacities of research provided by JXTA to get an advertisement of a specific role it wants to play and the group it needs to join. In fact each signature contains the description of its roles. Agents have to construct queries based on capacities that will match with one or many roles exiting in some groups and published in the network.

6 ARS model allowing adaptability

As we said previously, agents are continuously listening to their components evolution. As soon as a new event happens the agent will treat it. If the Component creates a new sub-service, the agent will advertise it and try to join the group that corresponds best to the new created sub-service and recalculates its appurtenance in different groups. In order to join the suitable group the agent creates a request containing the description and the functional properties of the new sub-service. From the Agent’s point of view, sub-services are seen as roles played by agents. In fact, there are types of roles, and instances of roles. Agents play roles that are instances of types of roles. Instances don’t have exactly the same characteristics but respect the same rules and ontology. For simplicity reasons we will not make difference in this article between role and type of role and will use for both concept the term role. The request contains, in addition of the searched role’s criteria, some personal information describing the component holder, level, known ontology, and so on. In the next sections we explain how groups are managed and how such request is treated.

6.1 The Group’s life cycle

Before explaining how P2P agents manage Groups let’s present the Group’s life cycle. This cycle is made up of four phases: creation, operations, evolution and dissolution.

Creation : When a Peer agent representing the component doesn’t receive any positive answer to the request it sends to the different existing Groups in order to join one of them, that means its signature doesn’t correspond to any existing Group’s signature. In other words, no existing group contains the role it is searching for. Two cases are possible. The agent can add a new type of role in one of the groups it belongs to. This is possible if the ontology on which the group’s signature is based supports the ontology used to describe the new Role. In the second case the agent can’t add its role to any one of the groups where it is a member. In this case, the agent creates a new Group, adds a new type of role. It specifies the rules and the goals of this Group depending on the goals and requirements of its Component. The signature of this new group will be based on the signature it broadcast and that was refused by other groups. When an agent sends a request that corresponds to the new role created, the new group will be the only answer it receives.

Operations : There are two possible operations in a Group :
Joining: This operation is based on the use of signature. The representative agent of the learner who wants to integrate the group enters in communication with one member agent (the one that is playing randomly the role of manager). The former agent sends to the later its proposed signature. The agent that will receive the signature will compare it with the group’s one. Here, we can see the real utility of the ontology: the member agent thanks to a set of rules will try the overlay between the signature of the group and the learner’s one. If the overlay succeeds (with a predefined tolerance), the two agents will discuss the different conditions to enter into this Group. As soon as it has been authorized to integrate the community, the new member informs the other agents, via a broadcasted message, that it is a new member.

Leaving: A Component has to quit a Group when its signature does not match anymore with the Group’s one. Its agent has to inform the whole members that he leaves the group. The agent will recalculate the signature of the Component and creates a new one based on the old one with the new capabilities and goals of the component. Then it will search a new Group that accepts its component with the new created signature.

Evolution: This situation can occur in two cases. When a new agent integrates the Group, members discuss about what this member can bring to the community and choose whether they update the signature (for example, adding new objectives) or not. This decision is done automatically thanks to the rules that govern the community. Since each member has a copy of the Group’s signature, they will all use the same rules so they will apply the same changes and will have the same signature updated in a decentralized way. Here we can observe the importance of signature and the concept of rules to manage communities in a decentralized mode.

Dissolution: A Group disappears when it contains zero members. The Group should be deleted by a human actor or automatically after the departure of the last member. This depends on the way we want to configure the Groups. In our example, this is left to the constructor who decides if a Group can be useful or no in which case he deletes it.

6.2 Managing Groups

We mean by managing Groups the way by which new members are allowed to enter. Who gives them the authorization to join the group? How updates occur in the Group? When and who is the responsible of all these tasks?

As on the one hand our work aims to be used in the grid environment, and on the other hand learners in the informal learning are all equals and there is no central member, we consider managing groups as a decentralized process. Each member agent of a group must be able to manage its group in a decentralized way.

