

DIOGENE:

A service Oriented Virtual Organisation for e-Learning

Matteo Gaeta*, Nicola Capuano, Angelo Gaeta, Francesco Orciuoli, Laura Pappacena, Pierluigi Ritrovato

**Dipartimento di Ingegneria dell'Informazione e Matematica Applicata
Università degli Studi di Salerno*

*Via Ponte Don Melillo – 84084 Fisciano (SA), Italia
CRMPA Centro di Ricerca in Matematica Pura ed Applicata
C/O DIIMA – Università degli Studi di Salerno*

*Via Ponte Don Melillo – 84084 Fisciano (SA), Italia
gaeta@diima.unisa.it, {capuano, agaeta, pappacena, ritrovato,
orciuoli}@crmpa.unisa.it*

Abstract. In this paper we present the main results of the DIOGENE project where the characteristics of Virtual Organisations providing learning services have been identified and implemented using state of the art Web Services technologies. We also present a possible migration path towards the Grid emphasising the advantages stemming from the adoption of this technology.

Keywords. Virtual Organisations, Web services, Grid, Learning services

Introduction

Virtual Organizations (VOs) can be defined as a coordinated group of individuals and/or institutions who collaborate towards a common interest and share, on the basis of some policies, a set of resources (processors, instruments, data and services). The element that characterizes the participants of a VO is the common *purpose or business goal*. The participants of a VO share resources and establish sharing rules in order to obtain a common objective. Members of a VO can be geographically distributed and can access to the resources any time they are allowed.

In order to realize a VO, many problems must be resolved. Indeed, the VO is a dynamic environment, in which resources and/or individuals can join or resign in dynamic way. The resources are *heterogeneous, distributed* and *shared*. The resource sharing must be controlled by a set of rules that guarantee security, through mechanisms of authentication and authorization. The resources must be monitored in real time, since their state change quickly. The VO is a highly scalable environment, so it needs mechanisms for the discovery of the single resources. In brief, for being accessible and usable any time and any place technology solutions are crucial elements.

Among the technologies available for the realization of a VO, the Web service and the Grid ones offer many advantages. They are open and provide mechanisms to address the aforementioned issues. A key factor for the realization of a VO is the adoption of technologies based on industrial standards. For example, the VO paradigm is applicable inside the boundaries of an organization structure (an intra-organization

view) and across organization boundaries (an extra-organization view) but, currently, only Web Services and/or Grid technologies allow the realization of the latter view.

The following sections present the Diogene VO, which is a Web Service based VO for learning experience. We will describe the most important components of this VO and a possible deployment of the Diogene VO in a Grid infrastructure. We will end with an overview of some interesting features of the upcoming Microsoft operating system, code name Longhorn, which provides an innovative file system and communication infrastructure very useful to develop semantic and collaboration based applications.

1. THE Diogene virtual organisation

Diogene [1] is an EC funded project aimed at design, implementing and evaluating with real users an e-learning Web brokering environment for ICT individual training able to support the learners during the whole cycle of the training, from the objectives definition to the results assessment through the construction of customised courses.

The e-learning system which was developed in the framework of the Diogene project, uses state-of-the-art technologies like metadata and ontologies for knowledge representation and management, fuzzy learner modelling, intelligent course tailoring and includes a set of innovative features such as dynamic learning strategies, Semantic Web openness, Web services for Learning Objects handling and IPR management, Curriculum Vitae maintenance and searching facilities, free-lance teachers support and assisted Learning Objectives definition [2].

The Diogene scenario is modelled as a Virtual Organisation (VO) strongly based on the use of Web Services. The Diogene VO is populated by entities offering and consuming services. Such entities are grouped into four categories: **users** (human beings that provide and consume services), **organizations** (physical entities that as software components provide and consume services), **utility components** (software components embedding the technology assuring the architecture integrity) and **accessory components** (software components providing the access from outside to the Diogene VO and from the Diogene VO to outside).

1.1. Diogene Users

Diogene Users can be grouped into the following main categories (it is important to note that the same physical user can play different roles in Diogene so he/she can belong to more than one of these categories).

- Learner: is a student registered in the Diogene VO. He/she acquires knowledge through learning experiences provided by Diogene specific learning services.
- Expert Learner: is a Learner skilled about some topics. He/she can offer mentoring support to other learners about such topics, usually for free.
- Freelance Tutor: is a professional tutor offering his/her specialized mentoring support to learners about specific topics usually under the payment of some price.
- Skill Searcher is an enterprise manager interested in hiring certified staff. He/she performs queries on skill repositories.

Diogene includes further sets of users like Content Providers, Knowledge Managers and Administrators that, to improve simplicity and clearness, are not discussed in this paper.

