Enabling The Construction Virtual Enterprise: The Osmos Approach

SUBMITTED: July 2001
REVISED: October 2001
PUBLISHED: December 2001 at http://itcon.org/2001/8/

Ian Wilson, Research Fellow,
Information Systems Institute, University of Salford, UK;
email: i.e.wilson@salford.ac.uk http://www.isi.salford.ac.uk/staff/iw/

Simon Harvey, Research Fellow,
Information Systems Institute, University of Salford, UK;
email: s.harvey@salford.ac.uk http://www.isi.salford.ac.uk/staff/sh/

Rémi Vankeisbelck, Research Engineer,
CSTB - Centre Scientifique et Technique du Bâtiment, Sophia Antipolis, France;
email: r.vankeisbelck@cstb.fr http://cic.cstb.fr/ilc/people/rv/rv.htm

Abdul Samad (Sami) Kazi, Research Scientist,
VTT – Technical Research Centre of Finland;
email: Sami.Kazi@vtt.fi http://cic.vtt.fi/kazi

SUMMARY*: This paper gives a comprehensive overview of the findings and results from the OSMOS (IST-1999-10491) project. OSMOS aims to highlight and go some way to meeting the needs of the industry by providing a set of tools, models, APIs and techniques to support the construction "Virtual Enterprise" (VE). Key to the OSMOS approach is that the tools will allow companies (especially SMEs) to partake in a project-based VE quickly and at a low entry-level. Through a combination of IDEF0 and UML modelling, within an iterative and incremental project methodology, the OSMOS consortium has elaborated a generic process model for the set-up and structuring of the construction Virtual Enterprise, which has formed the basis for the technical implementation of the tools and API. The tools, once designed, built, and made available for testing, have been evaluated within construction based case scenarios by end-users, and subsequently refined and re-tested. The resultant solution being offered through the OSMOS approach will involve some potential process changes within the companies wishing to take part in the VE, and the project aims to provide a proposed migration path to this end.*

KEYWORDS*: construction, virtual enterprise, API, Internet- based services, information modelling.*

1. INTRODUCTION

The convergence and widespread acceptance and use of information and communication technologies (ICT) have engendered, amongst many others, the notion of "Virtual Enterprise" (VE) in many industry sectors. Numerous definitions of the term (including 'virtual organisation' and 'virtual corporation') exist in the literature, for example Goranson, 1999, Tapscott, 1996, Dutton, 1999. This paper does not intend to offer a new definition, but it is important to note that generally the VE is defined by the organisations and groups involved, and characterised by their geographical dispersion and supported through the use of ICT. The authors believe, however, that *agreements* made between the relevant actors are also a key feature. Such agreements will include both the extent to which information and knowledge are managed and shared and the tools made available to do this, and also the degree of control employed. Rather than the VE being a new organisational form arising from the capabilities of advances in ICT and globalisation, it has been noted elsewhere (Rezgui *et al*, 2000) that the construction industry has for decades adopted the *modus operandi* of the VE. It is characterised by non-collocated teams of separate firms who come together for a specific project and may then never work together again.

This paper presents an overview of the findings and results to date of the ongoing Open System for Inter-enterprise Information Management in Dynamic Virtual Environments (OSMOS) (IST-1999-10491) project. OSMOS aims to highlight and go some way to meeting the needs of the industry by providing a set of tools, models, APIs and techniques to both support and enable the construction VE. This is being achieved via the specification of Internet-based services providing interconnection through semantic cross-referencing of objects held in different applications, coupled with an efficient VE management set-up.

1.1. Construction industry context

Within its framework of geographical dispersion the authors believe that the construction industry is characterised by various challenges in terms of working practices and solutions. These include the following:

- Fragmentation.
- No dominant actor to enforce ICT solutions.
- Information exchange within any construction project is mainly between others than the client and is not, therefore, contractually controlled.
- All actors are involved in numerous VEs at the same time.
- The industry is project oriented: this influences the incentives, accounting, etc. Any ICT must be deployable and profitable within one project to all/several partners.
- Temporary and often short-term business relationships: VE partners may never work together again.

Organisations and individuals participating in construction teams bring their own unique skills, knowledge and resources, which include proprietary and commercial software applications. The ICT solutions employed on construction projects tend to be fixed rather than open, and frequently lack support. They are often prohibitively expensive particularly for small to medium sized enterprises (SME), and offer only limited growth paths in terms of hardware and software. Furthermore there is often a requirement to organise the enterprise around the adopted technological solution. FIG. 1 shows the construction industry context, with various teams working 'virtually' around the construction project.

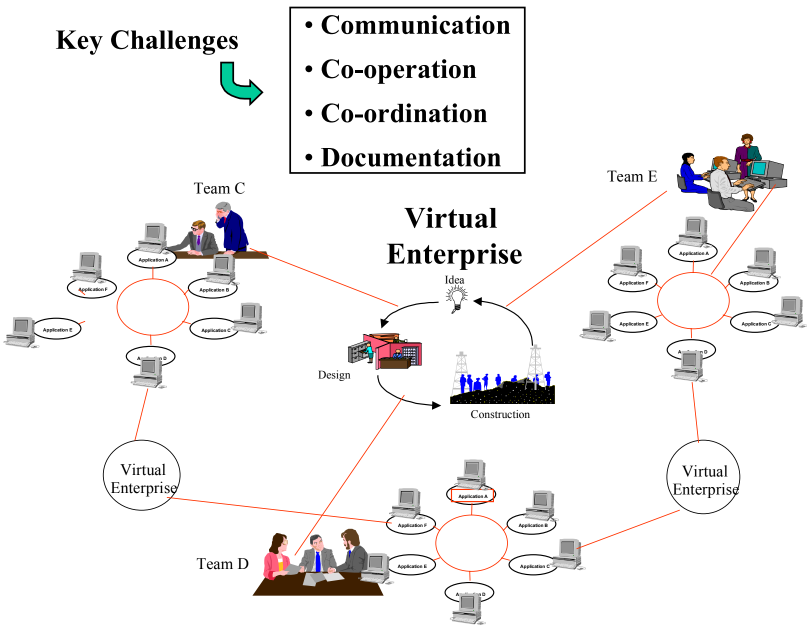


FIG. 1: Construction Industry Context

Due to these characteristics and the context depicted above, various problems have been identified including the following:

- ICT support to handle fragmentation imposed by the very nature of the industry in terms of communication and information exchange still needs improving.
- Interactions between actors are still not well co-ordinated, especially because of the inherent dynamic business relationships taking place in the construction industry.
- ICT support for information and document management varies from one company to another, but overall is still done in a traditional and ad hoc way.
- Project documents present a great deal of redundancy and often lack structuring.

1.2. OSMOS objectives

The overall aim of the OSMOS project is to enhance the capabilities of construction enterprises, including SMEs, to act and collaborate effectively on projects by setting up and promoting value-added Internet-based flexible services that support teamwork in the dynamic networks of the (European) construction industry. This translates into the Scientific and Technological measurable objectives described below:

1) Specify Internet-based services for collaboration between dissimilar construction applications and semantic cross-referencing between the information they manipulate.
2) Specify Internet-based services allowing the co-ordination of interactions between individuals and teams in a dynamic construction VE.
3) Specify a model-based environment where the release of, and access to, any shared information (including documents) produced by actors participating in projects is secure, tracked, and managed transparently (in real time whenever possible, otherwise asynchronously).
4) Provide low entry-level tools (cheap and user-friendly) to small enterprises to act and participate in construction VEs.
5) Allow end-users to use their proprietary and commercial applications on projects, by implementing the services specified in objective 1 (e.g. via plug-ins), and allow them to transparently participate to collaborative work in dynamic VEs.
6) Implement the model-based environment, specified in Objective 3, providing a distributed information management support for the VE.
7) Set up two OSMOS Internet-based teamwork service providers for the purpose of the project, and ensure their take-up, as commercial offers, after the completion of the project.
8) Define the migration path to using the OSMOS approach.
9) Analyse the likely benefits of adopting the OSMOS approach.

Key to the OSMOS approach is that the tools will allow companies (especially SMEs) to partake in a project-based VE quickly and at a low entry-level.

1.3. Methodology

The OSMOS consortium (see Appendix – section 10) has adopted an iterative and incremental approach to address the objectives of the project. The work is being carried out across three iterations spanning a 27-month period. The workpackages within each iteration of the project are interdependent and provide feedback in a cyclical manner. This project methodology allows continual assessment and validation of the infrastructure and models, and addresses the potential risks in relation to the implementation of the proposed solutions.

Requirement for the proposed system was provided by an analysis of the current business processes and information management practices (both intra-company and inter-company) within the end-user organisations (Derbi, Granlund and JM, in France, Finland and Sweden respectively). An analysis was also made of their currently used software applications. The process analyses led initially to the development of models using IDEF0 functional modelling (NIST, 1993) describing the basic processes taking place in a construction VE. By abstracting from these models a Generic VE Process Model (GVEPM) was designed to determine the high-level process activities. At a lower level the Unified Modelling Language (UML) (Object Management Group, 1999) was employed to detail (via Use Cases) the ways in which the OSMOS system can be used at a business level, and to derive the required functionality of the system. The ensuing Use Cases were the bridging link between the requirement capture and the system specification (FIG. 2).