- Managing new entries: Since all the agents share the same signature and are engaged to respect its rules, every agent could be at any moment a manager. In fact the signature contains the rules and the conditions that allows a new member to join or not the group. We propose a mechanism of randomization
allowing agents to play the manager role. At the reception of a new candidature from a requester agent, the current manager, will compare the received signature with the group’s one and compare rules thanks to mechanism of inference, this will decide whether the received signature is acceptable or not. The agent will then return back to the requester the answer resulting from the rules inference. By this way, if an agent is not available for any reason (network link broken, agent killed, and so on) another agent will become the current manager and the system works normally without disturbance. By this way the system is managed in a decentralized way. We remark that all the requests treated by any member will have the same answer since the rules used are the same.

- Managing Group’s signature update : When a component changes, and if one of its signatures in one of the groups it belongs to doesn’t match signature with one Group’s one, rules of the signature are invoked and generate a new fact. This fact will be interpreted by the PeerAgent, as a request for leaving the Group and will search for a new Group that corresponds better with the new profile of its component.

Therefore, some rules could specify update’s criteria. These rules specify how signature’s parameters can change. For instance, in the landing tutoring procedure if the average of the members’ score in a specific role (flaps for example) exceeds the current lowest score allowed in the signature by four (new average = lowest acceptable score + 4), a rule changes the lowest allowed score to the new average. This kind of rule manages the signature’s update and consequently manages the group. In fact, any agent that has its role (the service in the component view) with a score lower than the new lowest value will have to leave the group. We distinguish three types of rule :

Non functional rules : These rules specify how the signature is updated, how the events that require a signature are updated and also precise the conditions to access the group (ontology verification, capacities needed and constraints on some data).

Functional rules : These rules structure the relations roles-agents. They specify the conditions to play a specific role, the obligations and capacities that the agent should respect when playing a role. Some of these rules reward agents providing sub-services offered by the role capacities to other agents. They also punish agents that don’t respect role obligations.

Meta-rules : In order to satisfy group dynamicity, the structure of the group should change as members evolve. The structure is managed by functional and non functional rules, modifying these rules dynamically means modifying the group structure.

To conclude we can say that agents of the ARS model guide self-adaptation of components. Therefore, they have these functions and capabilities :

- Access to the adaptation points of the components. On the one hand, the agent verifies if an internal need is present to try to find a composition (by listening to new events in the component), and on the other hand the agent manages the collaborative sharing of services.
– Search for capabilities based on search patterns. The search is done in a P2P mode adopted by every agents of ARS model. We implement the mechanisms of research via the JXTA platform.

– Publish services. JXTA advertisements are used to publish services of components. These services correspond to roles in the ARS model.

7 Conclusion

The first idea is that a generic high-level service of distant qualification on Grid Communities can be supported in the very system itself by empowering the users to participate in the set up of Virtual Communities, by personalization, and ambient learning. What the e-Qualification is bringing when peer members share their acquired expertise is not limited to the e-learning domain but is also applicable to other problem domains.

A second strong idea states that e-Qualification largely controls motivation for learning (mostly for informal learning) and whether students achieve their goals. We want users to become independent, lifelong learners in Vocational Education and Training. Therefore they learn to be responsible for their own learning. However, Virtual Learning Community should involve them as accredited members in qualification groups (CoC), after they have collaborated as peer members in a Community of Practice. This is consistent with our training values and practices.

E-Qualification plays a crucial role as "continuous" moderating system in mediating between the informal dialogues of a user community and the available content extracted from their e-Portfolios.

Beyond the context of learning, the overall goal of this paper is to show that the marriage of the Agent Technology based on P2P networks, with JXTA, and the component paradigm is an ideal match. It is a promising way to resolve complex problems such e-Qualification, and more generally grid services composition problems.

In relation to this idea, we have presented a flexible model of component intended to adapt and compose dynamically grid services (e-Portfolios in our application). We have focus on adaptation points presenting available competencies with functional and non-functional properties.

The other contribution is the management of a flexible e-Portfolio component by P2P agents according to the ARS model.

Acknowledgment

"Work partially supported by the European Community under the Information Society Technologies (IST) programme of the 6th Framework Programme for RTD - project ELeGI, contract IST-002205. This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing there in".
References