1.2. Diogene Organisations

Organizations of different type offer services with respect to the specifications of the Diogene VO and collaborate for the realisation of their own services. The kind of organisations supported by Diogene are summarised in the following list.

- Publishing Houses (PH). They store training content and provide remote access to it. They provide search and retrieval functions on the local repository via metadata-based queries.
- Web Catcher Agencies (WA). They are able to extract training content directly from Web and Semantic Web. Through a keyword-based text categorisation algorithm they can, where absent, automatically extract metadata from textual learning objects and link them to structures maintained by the KA.
- Tutor Agencies (TG). They work as entry point for freelance tutors; they manage the freelance tutor archive (containing tutor models) and provide searching facilities on such archive.
- Brokerage Offices (BO). They research, prepare and provide training offers for learners on demand, based on customisation information (learner model, learning strategies, price, learning goals). They collaborate with PH and WA in order to retrieve didactical material that best fit the customisation requests. They collaborate with TG to retrieve and provide freelance tutors that best fit learner needs.
- Training Agencies (TA). They provide the basic environment for learning experiences and tutoring activities. They are responsible for the delivery of courses and for the provision of course management and execution functions. They maintain learner models and are able to provide (on demand) course offers by exploiting services offered by BO.
- Knowledge Agencies (KA). They maintain and manage knowledge structures (concept dictionaries and ontologies) for the whole Diogene Network.
- Café (CA). They support social interactions, mentoring and information exchange by providing users a set of collaborative synchronous and asynchronous facilities. They automatically arrange groups among users of registered TA by individuating and grouping learners with similar needs and/or profiles.
- Skill Agencies (SA). They provide search engine capabilities on Learner Models Databases of registered TA in order to let third parties interested to hire certified staff to find qualified professional (with respect to privacy requirements). They maintain, moreover, requests statistics in order to rank the required competencies.
- Bank (BA). It's a singleton organisation that executes and logs transactions information occurred inside the Diogene VO. It interacts with an e-commerce engine to execute transactions involving real payment processes.

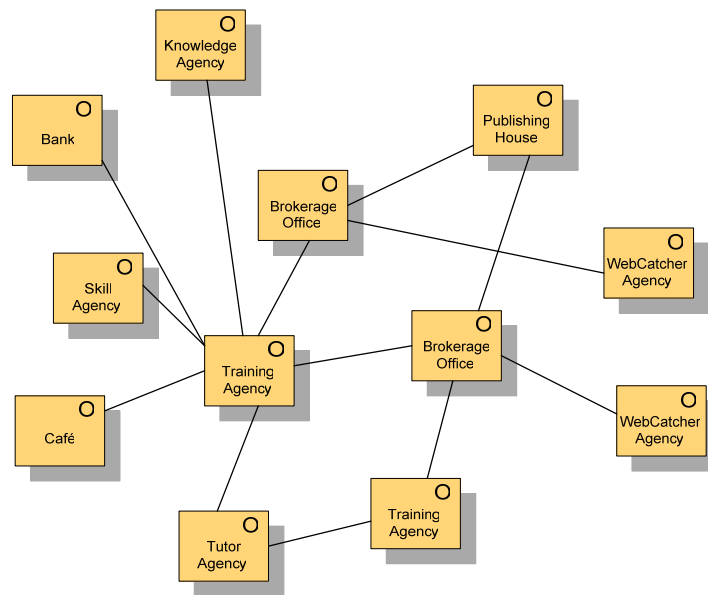


Figure 1 – Interactions among Diogene Organisations

More instances of the same Organisations live inside the VO. The various instances stand on different hosts and are managed (administered) by different users. Figure 1 depicts some interactions between Diogene organisations.

Following the principles of the Convergent Architecture [3], Diogene Organizations can manage Processes, Resources and other Organizations. *Resources* live encapsulated within the Organizations and collaborate with them to accomplish their own tasks. Resources can also be used as containers for the information exchanged between organisations. Resources can manage its own data and other Resources.

The *Processes* express and are realised by the collaboration between Organisations or between Organisations and internal Resources. The Processes can create and consume/use Resources and other Processes.

1.3. Utility and Accessor Components

Utility Components have no business-domain relevance but embed the technologies that explicitly assure the Architecture integrity. The Diogene VO includes the following two singleton components.

- **Authentication Agency:** it provides all services for the registration and authentication of users of the whole Diogene VO. After authentication, it releases a token that must be used as a ticket to access and be authenticated in all VO services.
- **Diogene Services Registry:** it includes Diogene White Pages (containing information about services offered by organisations), Yellow Pages (organizing services in categories) and Green Pages (including information about how to execute services).