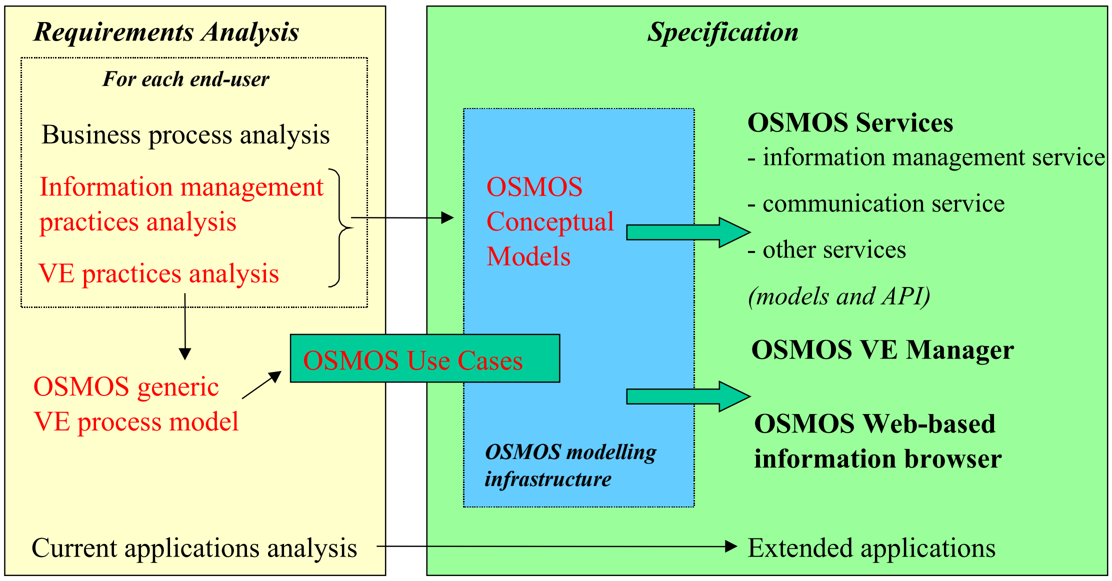

Requirements Analysis
For each end-user
Business process analysis
Information management practices analysis
VE practices analysis
OSMOS generic VE process model
OSMOS Use Cases
Current applications analysis
Specification
OSMOS Conceptual Models
OSMOS modelling infrastructure
OSMOS Services
- information management service
- communication service
- other services
(models and API)
OSMOS VE Manager
OSMOS Web-based information browser
Extended applications

FIG. 2: Methodological link between analysis and specification

With the Use Cases defined, sequence diagrams and class diagrams were employed to specify the internal system design. In parallel to this, and from the analysis of construction common software applications, the set of interfaces to be implemented between the various software applications and the OSMOS architecture could also be defined, resulting in first the definition of and then after agreement, development of the OSMOS API.

The principal means of testing and evaluating the OSMOS approach is being provided through field trials simulating work in a construction VE. Criteria have been determined for technical, social and economical evaluation of the OSMOS system together with legal, contractual and organisational aspects.

1.4. Structure of the paper

The following section provides an overview of the analyses carried out within the requirement capture phase of the project. This is followed by detailed descriptions and illustrations of the elements involved in the specification of the OSMOS platform. The next section then focuses on the implementation of the OSMOS platform. Initial results from the testing and validation of the approach are then presented, before a brief section outlining the future direction for the project.

2. REQUIREMENT CAPTURE FOR THE CONSTRUCTION VE

In terms of the migration from initial requirements capture to internal design and API development of the OSMOS system, the methodology included the following elements:

- Elaboration of core business-oriented process models
- Definition of process models describing team working on projects in organisations
- Generalisation of, and abstraction from the above, leading to the development of a proposed OSMOS Generic Virtual Enterprise Process Model (GVEPM)
- Architecture specification

This section focuses on the capture of the process models and the results found from this effort.

2.1. Business process and information practices analysis

The information input to the process models was provided by the end-user organisations, based on their current working methods in the construction industry, and detailed in IDEF0 format. It is worth mentioning that the consortium was aware of the limitations of IDEF0 alone for process description, and for this reason an OSMOS standardised format was created to provide capture of further information related to each process/functional activity. This format allowed the end-users to present their current processes as IDEF0 diagrams, and to include additional information as required. This information included a description of the activity being modelled and the operational context in which it is applied. Additionally, the actors involved in the activity, any existing pre-conditions and/or post-conditions, exceptions, and other remarks pertinent to the activity could also be provided. The resultant models, which are presented elsewhere (The OSMOS Consortium, 2001*a* appendices) provided a comprehensive view of the intra-company business activities and the methods of information handling between actors. This formed the first step towards the specification of the OSMOS generic solution.

2.2. Analysis of interactions between teams on projects

The next phase of the requirement capture was an analysis focused on the current management of teams and other actors in the context of a VE. This analysis provided a comprehensive view of the inter-company interactions of the actors commonly involved in a construction project. The results of the analysis indicated the many variables to be taken into account during the life of a VE. It became clear that a VE in the construction industry is contemporaneous with the lifecycle of any specific building project. Depending on the actors involved therefore, a single VE may for example exist for the complete lifecycle of a building, whilst others may exist only during the design phase or facilities management (FM) phase. Each of these examples would require different infrastructures, available services, information management practices, and so on. The analyses showed that a generic VE solution, therefore, had to take into account at least the following variables:

- Required infrastructure for the project
- A methodology to agree on procedures and protocols
- Contractual agreement
- Available services according to the contract
- Setting up of a VE administrator account
- Structure of information and information entities
- Training of personnel (e.g. project administrator, personnel using services provided by third parties, etc)
- Management of changes – including actors, classes, access rights, information, infrastructure and configuration data, rapid change in technology, the building itself, etc.
- Data security
- Transfer of accumulated data

The models produced are presented in full elsewhere (The OSMOS Consortium, 2001*b*). This analysis highlighted the interactions and processes that were common to all of the end-user companies, and also those that were specific to each. The combined results of the analyses so far described, provided the basis for the GVEPM. Before presenting the GVEPM, however, it is important to note that the analyses also identified three potential "roles" within the OSMOS model. These are presented below.

2.3. Perceived roles in the OSMOS approach

It was recognised that three distinct roles would interact to both enable and make use of OSMOS in a typical VE setting. The three roles (FIG. 3) are:

- *OSMOS Service Provider (Role A):* The companies adopting this role are primarily concerned with hosting the OSMOS core infrastructure through provision of and access to both OSMOS core services and third party services (TPS) (see section 4.1.1 below). Role A, through the OSMOS core, has the capability to host multiple VE projects and to make available different services (both core and TPS) to different projects.

- *OSMOS Third Party Service Providers (Role B):* These companies plug-in their services and register associated methods through a Role A provider and make them available for use in a VE. Typically, these services would be geared to serving a particular purpose for the VE to which they are being made available. Examples of these services include HVAC, facilities management, document management, CAD services, etc.
- *OSMOS Clients (Role C):* These companies use, and take part in VEs that are supported and enabled through the OSMOS platform. While one company would configure and administer the VE, others would make use of the core and TPS services made available to the project.

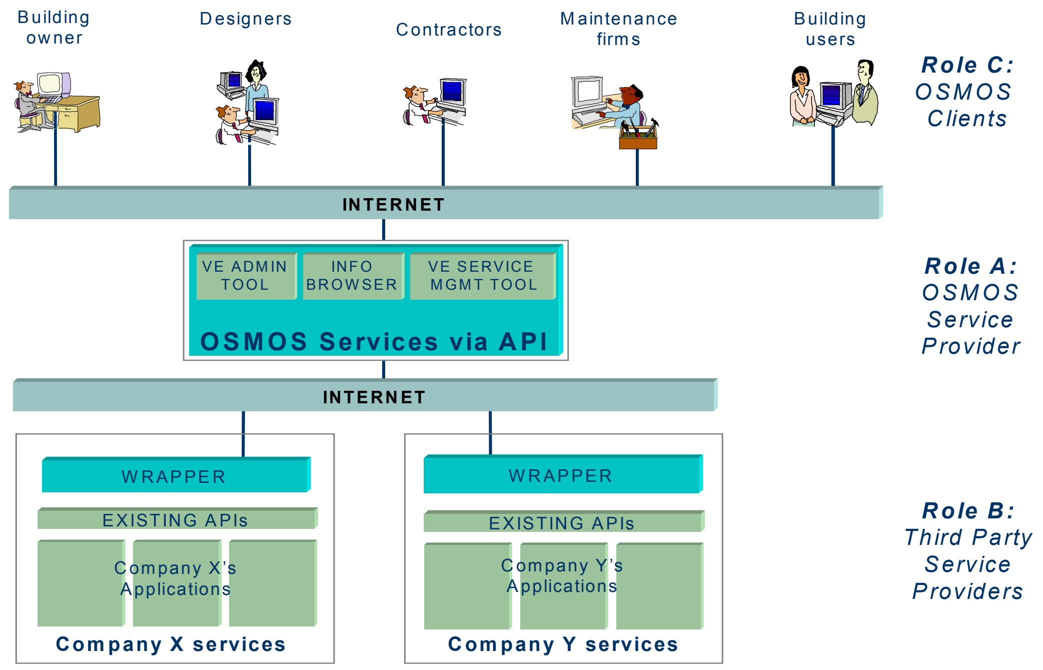


FIG. 3: Conceptual view of the interacting roles in an OSMOS enabled VE

In brief, therefore, Role A provides the OSMOS platform and core services, Role B provides some specific TPS applications, and Role C makes use of the OSMOS core and TPS to enable inter-enterprise information exchange in the VE(s) in which it is participating. It must be noted that though the three roles are mutually exclusive they may not necessarily be exclusive to a specific company. For example a company acting in Role A may also provide one or more TPS, therefore taking on both Role A and Role B.

2.4. The OSMOS Generic Virtual Enterprise Process Model (GVEPM)

Having recognised and agreed upon the perceived roles within the OSMOS approach, the GVEPM was developed by abstracting from the preceding analyses. The resulting OSMOS GVEPM revealed the high level generic processes that are required to enable the VE. It then provided the basis for the necessary Use Cases required to describe how the OSMOS system can be used at a business level, and to derive the required functionality of the system.

At the highest level the model represents all the actions required to *Manage and Use the OSMOS* platform to run a complete VE Project from initial client requirements to the end of the contract. The overall inputs were found to be *Requirements* (including Project Requirements, Client Requirements, Legal Requirements and Industry Requirements as appropriate) and *TPS*. A *product* (or service) and accompanying *information* in varying formats form the outputs. The *legal environment* operating at the time and *current market forces* control the activity, and the *VE Service Provider* using the *OSMOS Tools* performs the activity. Due to space limitations it is not possible to present all of the IDEF0 diagrams here (see The OSMOS Consortium, 2001*b*), but two examples are included below to illustrate the discussion.

The management and use of the OSMOS platform was found to decompose to two key activities: *Provide and maintain VE Services* and *Provide and maintain VE Project* (FIG. 4). The two activities at this level represent distinct processes in the VE.

Provide and Maintain VE Services incorporates the processes required to provide and maintain all of the services that are currently available to companies that wish to run a VE through the OSMOS platform. This activity is, therefore, equivalent to the OSMOS Role A. The main input to the activity is available TPSs that may be provided to a VE project as required. Provision has, therefore, to be made to provide and register the availability of such services, maintain them once provided and remove or replace them as changes in technology and/or requirements dictate.

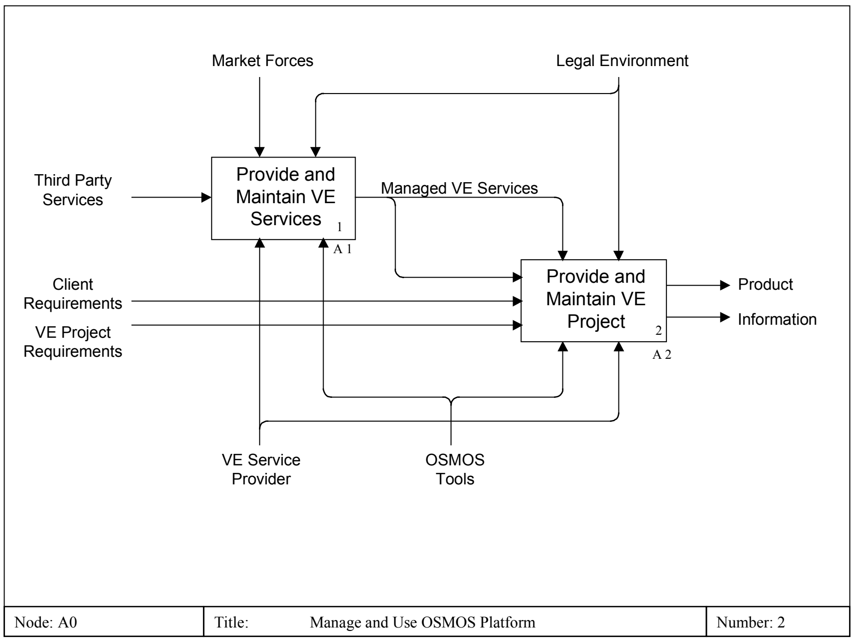


FIG. 4: Manage and Use OSMOS Platform

Provide and Maintain VE Project in contrast is the totality of processes required to run any individual project. It was also recognised that a VE customer may wish to run more than one project concurrently and capability for this had to be built in to the OSMOS models and consequently the tools. A VE project can only be enabled through the OSMOS platform once available managed VE services are in place as shown by the output from *Provide and Maintain VE Services* forming both the input and control for *Provide and Maintain VE Project* in FIG. 4. The necessary inputs to the activity include the requirements of the specific project client and the unique configuration of requirements for the project itself. A project management committee would be formed and a contractual agreement made between the actors (including the OSMOS Role A company) involved in the prospective VE. The contractual agreement and the VE project management committee would then control the processes required to set up and configure the particular VE project environment and operate the project from inception to completion.

Prior to the launch of a VE project the OSMOS environment would be configured to make available the particular services required by the project actors according to the specific project requirements. A set of management protocols and procedures would be agreed allowing the definition and assignation of project roles and access rights.

From this framework the consortium recognised the need to develop two distinct administration tools – an OSMOS VE Service Administration Tool, and an OSMOS VE Project Administration Tool. In order to keep the OSMOS solution as generic as possible, great consideration was given to the complete concept of actors, their roles and access rights. The resultant model was formulated taking into account the need for multiple project support and with legal and contractual issues in mind. Any one project (and therefore VE) comprises a set of actors at the organisation/company level, a unique set of individuals, required services, information objects, and varying degrees of access to services and information depending on legal, contractual and intellectual property rights (IPR) considerations. FIG. 5 shows the required high-level activities recognised in this regard in the process of operating a VE project.