Accessor Components provide the access to the Diogene VO services and also the access through the Diogene VO components to external services. The only Accessor Component of the Diogene VO is the following.

- Diogene Network Reception: the Internet entry point for all Diogene users. It provided features of user and organisation registration, it publishes characteristics and services offered by the Diogene VO, it provides administration features for the Services Registry.

1.4. Some Orchestrated Scenarios

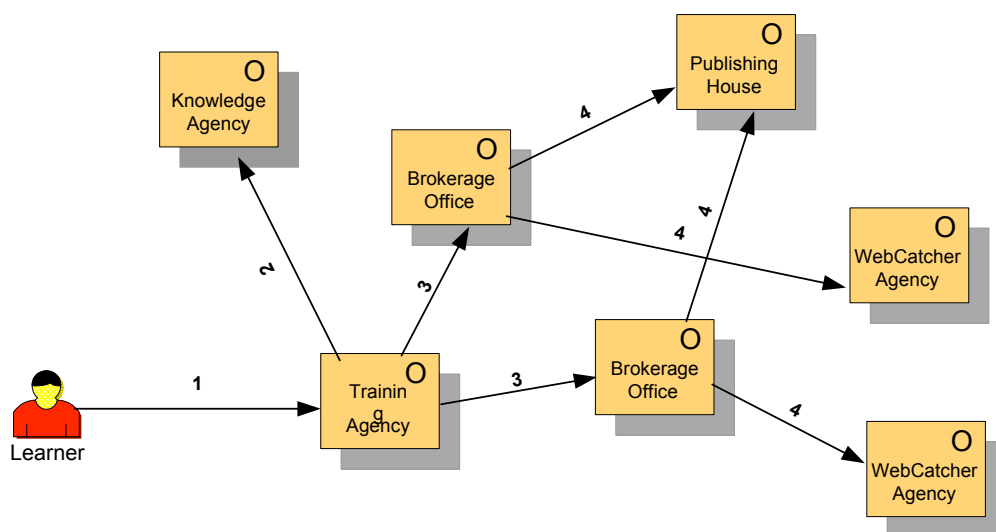


Figure 2 – Orchestrated scenario a)

In a first scenario, a **Learner** asks for a personalized training offer to his/her **Training Agency(1)** through its entry point (Training Agency web portal), starting from a concepts list that he/she wants to learn (Target Concepts). Training Agency calls the **Knowledge Agency(2)** passing to it the target concepts in order to build the learning path; it will be composed by all concepts necessities to the learner to reach the target concepts. Knowledge Agency builds the learning path using the owned knowledge structure, ontology and concepts dictionary, and the learner acquired knowledge. Once the learning path is ready and fixed the best learning strategies for the learner the Training Agency calls the **Brokerage Office(3)** of the Diogene Virtual Organization in order to obtains the training offers to propose to the learner. Each Brokerage Office starting from the data received by Training Agency (learning path, learning strategies, learner model, etc..) will prepare a own training offer. To do this the Brokerage Office calls the **Publishing House(4)** and the **WebCatcher Agency(4)** asking them to provide the list (list entry are the metadata that describe the single Learning Object, included the price) of available material, that best fit the Training Agency requirement.

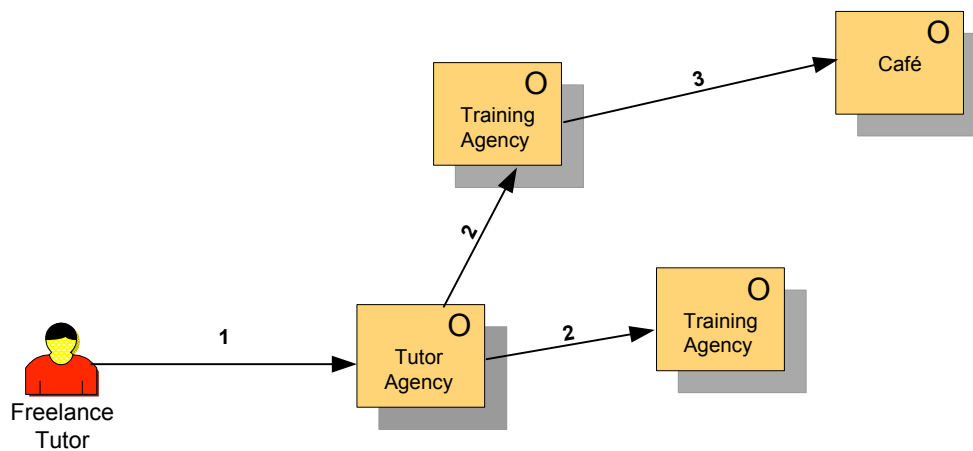


Figure 3 – Orchestrated scenario b)

In a second scenario, a **Freelance Tutor** accesses through **Tutor Agency (1)** to his/her jobs list and clicking on a specific job she/he can be redirected to a specific **Training Agency (2)**. The Training Agency offers a tutoring environment where the tutor can control learner activities, status, profile, results and progresses. Training Agency offers also an entry point for the **Café(3)** where tutor can meet his learners and collaborate and communicate with them.