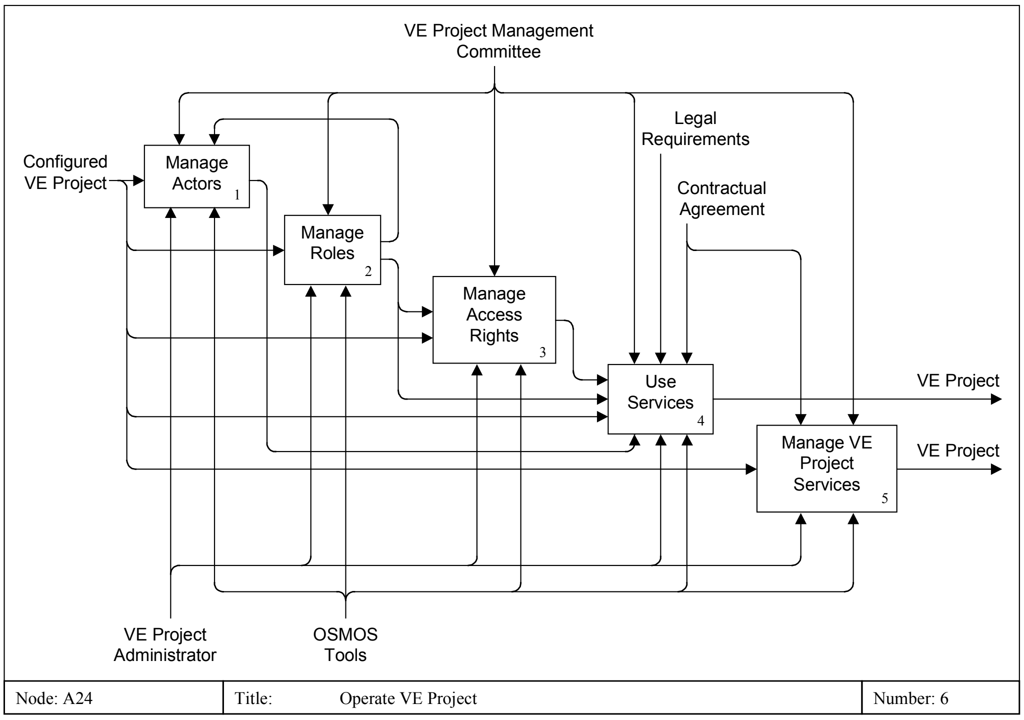


FIG. 5: Operate a VE Project

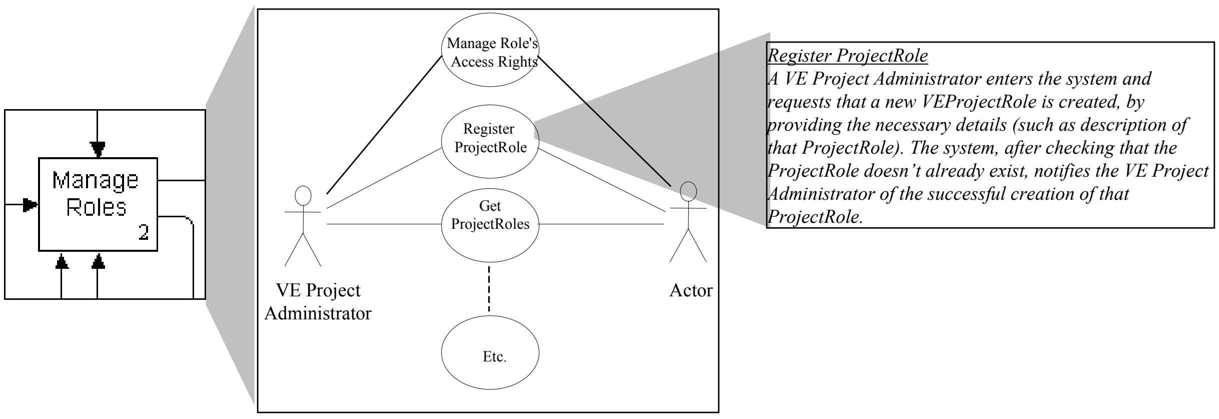
One or more individuals within one or more companies/organisations will hold a specific project role. It is through the project role that access rights will be given to an individual within the project. A project role in one VE, however, may not necessarily have the same profile in another VE, even where the actors for the two VEs may be exactly the same (due to the fact that the agreements, protocols and procedures will differ). The result is that through the OSMOS approach access rights are assigned to specific defined roles, which themselves are assigned to actors at the company/organisation level within the VE. The company/organisation then delegates the available role to the individual(s) as required.

3. SPECIFICATION OF THE OSMOS PLATFORM

3.1. Definition of the Use Cases

With the GVEPM defined and accepted, each of the nodes at the lowest levels of the IDEF0 model were further decomposed and described as a set of Use Cases, i.e. a set of textual descriptions at the *business* level, of how the OSMOS platform would be used. Furthermore some of the resultant Use Cases were decomposed into smaller Use Cases if they were found to be too complex. A large number of Use Cases were defined, and it is not possible to present these in full within this paper. However an example is provided here, the registration of a Role on a Project (from the IDEF0 node A242 *Manage Roles* illustrated in FIG. 5 above), to give the reader an appreciation of the process that was undertaken within the OSMOS Project.

FIG. 6 shows how the *Manage Roles* process was broken down into a number of individual Use Cases. Each Use Case was then described in a textual format that stated who performed it and what the performer (actor) expected the OSMOS System to do. The example given is for the *Register ProjectRole* Use Case. A complete formal description of each of the Use Cases and their corresponding textual descriptions is provided in Marache *et al.,* 2001.

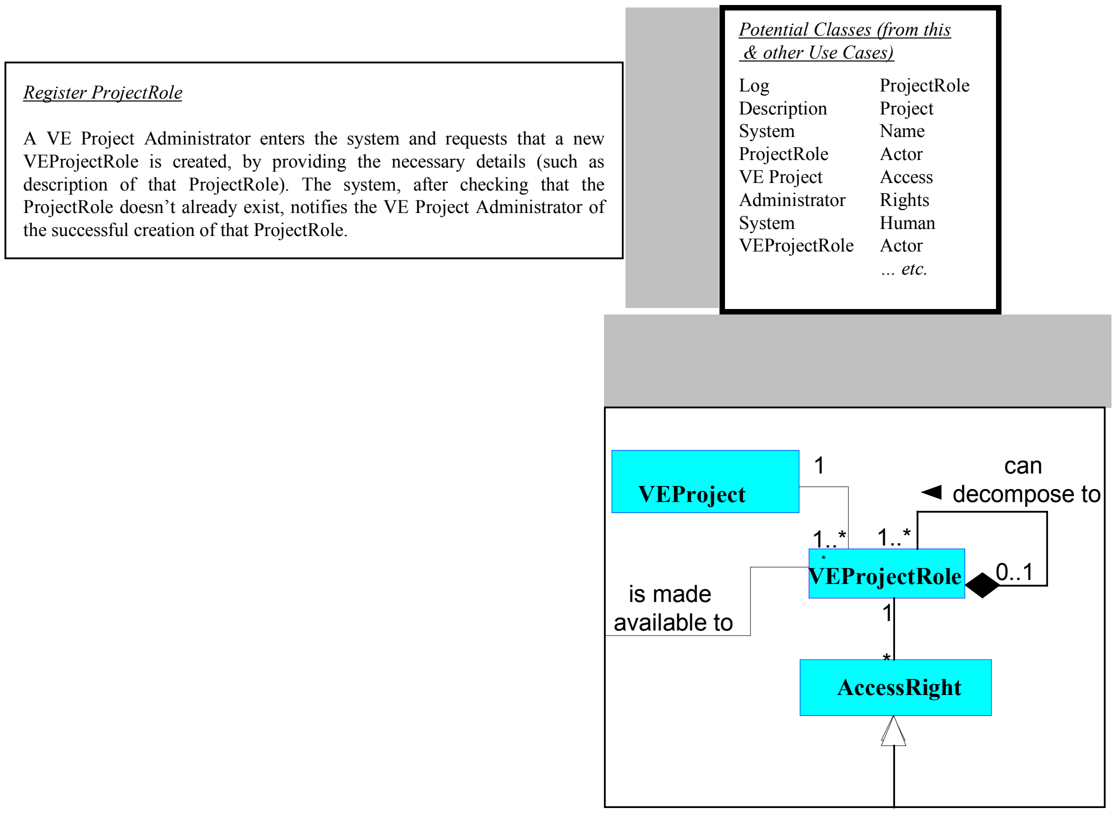


Register ProjectRole
A VE Project Administrator enters the system and requests that a new VEProjectRole is created, by providing the necessary details (such as description of that ProjectRole). The system, after checking that the ProjectRole doesn't already exist, notifies the VE Project Administrator of the successful creation of that ProjectRole.

FIG. 6: From process models to Use Case descriptions

3.2. From Use Cases to System Design

From the point that the Use Cases were defined, the OSMOS Project used an Object-Oriented approach to system design, which itself is a subset of the Rational Unified Process (Rational Corporation, 2001), using the UML (Object Management Group, 1999). This approach is widely known and accepted within the Information Systems community, and not described in detail here (see Barker, 2000, for an overview and Bennett, McRobb and Farmer, 1999, for a detailed exploration of the design process). However, a brief overview of the process will be given before the authors present the results of the design phase as a set of conceptual models.



Register ProjectRole

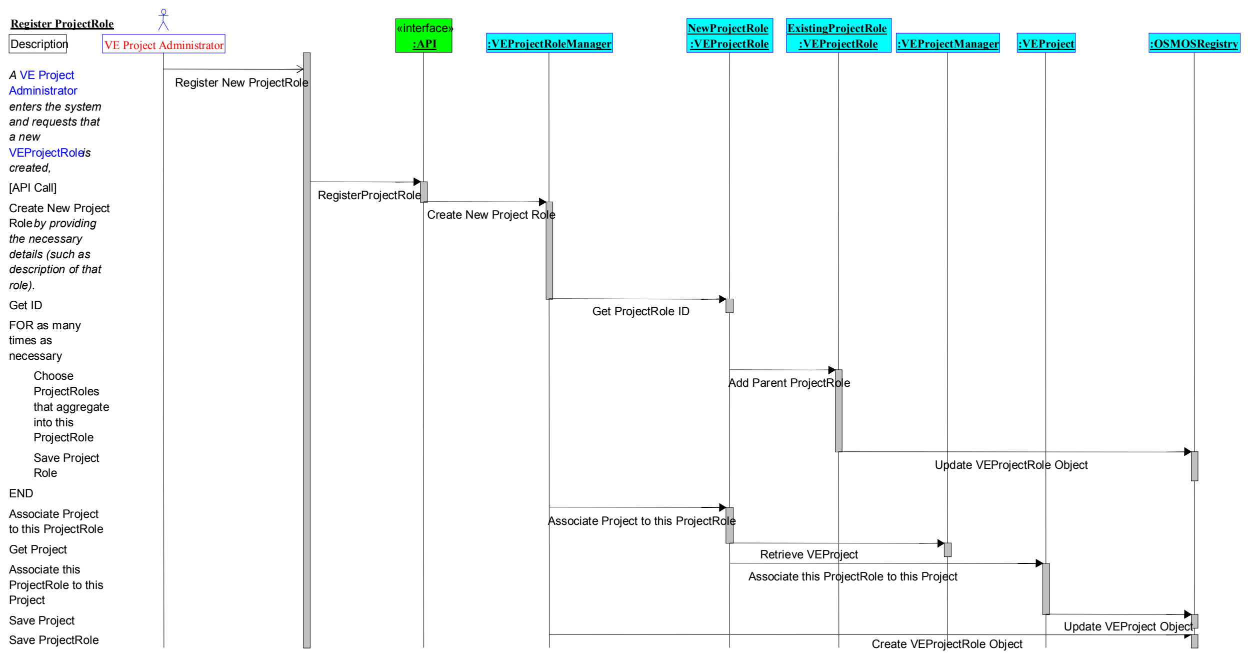
A VE Project Administrator enters the system and requests that a new VEProjectRole is created, by providing the necessary details (such as description of that ProjectRole). The system, after checking that the ProjectRole doesn't already exist, notifies the VE Project Administrator of the successful creation of that ProjectRole.