2. Anatomy of the Abstract organisation

Organisations within the Diogene VO are seen as service providers. Therefore the VO is constituted by a set of cooperating services that give tangible results to the users through the available access points (web portals). Then Organizations live inside Internet and offer their services as Web-Services, they live inside a Web Server and are visible through a Web Portal.

2.1. Abstract Organisation Components

Each Organization included in Diogene VO has the same core infrastructure: the Diogene Abstract Organisation (DAO) shown in Figure 2. Main DAO elements are described below.

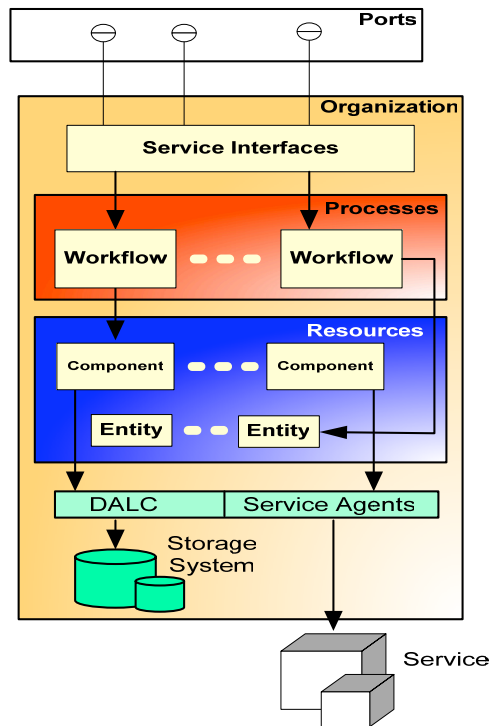


Figure 4 – Diogene Abstract Organisation.

- **Ports** are access points to particular services of an Organisation. Each service has to exhibit: a Notification Source Port (displaying an interface for those who want obtain a particular service) a Notification Sink Port (receiving the notification of the results of the required service) and a Notification Subscription Port (serving for the subscriber to manage in a source service the subscription to a particular service). Ports offer a general schema for publish and subscribe pattern.
- **Service interfaces.** To expose business logic as a service, it's necessary create service interfaces that support the communication contracts (message-based communication, formats, protocols, security, exceptions, and so on) that different consumers require. Service interfaces are sometimes referred to as business facades. Organizations will use Façade Pattern in order to collect services invocations and dispatch calls to processes at the second layer. The business façade object authenticates and authorizes. Then façade invokes a Workflow that performs the business. The Façade Pattern provides a unified interface to a set of interfaces in a subsystem. Façade defines a higher-level interface that makes the subsystem easier to use.
- **Workflows.** Workflows define and coordinate long-running, multi-step business processes. Workflows could be wrapped in specific components that can be managed by an Application Server like COM+ in order to handle transactions.
- **Components.** Regardless of whether a business process consists of a single step or an orchestrated workflow, your application will probably require components that implement business rules and perform business tasks.

Components implement the application business logic and can be managed in an Application Server like COM+ to handle object pooling.

- **Entities.** Applications require data to be exchanged between components. The business entities that are used internally in the application are usually data structures that wrap DataSets, DataReaders, or XML streams, but they can also be implemented using custom object-oriented classes that represent the real-world entities the application has to work with.
- **Service Agents.** When a Component needs to use functionality provided in an external service, it is necessary to provide some code to manage the semantics of communicating with that particular service. Service agents isolate the idiosyncrasies of calling diverse services and can provide additional services, such as basic mapping between the format of the data exposed by the service and the format the application requires.
- **DALC (Data Access Logic Components).** Most applications and services need to access a data store at some point during a business process. DALC Components provide a standard and abstract way for accessing to the storage systems.

2.2. Some Technological Details

The Diogene Virtual Organization lives into Internet environment. Organizations expose their services as Web Services described by the WSDL standard. Actually, we think that Web Services choice is the best one for distributed environment because they are based on standards like SOAP, HTTP (protocol) and XML (meta-language for message describing) that encourage interoperability. The DAO will be implemented using Microsoft .NET technologies. Ports are Web Methods exposed as Web Services. Workflows and Components they will live in COM+ Application Server in order to take advantage from enterprise services of this environment like the Transactions management, Object Pooling management, etc... For the running and tracing of the workflows we have also used Microsoft MSMQ. Entities are simple light weight Microsoft.NET components that we have modelled using ValueObject design pattern. They are used to transport simple or complex information through the different levels of the layered architecture. DALC builds a component layer that is used to abstract from communication with storage systems.