Potential Classes (from this & other Use Cases)

Log	ProjectRole
Description	Project
System	Name
ProjectRole	Actor
VE Project	Access
Administrator	Rights
System	Human
VEProjectRole	Actor
	... etc.

FIG. 7: From Use Case descriptions to Conceptual Models

Firstly, as shown in FIG. 7, the textual descriptions of the Use Cases were examined through a process known as Noun Phrase Analysis. The nouns were extracted from such textual descriptions, grouped together and arranged into a list of potential classes that could be used within the System. This list was then analysed to remove repeating or similar names, potential attributes and actual Actors (human users or external systems), that left a list of classes of objects that would be required by the OSMOS System. The resulting classes were then used in the conceptual modelling and design of the System. Relationships between such classes were examined and discussed based upon the Use Cases, which helped to define the attributes and relationships between classes of objects in the design of the OSMOS System.



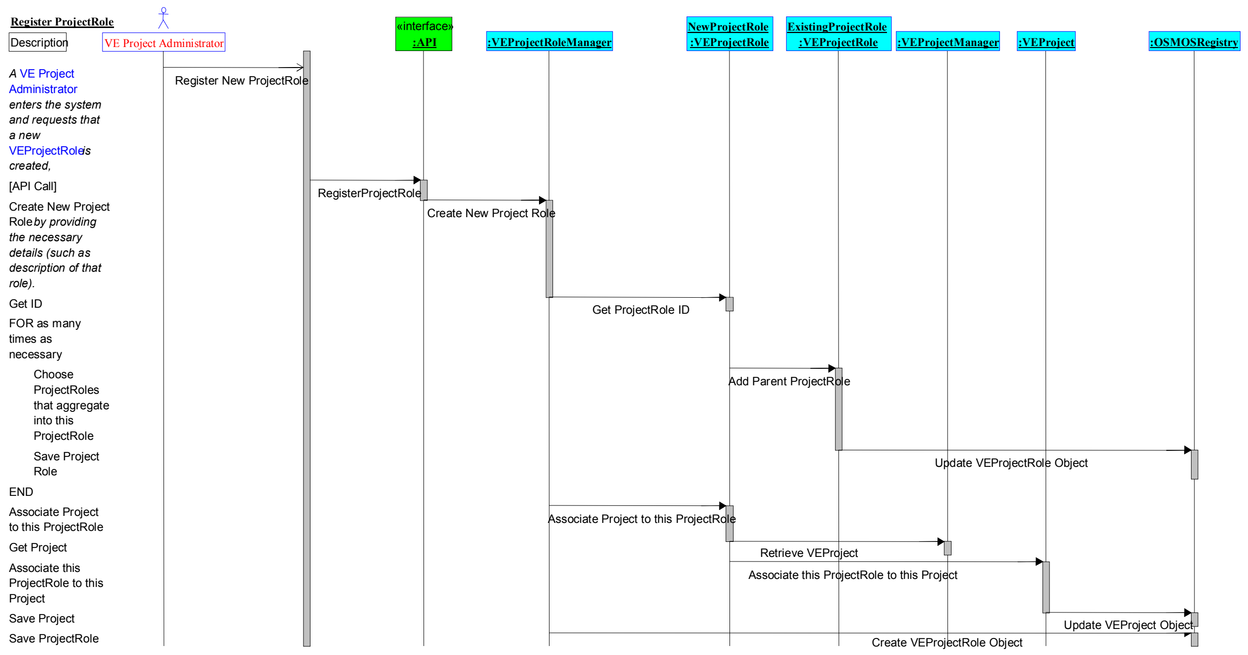
Register ProjectRole
Description
VE Project Administrator
«interface» :API
:VEProjectRoleManager
NewProjectRole :VEProjectRole
ExistingProjectRole :VEProjectRole
:VEProjectManager
:VEProject
:OSMOSRegistry
A VE Project Administrator enters the system and requests that a new VEProjectRole is created,
[API Call]
Create New Project Role by providing the necessary details (such as description of that role).
Get ID
FOR as many times as necessary
Choose ProjectRoles that aggregate into this ProjectRole
Save Project Role
END
Associate Project to this ProjectRole
Get Project
Associate this ProjectRole to this Project
Save Project
Save ProjectRole
Register New ProjectRole
RegisterProjectRole
Create New Project Role
Get ProjectRole ID
Add Parent ProjectRole
Update VEProjectRole Object
Associate Project to this ProjectRole
Retrieve VEProject
Associate this ProjectRole to this Project
Update VEProject Object
Create VEProjectRole Object

FIG. 8: From Use Case descriptions to technical design

In parallel to this effort, each Use Case was also analysed to decide how the classes and objects discovered above would interact within the OSMOS System, to satisfy the expectations of the calling Actor. The results of such analysis were presented as a set of UML Sequence Diagrams, one per Use Case, such as the example given in

Indeed, the designers discovered that the approach of producing these models in parallel with the Sequence Diagrams raised a number of questions within the Sequence Diagrams that were answered by working on the Conceptual Models, and vice versa. Consequently this process was repeated a number of times, using an incremental and iterative process, which gave enormous benefits such as increased stability, validity and levels of synergy within the presented results. As the results of the design phase run into a few hundred pages, the authors present only the outputs from one task: the OSMOS Conceptual Models and the OSMOS API.

3.3. The OSMOS Conceptual Models



Register ProjectRole
Description
VE Project Administrator
«interface» :API
:VEProjectRoleManager
NewProjectRole :VEProjectRole
ExistingProjectRole :VEProjectRole
:VEProjectManager
:VEProject
:OSMOSRegistry
A VE Project Administrator enters the system and requests that a new VEProjectRole is created,
[API Call]
Create New Project Role by providing the necessary details (such as description of that role).
Get ID
FOR as many times as necessary
Choose ProjectRoles that aggregate into this ProjectRole
Save Project Role
END
Associate Project to this ProjectRole
Get Project
Associate this ProjectRole to this Project
Save Project
Save ProjectRole
Register New ProjectRole
RegisterProjectRole
Create New Project Role
Get ProjectRole ID
Add Parent ProjectRole
Update VEProjectRole Object
Associate Project to this ProjectRole
Retrieve VEProject
Associate this ProjectRole to this Project
Update VEProject Object
Create VEProjectRole Object

FIG. 8. These diagrams helped the designers to decide the *methods* or *services* provided by such objects. In addition, careful analysis of such diagrams helped the designers to define the OSMOS API by thinking about the architecture of the OSMOS System at this stage, as well as further classes that are required for purely technical reasons (such as a *Database* object that may be required for persistent storage of other objects in the system).One of the major outputs of the OSMOS Project is a robust set of models that show the underlying philosophies and business logic that would be enforced by the implemented solution within the context of a VE. The models, which have evolved from the analysis and design phases discussed above, have also been influenced by the careful analysis and simplification of relevant parts of the models produced in other projects such as COMMIT (Brown *et al.,* 1996) and CONDOR (Rezgui and Cooper, 1998).

The models that have been defined at the time of writing are a set of models concentrating on the *High Level Security Service* and *Semantic Information Management.* In addition, models are being developed that refer to *Communication Services* within the VE.

3.3.1. OSMOS High-Level Security Service

The OSMOS High-Level Security Model evolved from an earlier OSMOS Distributed Object Management Model (Harvey *et al.,* 2001). It is concerned with the management of the Actors, Projects and the overall provision of the VE. It attempts to address some of the primary issues that are central to management of a VE such as: rights & responsibilities and contractual obligations. It is intended that future versions of the models will also take into account legal and IPR issues.

An *Actor* (see FIG. 9) is an electronic representation of a user in the OSMOS system. There are two types of Actors. Firstly, *OrganisationActors* represent organisations and other legal entities within a VE, for example a Company. Secondly, *HumanActors* represent individual people who work for an *OrganisationActor*. In the second case, the usual information is held about a user, such as name, address, email address, telephone number, etc. However, as will be discussed below, it also contains information about what *Subscription* objects the particular user holds (i.e. for which *InformationMetadata* objects that user wants to be notified of any changes to that object).

More importantly, to make the administration of enforcing *DefaultAccessRights* to particular objects easier, a user can only interact with any object in the OSMOS architecture through the *ProjectRole*s (s)he has been allocated. An *Actor* must hold at least one *ProjectRole*.

A *VEProjectRole* corresponds to an actual ProjectRole that is held within a VE (or *VEProject*). An example of a *ProjectRole* could be “Project Manager”, “Client”, “Secretary” or “Architect”. This is vital because it allows the OSMOS System to capture some semantics about who performed a task within the VE and the *ProjectRole* in which they performed this task, and is also used to enforce access rights (*AccessRights*) to certain methods, functions, options, etc (*TechnicalMethods*) on a particular *SystemObject* or service.