Authentication services have been developed ex novo and have been released as a Utility Component called Authentication Agency. Services provided by Authentication Agency are: User registration, User authentication and Validity check of Security Token. Security Token is a representation of security-related information (e.g. X.509 certificate, Kerberos tickets and authenticators, mobile device security tokens from SIM cards, username, etc.).

Security Token will travel across the Diogene Network (from a service to another one) embedded in a SOAP message. In order to achieve a consistent Token passing across the Diogene Network we will use WS-Security. WS-Security is a piece of GXA specifications that describes how to attach signature and encryption headers to SOAP messages. In addition, it describes how to attach security tokens, including binary security tokens such as X.509 certificates and Kerberos tickets, to messages.

For the implementation of Diogene Services Registry we used the known UDDI (Universal Description, Discovery and Integration protocol) technology that is one of the major building blocks required for successful Web services. UDDI creates a standard interoperable platform that enables companies and applications to quickly, easily, and dynamically find and use Web services over the Internet.

UDDI also allows operational registries to be maintained for different purposes in different contexts. The UDDI project takes advantage of World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF) standards such as Extensible Markup Language (XML), and HTTP and Domain Name System (DNS) protocols.

Additionally, cross platform programming features are addressed by adopting early versions of the proposed Simple Object Access Protocol (SOAP) known as XML Protocol messaging specifications found at the W3C Web site. The UDDI protocol is the building block that will enable businesses to quickly, easily and dynamically find and transact with one another using their preferred applications.

3. Migrating toward a Grid Infrastructure

Among the technologies enabling the VO, the Grid ones are taking a leading role. The Grid computing, thought as the evolution of metacomputing, addresses issues related to access provisioning, coordinated resource sharing and problem solving in dynamic, multi institutional VO [4]. This sharing capability and support for VO implementation has been one of the main key success factors of Grid as enabling technology for e-science infrastructure. The new research directions for the development of Grid technologies are moving towards a service oriented view by the definition of the Open Grid Service Architecture (OGSA) [5] that, joining Web Service and Grid technologies, defines an open and extensible framework for distributed and highly collaborative applications.

The Diogene model can become, in simple and feasible way, a Grid-Aware model. It presents several Grid features: it is strongly based on the use of open and standard based technologies and it adopts the VO model. Furthermore, it provides, as Utility Components, a single sign on authentication mechanism to access the VO services and a Service Registry. But it has some drawbacks that, using OGSA complaint technologies instead of the web service ones, can be easily addressed in order to enhance it.

OGSA is strongly based on the service orientation and virtualization, where the first is related to definition of a uniform exposed service semantic and the identification of protocols that can be used to invoke a particular interface, and the second is related to the encapsulation behind a common interface of diverse implementation, so everything (Resources, Learning Objects and so on) in this environment is a service.

Currently, Diogene doesn't take care of (Physical) Resource Management issues. The VOs are composed by heterogeneous and distributed resources and there must be a standard way in order to manage these remote resources. Local Organizations may use a wide variety of management mechanisms (schedulers, queuing systems, reservation systems, and control interfaces), but a client need to learn how to use only one mechanism to request and use these resources.

In a Diogene VO, the Organizations are seen as service providers but there is no agreement between a provider and a client on the service level guarantee. Organizations provide services "as they can", they collaborate between them but they use only the resources of the host where their instances run. Organizations have not a full vision of all the resources of the VO, so they are not able to promise a level of QoS. Moreover, the Diogene Services Registry is a single centralized component: it maintains only static information about the services and organizes services in categories. There isn't information about the physical resources and it hasn't mechanisms to store dynamic information related to the status of the VO resources (e.g. available memory, CPU usage and so on). The VOs are dynamic, resources in a VO

join and resign and their status change quickly; so it is necessary to enhance and distribute it, to avoid bottlenecks, the Diogene Service Registry.

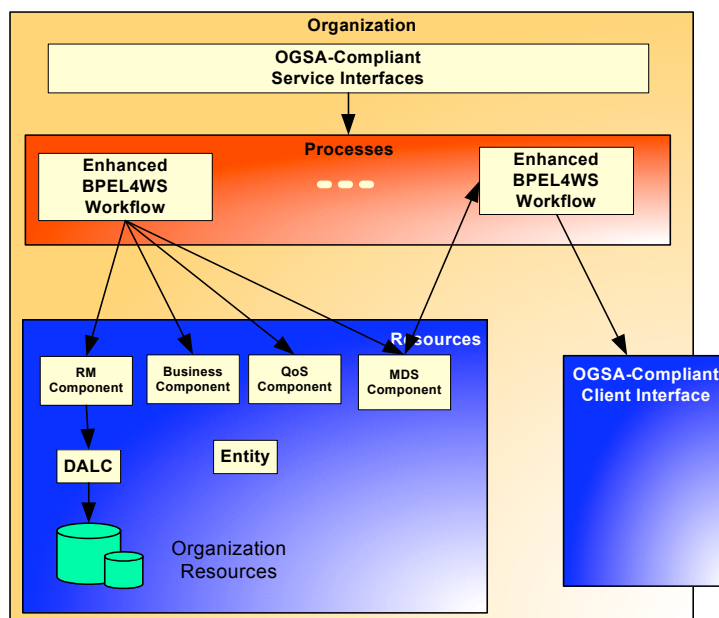


Figure 5 – DAO Grid-Aware.

In order to shift Diogene towards a Grid Service Oriented infrastructure, the Organizations should act as distributed Grid Service Providers (GSP). They should provide OGSA compliant services (e.g. Grid Services [6] or WS-Resource [7]) to clients. Organizations can share physical resources between them on the basis of VO business and security policies. They manage their own resources and they should be able to use and/or buy external resources from other Organizations.

Furthermore, the Utility Components become OGSA compliant services. They are accessible, in a standard way, from internal Organizations and external users. The Diogene Services Registry has to be modified to become an OGSA compliant Monitoring and Discovery Service (MDS) [8].

3.1. DAO Grid-Aware.

The figure 3 shows a Grid-Aware version of the Diogene Abstract Organisation. The Organizations provide OGSA compliant services. The service interface exposes the business logics and extends the notification pattern, as described in [6]. A DAO Grid-Aware does not need the Ports, as they are described in the section 3.1. Indeed, they are directly available through the OGSA Grid services.

The Organizations provide complex services obtained by composition and coordination of simple services. The Workflows, in this Grid-Aware model, are BPEL4WS [10] scripts enhanced with metadata in order to manage the Grid features (e.g. lifetime management) of the services to compose. Each organization has a BPEL enhanced engine able to parse the enhanced scripts and execute the workflow. This

engine is exposed as OGSA service and accessible, by the service interface, in a standard way.

Each Organization has a Component called “MDS Component”, which is functionally equivalent to a Grid Resource Information Service (GRIS) [4]. It maintains information about the service/s provided by the Organization and the resources of the Organization and has the capabilities to monitor the resources in order to obtain updated information about their status. In the Grid-Aware DAO Components, we find the Business Components that implement business rules and perform business tasks, a QoS Component that performs the mapping between high level figures of QoS (e.g. Time response) and low level figures (e.g. Required memory, Bandwidth) that can be monitored and measured and a Resource Manager (RM) Component that implements the logics to manage the local resources and exposes a standard interface.

The Organizations are GSPs but, by the other side, they take the role of clients when they negotiate or use services provided by other Organizations, so there is another interface (in figure 3 it is called “OGSA Compliant Client Interface”) that allows an Organization to interact with other members of the VO. The communication with external services happens with this standard interface and the Service Agent is not required.

3.2. A simple scenario

To clarify our ideas, we explain a simple and not exhaustive scenario. In order to realize a real business scenario, the VO needs new Utility Components, besides the ones described in the sections 2.3. The first is the Resource Manager Utility (RMU), which is able to manage VO resources (virtualized as services) on demand based on the VO policies. The other Utility Components are a Negotiator (N), which provides negotiation facilities between a client (maybe a user or another Organization) and a “VO MDS”. This last is equivalent to the Grid Index Information Service (GIIS) [4] and it maintains information about the services provided by the VO and information about the resources of the VO (e.g. status, access policies and so on) It interfaces itself with the “MDS Component” of the Organizations in order to obtain and index the information about resources and services.

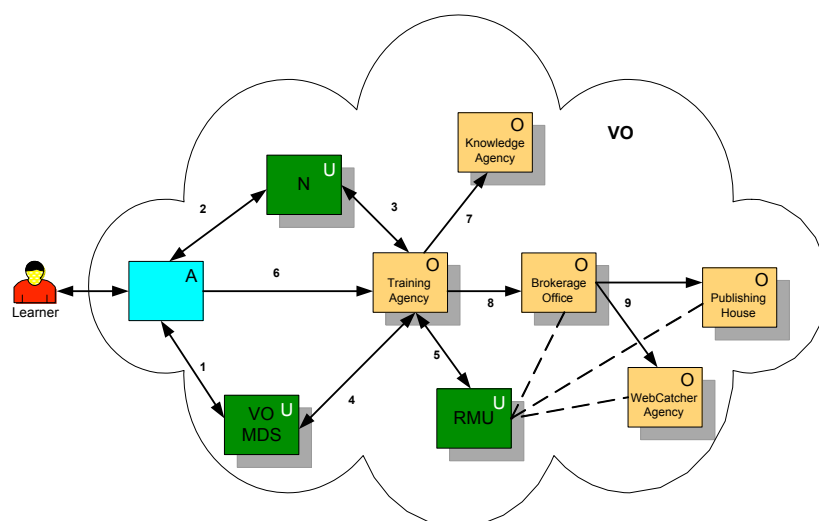


Figure 6 – Diogene VO.