ProjectRoles are actually assigned to *OrganisationActors*. This is to clarify that the Organisation is contractually responsible, in a Construction Project, to deliver goods and services. It is *HumanActors* that, through their daily work, allow the *OrganisationActor* to do this. Therefore, it makes more sense to assign *ProjectRoles* to *OrganisationActors*, and then allow the *OrganisationActor* to decide whom, in their company, will carry out each *ProjectRole* on behalf of the organisation. *ProjectRoles* are defined entirely for the VE Project in question, as they would differ on a project-to-project basis.

The vast majority of objects within the OSMOS Architecture inherit from the *SystemObject* object, which allows control over the access to any *Operation* of any object within the system, for anyone holding a particular *ProjectRole*. This is vital because control of access will also be required to objects within the OSMOS architecture that are not directly related to *InformationObject*-type objects being managed. This enforces desired rules for example, a user will be unable to modify his or her *Actor* profile to assign himself or herself a *ProjectRole* of VE Administrator, or to modify access rights for *ProjectRole*s in general. Allowing permission to invoke this operation only to users holding the *ProjectRole* of Project Administrator can enforce this.

The alternative to this is controlling access based upon Actors themselves. However, during the lifetime of a VE it is more likely that users will change frequently whereas the ProjectRoles held by those users would stay relatively static. Therefore, the VE will be easier to administer in terms of access rights if these are based upon ProjectRoles and ProjectRoles are assigned to Actors.

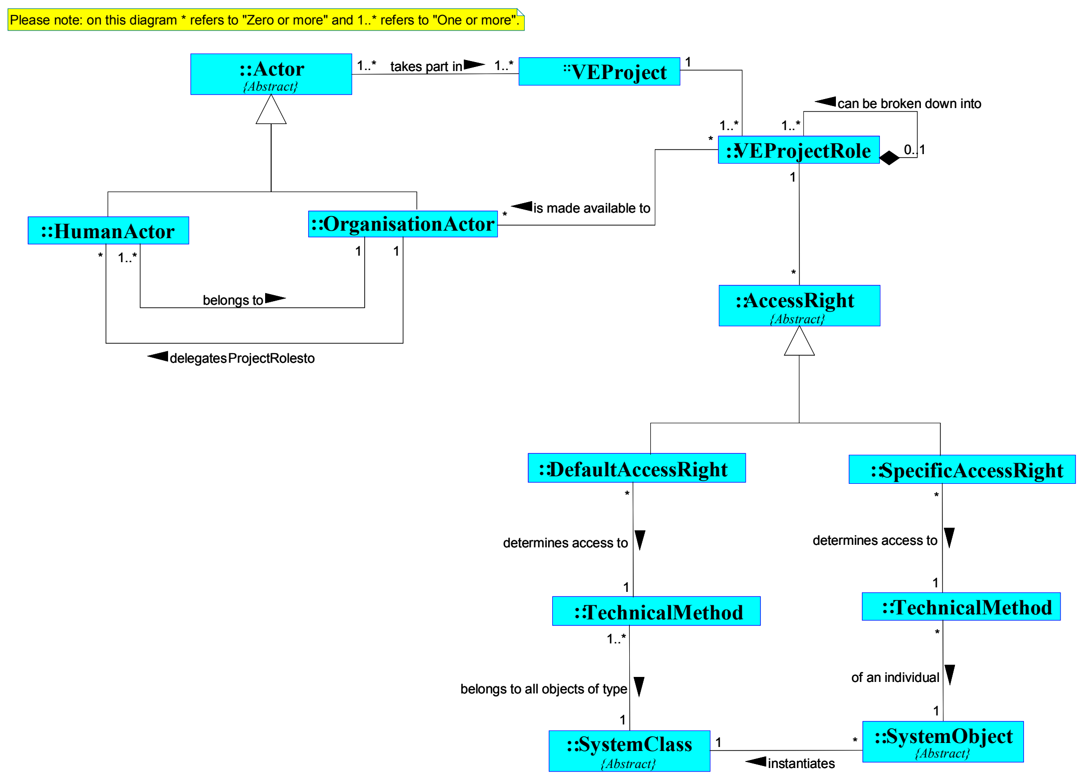


FIG. 9: OSMOS High Level Security Model (Class Diagram)

An example will be used to illustrate this concept: taking an object of type *InformationVersionMetadata* called “D2.1”, one of the *Operations* of this object could be “OpenDocument()”. It may be desirable to allow all users holding the *ProjectRole*s of "Architect", "Client" and "Project Manager" to be able to open this document. However, it may be undesirable to allow those that hold the *ProjectRole* of "Secretary" to be able to do this. Firstly the "OpenDocument()" operation finds out the *ProjectRole*s assigned to the user who has logged in, and checks if (s)he is authorised to invoke this operation. If the user is authorised then the document is opened; otherwise the user is presented with an unauthorised action message. Therefore, using *DefaultAccessRight* objects, for the "D2.1" object, those holding a *ProjectRole* of "Architect", "Client" or "Project Manager" can invoke the method "OpenDocument()", whilst those holding the *ProjectRole* of "Secretary" cannot invoke this method, and this is automatically enforced when the "OpenDocument()" operation is called.

In cases where an *Actor* holds more than one *ProjectRole*, then each *ProjectRole* needs to be checked to see if access is granted. In the example above, if a particular user has the ProjectRole of "Architect" and that of "Secretary", then invoking the "OpenDocument()" operation would be allowed because an "Architect" has permission to do this.

The whole concept of deciding access rights based on *ProjectRole*s, rather than directly on *VEParticipant*s, and also to base access rights on *TechnicalMethods* instead of the usual "CRUD" (Create, Read, Update and Delete) metaphor used in the majority of IT-based systems, allows the VE to limit access to operations based on their meaning rather than how those operations are carried out. This approach has been successfully demonstrated as part of the information management models in previous projects such as COMMIT (Brown *et al.,* 1996) and CONDOR (Rezgui and Cooper, 1998). The *AccessRight*s extends this, although the discussion here concentrates on two specific classes that inherit from *AccessRight*.

The authors propose that an *AccessRight*, as discussed above, is either a *DefaultAccessRight* or a *SpecificAccessRight.* The *DefaultAccessRight* allows access for a particular *ProjectRole* to a particular method for all objects that are instances of a particular class. For example, all users holding the ProjectRole of "VE Manager" may be allowed to call the "modifyProjectRole()" operation on **all objects of the class** *VEParticipant*; whilst not allowing any other user holding a different *ProjectRole* to do this. However, there may be occasions where all users that are holding a certain *ProjectRole*, e.g. "Project Manager", will need to be able to call that same method, "modifyProjectRole()", **on a particular instance of the class** *VEParticipant*, e.g. "Joe Smith". This can be done by using the *SpecificAccessRight* as well as the default right presented above. So, in this example, all "VE Managers" and "Project Managers" can call the "modifyProjectRole()" in "Joe Smith", whereas only "VE Managers" can call the same method in any other *VE Participant* object. When discussing access to a particular method of a particular object for a particular ProjectRole, therefore, a *SpecificAccessRight* will override that object's *DefaultAccessRight.* If the former does not exist, then the latter will apply. Every class should have a set of *DefaultAccessRights* for each of the *ProjectRole*s within the OSMOS architecture.

3.3.2. OSMOS Information Management Service

The OSMOS Information Management Model is split across two separate diagrams for ease of explanation. The first (FIG. 10) covers three of the four main areas supported by the model: information versioning, information ownership and semantic relationships between information, the second (FIG. 11) details information classification.

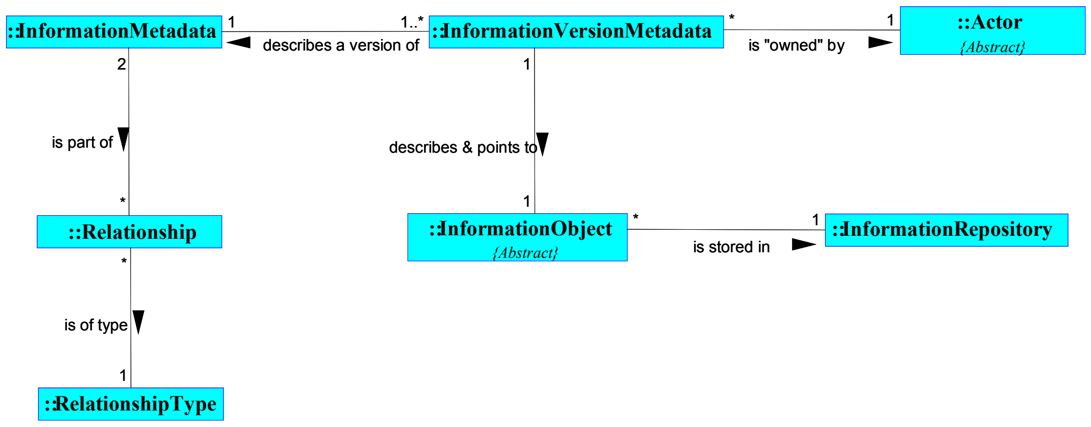


FIG. 10: OSMOS Information Management Model (Class Diagram)

An *InformationObject* could be a word processor document, an IFC File, a spreadsheet, a CAD drawing, etc. The OSMOS Server ("Role A Server") does not store such files. It simply holds metadata-type descriptions of these objects, and a reference to where they are actually stored. These objects are actually stored in a Document Management System provided by a Third-Party Service Provider ("Role B Server"), such as SGTi at Derbi for certain types of document, or a Product Model Server for other types. An *InformationMetadata* object **is** stored within the OSMOS Server ("Role A Server"). It contains a *description* of what that object is, who uploaded it, where it is stored, the date/time, keywords, a list of "subscribers" and "owners" of that object, etc. This *metadata* information is stored in an *InformationMetadata* object that is stored in the Role A Server's *OSMOSRegistry*.