Referring to Figure 4, a learner registered in the Diogene VO asks for a training session and, through the Accessor Component, performs a query against the VO MDS (1). He/she obtains a list of services provided by different Organizations inside the VO. He decides to gain experience in the business field and to play with a Business Game, but he/she provides some visualisation constrains (he/she asks for 3D representations of business game data) so he/she requires narrowing requirements on the visualisation and performance of the service. Through the Accessor Component, he/she negotiates with the Negotiator Utility (2) the QoS for the required service in a transparent way. The Negotiator interacts with TA Organization (3), which starts a simple workflow. The first operation is to map the required constrains with the resource requirements needed to satisfy it. The QoS Component performs this task. Once obtained the results, the second step is to query the VO MDS (4) to get a list of available resources and/or services of the VOs that fulfil the requirements, on the basis of the current status. Once the TA has the list of the available resources, it contacts the RMU of the VO (5). The RMU asks for the RM Components of the Organizations, which can provide the required resources, the resources reservation (in the figure, we suppose that all the necessary resources can be provided by the BO, WA and PH Organizations and this operation is shown with the dashed lines). At the end, a SLA is caught up and returned to the learner.

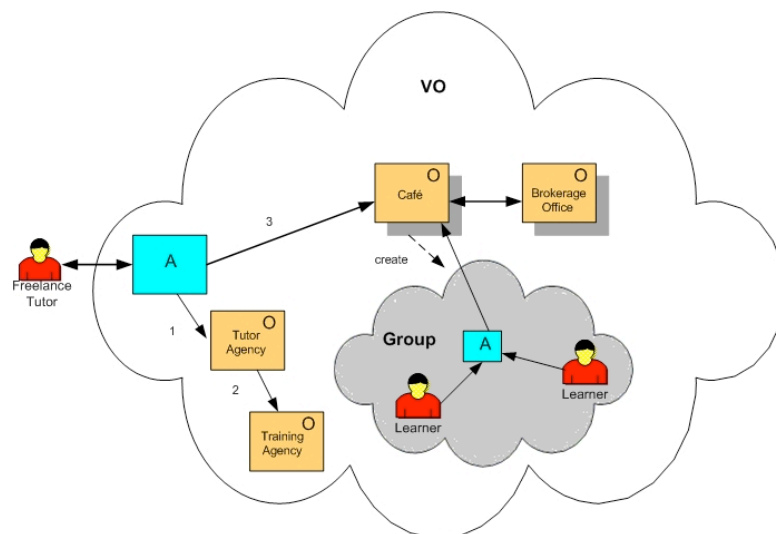


Figure 7 – Collaborative scenario.

The other steps, from (6) to (9) in figure 4, are equivalent to the first scenario presented in section 2.4 and the BO prepares and provides the business training offer for the learner based on the obtained SLA and other customised information. We suppose that the Diogene Bank Organization is able to perform a “pay per use” mechanism also on the resource usage. We haven’t discussed the Accounting/Billing and SLA Monitoring issues for simplicity.

Another more complex scenario is the following, which emphasizes the features of a collaborative and experiential based learning model. Referring to Figure 5, we have a freelance tutor who wants to offer support to a group of learners. The steps (1) and (2) are the same of the second scenario described in section 2.4. Furthermore, an added value of the Grid technologies is the capability to provide collaborative features creating a group and inviting some learners, sharing needs and skills, to join a collaborative learning session. Tutor can access the **Café** organization (3) providing some target concepts and rely upon the collaborative features of the organization to provide highly realistic virtual scientific experiments. The **Café** asks **Brokerage Office** to prepare and define, interacting with other organizations, a standard didactical model for the achievement and representation of such experiments. The learners belonging to the Group can access to the virtual experiment and interact with it, using a group Accessor Component that interacts with the **Café** organization.

In this type of model, a learner is immersed in a specific context which, through appropriate simulations, develops active learning processes with progressive abstraction levels, leading to the construction of knowledge in a dynamic way. In this learning model, the student can also receive the support of other users (collaborative aspects) and, through the comparison with them; they can build a new “mediated” knowledge.

4. Conclusions and Future works

In this paper we have presented a Service oriented VO for E-Learning. We have described the Diogene model given particular emphasis on the interactions between the different members of Diogene VO in order to provide training services. Furthermore, we have shown some ideas in order to deploy the Diogene model in an OGSA compliant infrastructure, emphasising the benefits that the Diogene architecture could have from the adoption of Grid technology and in particular OGSA.

Our future work will be mainly focused on two key aspects: The implementation of Grid-aware version of DAO and the evaluation of the deployment over the new Microsoft Operating System code named Longhorn. Concerning the implementation of the Grid-aware version of the DAO we plan to use the middleware under development in the GRASP – Grid based Application Service Provision project taking into account the new development in the Grid domain concerning the WSRF (Web Services Resource Framework). Indeed, Grid community, for the motivations described in [16], is refactoring the Open Grid Service Infrastructure [6] in terms of web services specifications and technologies. The WSRF is based on the concept of WS-Resource, which is a stateful resource that participates in the *implied resource pattern* [7]. The pattern is codified using existing web service technologies, as XML, WSDL and WS-Addressing.

On the other side we will investigate how to deploy the Grid-aware version of the Diogene DAO using the new Microsoft Operating System. Longhorn, represents the first operating system built with *managed code* (i.e. .NET code completely compiled into Intermediate Language to be executed inside a CLR). Longhorn presents three key innovations for Presentation Layer, Data Layer and Communications Layer. All these layers provide their functionalities in terms of Web Services.

For the Presentation Layer, Longhorn introduced *Avalon Subsystem* [13] representing a significant evolution of the presentation technology with the goal of allowing developers to easily build rich and compelling user interfaces that can seamlessly integrate high-quality document and multimedia content.

For Data Layer, *WinFS Subsystem* [12] has been introduced. *WinFS* is the name of the new file system that provides data and storage model for “Longhorn” and has been introduced in order to simplify the process of finding and storing user data.

Indigo Subsystem has been released to fill the Communications Layer. *Indigo* is a set of technologies for developing connected applications on the Windows platform. It provides a complete and flexible messaging platform independent of network topology. With *Indigo*, developers can write applications on a simple yet powerful programming framework.

Furthermore, Longhorn exposes a set of comprehensive API, called *WinFX*, for writing Longhorn-based applications. *WinFX* APIs are entirely available in managed code.

In this frame our effort will be devoted to investigate how to extend Longhorn layers and in particular WinFS and Indigo in order to integrate the WSRF specifications. Both Indigo and WinFS provide a broad array of distributed systems capabilities in a composable and extensible architecture, spanning from data access, indexing, search and retrieval to transports, security systems, messaging patterns, encodings, network topologies and hosting models. In this way a developer can rely on an OGSA compliant infrastructure “embedded” in the operating system.

References

- [1] Web Site “Diogene: a Training Web Broker for ICT Professionals” FP5 Project (IST-2001-33358). <http://www.diogene.org>.
- [2] N. Capuano, M. Gaeta, L. Pappacena. An e-Learning Platform for SME Manager Upgrade and its Evolution Toward a Distributed Training Environment. Proceedings of the 2nd International LeGE-WG Workshop "e-Learning and Grid Technologies: a fundamental challenge for Europe" Paris, France, 2003.
- [3] Richard Hubert. Convergent Architecture, Building Model-Driven J2EE Systems with UML. John Wiley & Sons.
- [4] Foster, C. Kesselman and S. Tuecke, “The Anatomy Of The Grid” 2001
- [5] Foster, C. Kesselman, J. Nick and S. Tuecke, “The Physiology Of The Grid” 2002
- [6] Foster, C. Kesselman, S. Tuecke et al. “Open Grid Service Infrastructure V1.0” 2003
- [7] Foster, Frey, Tuecke et al. “Modelling Stateful Resources With Web Services” 2004
- [8] <http://www.globus.org/mds/>
- [9] D. De Roure, N. R. Jennings, N. R. Shadbolt “The Semantic Grid: A Future e-Science Infrastructure” 2002
- [10] <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- [11] <http://msdn.microsoft.com/longhorn/understanding/pillars/indigo/default.aspx>
- [12] <http://www.msdn.microsoft.com/Longhorn/understanding/pillars/WinFS/default.aspx>
- [13] <http://www.msdn.microsoft.com/Longhorn/understanding/pillars/avalon/default.aspx>
- [14] <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-addressing.asp>
- [15] <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwssecur/html/securitywhitepaper.asp>
- [16] Foster, Snelling, Tuecke et al. “From Open Grid Service Infrastructure to WS-Resource Framework: Refactoring & Evolution” 2004