Information Versioning

The model supports Versioning of Information. Contrary to what is written above, the location of where the document is stored is actually held in an *InformationVersionMetadata* object, along with other metadata that is specific to that version (such as a brief description of what has changed in the new version, date/time of upload, version number, etc). Each *InformationVersionMetadata* object can only be referenced to one *InformationVersion* object (which itself can have many *InformationVersionMetadata* objects describing its many versions). In this way, every single version of every nugget of information that passes through OSMOS is stored for reference purposes (and is therefore available in the event of contractual or legal disputes).

Information Ownership

An *InformationVersionMetadata* object also holds a reference to an *Actor* object. This is to signify that the document the *InformationMetadata* object describes is "owned" by that particular *Actor*. This *Actor* can then define the *AccessRights* for that particular object if (s)he so desires. This feature would be important if workflow services were to be included in the VE (as *Ownership* could then imply *Responsibility* for a certain nugget of information at a certain point within a workflow).

Semantic Relationships between Information

In the original OSMOS Semantic Multimedia Document Model (Harvey *et al*., 2001), there were only two types of *Relationship* between information (or *InformationMetadata* objects). These were either *Aggregation* or *Reference* type relationships. This enforcement was seen to be too restrictive. For example, an individual project may have a requirement for other sorts of relationship that are specific to that project.

The new model has, therefore, generalised the concept of a *Relationship*, so that *RelationshipTypes* can be defined at the Project level. Each instance of a *RelationshipType*, a *Relationship* object, can then be formed between two *InformationMetadata* objects, which is where the semantics of that relationship can be captured. Indeed, examples of *RelationshipType* objects could be "aggregation" and "reference".

An object *(InformationMetadata)* is said to have an 'Aggregation Relationship' when it conceptually contains, or is part of, another object *(InformationMetadata)*. Although indicating a direct Association or Aggregation relationship in UML could have showed this, using this method means that it would have been impossible to show how the semantics behind this relationship would be captured in the model. The ability to capture the semantics behind an aggregation relationship is vital to allow the user to state, if applicable, **how** the first *InformationMetadata* object is related to the second *InformationMetadata* object. This knowledge, unless explicitly stated, can be lost when a user who wasn't closely involved in the creation of these objects reads the document at a later stage in the project. One example of such a relationship could be an aggregation relationship between a *Car* and an *Engine* object. The semantics that need to be captured about this relationship would be something along the lines of *"The car contains an engine because without the engine, the car wouldn't move!"*

Similarly the notion of a 'Reference Relationship' is present. Such a relationship comes into play when one object (*InformationMetadata)* needs to refer to another. Again, although indicating a direct Association relationship in UML could have shown this, using this method means that it would have been impossible to show how the semantics behind this relationship would be captured. The implicit knowledge that can be captured and stored about this relationship is again of tremendous potential value to aid the understanding of users, especially if they are new to the VE. The ability to capture the semantics with added value for the business behind this relationship is vital to allow the user to state, if applicable, **how and why** the first *InformationMetadata* object is related to the second *InformationMetadata* object. One example of such a relationship could be a reference relationship between a word processor document and a spreadsheet. The semantics that need to be captured about this relationship would be for example *"This document refers to financial data, this is available in: "*. Because relationships are defined at the *InformationMetadata* as opposed to the *InformationVersionMetadata* level, they "automatically" cascade down the line when new versions of documents, etc. are produced.

A final, slightly more complex, feature of the OSMOS Information Management Model is that of Information Classification. This has evolved from the InformationElementSemanticClassification class that was proposed in the first iteration of OSMOS (Harvey et al., 2001). Based upon the feedback from the end-users, it was felt that this area needed to be revised thoroughly and more explicitly defined at the modelling stage.

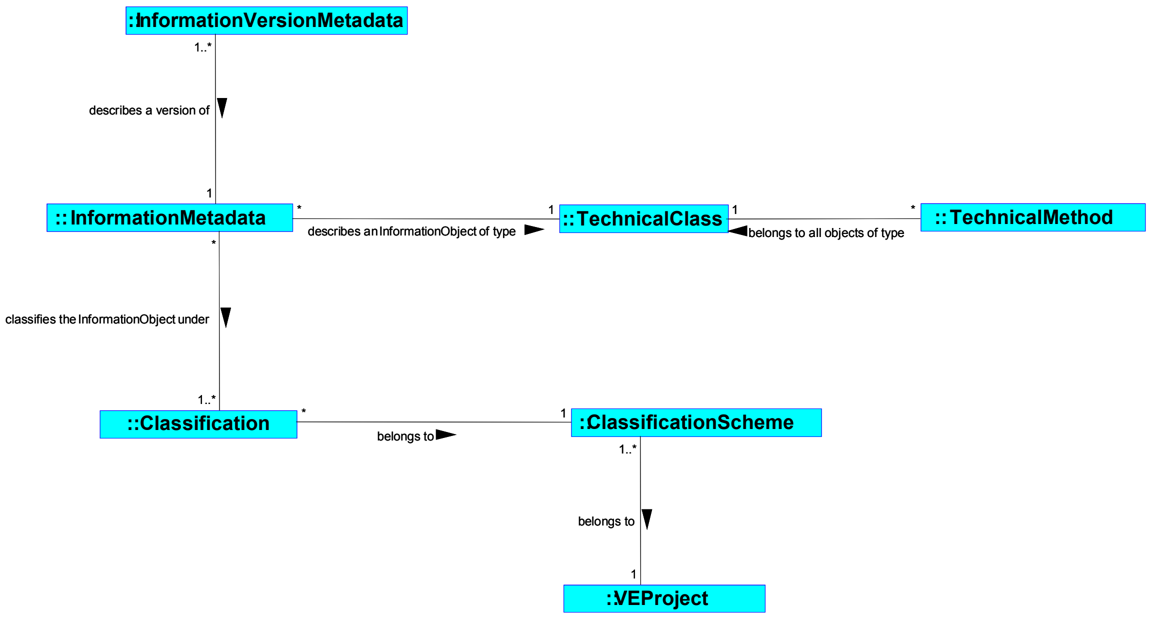


FIG. 11: OSMOS Information Classification Model (Class Diagram)

Information Classification

Information can be classified in a number of ways on a project. An object (for example a word processor document) could be classified by its status (e.g. draft, released), by project-specific identifiers such as workpackages (e.g. WP1, WP2, WP3), by its type (e.g. MSWord Document), by the project's iteration (e.g. 1, 2) or by its semantics (e.g. Project Deliverable). The problem is that such classification schemes change on a project-by-project basis, depending on how the VE agrees to manage its data.

The Classification Model (FIG. 11) was developed to support this fact. It allows the user to view the information held by a VE Project based upon certain criteria. These criteria are based upon *ClassificationSchemes* defined at the VE Project level. Taking the scenario mentioned above, examples of *ClassificationSchemes* include Workpackage Number, Semantics, Status, Type of Document and Iteration. For each of these schemes, a set of *Classifications* would be defined at the VE Project level, such that classifications for 'Status' could be Draft, Work in Progress, Approved, Unclassified, Signed Off and Delivered. For 'Workpackage Number' they could be 1, 2, 3, 4, Unclassified, 5 and 6 and so on.

The Model states that for every nugget of information managed by OSMOS, its *InformationMetadata* object must be referenced to one *Classification* object for each *ClassificationScheme.* As new versions of the information object are produced over the course of the project, then the object's author updates the *Classification*s accordingly. This therefore allows different *Actor*s to have different views of the same information that is managed by the OSMOS Server. At the current time, *AccessRights* are defined upon the *TechnicalClass* (itself, a particular *ClassificationScheme*) of an object, but depending upon feedback from the OSMOS partners, *AccessRights* could be based upon a particular *ClassificationScheme* instead.

To conclude, the OSMOS Information Management model allows the VE to integrate a set of repositories as information datawarehouses (the “Role B” Servers discussed above) and make them accessible to the applications, end-users and OSMOS Services. The OSMOS Information Management Service provides added value to those repositories, by allowing those who take part in projects to locate, then view, all of the information held in a project; to specify relationships between information; to classify information; and to track and manage versioning (including describing the differences between versions) of information, regardless of the physical location and technical structure of such objects or files.

In addition, the final iteration of OSMOS will look at further integration of the OSMOS Communications Service (which is currently under development) with the Information Management Services, because it is felt that project information is valid regardless of the form in which it is transferred (e.g. as an e-mail, fax, document, etc). Messages should therefore be handled in much the same way as other types of information managed within OSMOS.

3.4. The OSMOS Core Services API

In addition to the aforementioned models, the OSMOS API defines a set of API calls for both the High Level Security Service and the Information Management Service. This API is technologically neutral and, as discussed below, can be invoked in a variety of ways. The API has been presented in (Marache *et al.*, 2001), and is therefore not discussed in detail in this paper.

4. IMPLEMENTATION OF THE OSMOS PLATFORM

4.1. Implementation of Role A Server and API

The OSMOS platform federates some Services inside a common framework, and allows their use and collaboration. They are often distributed, heterogeneous programs, and developed by several companies. The OSMOS framework handles two categories of Services: Core Services and Third Party Services (TPS).

4.1.1. Core Vs. Third Party Services

Core Services

The Core Services are the heart of the OSMOS framework. They are Role A components designed, developed and maintained by OSMOS framework providers. They provide common functionalities for every OSMOS-enabled VE information system, such as Actors management, Roles management, Projects management, and so on, by implementing the OSMOS APIs.

The API and the business logic as designed in the Sequence Diagrams were implemented as a set of Java interfaces and objects. These objects were persisted into MySQL, a freely available and standards-compliant Relational DBMS (available for free download for many platforms at http://www.mysql.com). The mapping between Java objects and relational tables was provided by the open source Castor package (available for free download for any platform that supports Java at http://castor.exolab.org.). Everything (except MySQL) runs in a standard Java Virtual Machine (available for free download for any platform from http://java.sun.com.) and has been tested extensively on Windows and Linux platforms.

Third Party Services (TPS)

TPS are Role B applications made available via the World Wide Web. They provide high-level tools (Facilities Management, Building-oriented EDM, etc.) to the Role C end user to enable working in the VE, and they rely on the OSMOS Core Services to work within an OSMOS framework (using OSMOS features such as Cross-Referencing, Access Rights Management, Actors Management, etc.) The OSMOS framework is an entry point for accessing TPS, and the user, according to his or her role, will use them transparently in the VE.

4.1.2. System Architecture

The OSMOS Role A Server is made up of Core Services, TPS and an “Execution Framework” for these services, allowing their collaboration and access. As noted above the Core Services were implemented as Java Objects and the TPS as Web-enabled applications. A closer look at the underlying infrastructure of the Role A Server, also known as the OSMOS Service Manager, is now presented.

The Service Manager is a Role A Server software component whose role is to handle Core and TPS Services as described above. The OSMOS Services "registration" process will be used to illustrate this. When a Core or TPS Service is to be included in an OSMOS framework, the administrator of the OSMOS Role A Server has to register it into the Service Manager. For Core Services, the registration process is the binding of an Object Reference into the Service Manager. Once the reference is known, any Object can be used transparently as a Core Service. For TPS, the Role A Administrator has to provide an XML API (describing their published methods) and register this into the Service Manager. The TPS Web Site is then made available from the OSMOS framework. This approach is similar to the Web Services Description Language (WSDL) concept that has emerged with the SOAP technology from the World Wide Web Consortium (http://www.w3c.org).

The Service Manager is fully dynamic, and it allows run-time registration (or de-registration) of both Core and TPS Services. It also provides some Introspection (also known as Reflection) features, allowing Run Time Type Information about registered Core and TPS Services such as the methods and parameters that they support. The invocation of methods on both Core and Third Party Services is also provided by the Service Manager (using Delegation), so that clients do not have to manage the underlying Services directly.

4.1.3. Accessing the OSMOS API over the Internet

All the programs described above are Java Objects, local to the Java Virtual Machine (JVM) where they are running. A mechanism allowing remote access to the OSMOS API had also to be implemented. It was decided to implement a "Gateway" to the OSMOS Role A Server by using the Java Remote Method Invocation (RMI) technology. This powerful Distributed Objects feature, included by default within the Java Language, allows the creation of Server Objects that are available to Client Objects running in a separate JVM, regardless of the location, i.e. making these objects available transparently over a network or the Internet.

The OSMOS API Invoker is a remotely available Server Object that allows access to the OSMOS API over the Internet, by invoking methods on the Service Manager registered Core and Third Party Services. When a client invokes a method of the OSMOS API, the OSMOS API Invoker delegates the invocation to the Service Manager, and then sends back the return of the call to the client. The OSMOS API Invoker also helps to handle security issues. It is the only entry point to the OSMOS Role A Server (including Core and TPS APIs), and it relies on the OSMOS Security Service to do Access Rights Checking for each invocation on the OSMOS APIs. So, if an invocation cannot be granted, the OSMOS API Invoker rejects it by throwing a specific Exception.

It is worth noticing that Java was a good candidate for its native Distribution features, but the whole framework could have been implemented using other Distributed Objects technologies such as CORBA or COM+.

4.1.4. Interoperability Issues – The "X" Layer

Since the beginning of the OSMOS Project it was realised that the OSMOS framework had to provide technology-neutral access to the Role A Server. As shown above, the framework is entirely Java based and accessible by only the OSMOS API Invoker. To overcome this an interoperable layer based on XML was built on top of this component. The key concept is to describe method calls in XML and receive the returns in XML as well. Using this mechanism, any client can express an XML method description, invoke it, and handle the result of the invocation. This approach again is strongly inspired from the SOAP and XML-RPC specifications from the World Wide Web Consortium, who had the idea of describing and executing procedural calls using XML to hide any implementation detail to the client. At the time of development SOAP was not fully mature.

Using the HTTP protocol to carry these XML messages provides a simple request-reply mechanism, similar to the invocation of a method. Thus, the "X" Layer is implemented as a Java Web Server, serving OSMOS-specific requests only. It accepts HTTP requests containing valid OSMOS method descriptions and invokes the methods using the OSMOS API Invoker over RMI.

Below is a simple example of an XML method invocation description (for authentication of a user):

```
<XMLRequest>
```

```
      <user_id>ROLE_A_ADMINISTRATOR</user_id>
      <project_id>SERVER_ADMINISTRATION</project_id>
      <service_type>CS</service_type>
      <service_name>ActorManager</service_name>
      <method>
            <name>authenticateUser</name>
            <params>
                  <param>
                        <name>login</name>
                        <value>mylogin</value>
                  </param>
                  <param>
                        <name>password</name>
                        <value>mybirthdate</value>
                  </param>
            </params>
      </method>
</XMLRequest>
```

Since clients can invoke methods using XML, the returned values for these methods also have to be formatted in XML.

The methods of the Core Services (i.e. OSMOS API Calls) currently return Java objects or values. Thus, a mechanism allows the OSMOS framework to convert these Java types into XML documents. A "Java-to-XML" converter has been written so that return values, including "Business Objects" of the OSMOS API such as HumanActor, Project and Role can be expressed and handled in XML. When the "X" layer receives an XML request, it:

- Handles and parses the request, finding out which Core Service, method and parameters have to be invoked
- Invokes the requested method (using the OSMOS API Invoker)
- Gets the invocation returned value (Java object)
- Converts it to an XML message using the Java-to-XML converter
- Sends back the XML message to the caller

Using this mechanism, the invocation of a method on a TPS can be entirely realised using XML. The response of the invocation example would then be like this:

```
<osmos_response><result><value>TRUE </value></result></osmos_response>
```

TPS are Web enabled, and thereby the return type of a method invocation is a string, usually containing HTML or XML. In the OSMOS case, they will always return HTML because OSMOS targeted Third Party Services are classical Web Sites and not Web Services. Thus, the return does not need to be converted to XML in opposition with Core Services invocations.

Invoking a Third Party Service using the OSMOS API Invoker returns a Java object wrapping the HTTP response as a string. The "X" Layer then receives the content of the reply and returns it to the caller. Therefore, when the "X" Layer receives an XML invocation request, it:

- Handles and parses the request, finding out which Service and method has to be invoked
- Invokes the requested method (using the OSMOS API Invoker)
- Extracts the HTML resulting of the HTTP request that was sent to the target Web Site
- Sends back the HTML message to the caller

The caller then extracts the HTML from the response and handles the result, usually by displaying the HTML page. By using these mechanisms, the "X" Layer provides a simple and technologically independent access to the whole OSMOS API.

4.1.5. OSMOS Tools (reference implementation)

Several tools (reference implementations) have been developed (some are still under development) that provide users with access to the OSMOS platform. The tools basically make calls to the OSMOS API and invoke required methods to deliver a given functionality. It should be noted that these tools only constitute reference implementations. With OSMOS API response delivery in XML, such tools can easily be customised using templates, style sheets, etc.

Role A Server Tool

This tool (implemented using Java Swing classes) provides functionality to a Role A Server Administrator to manage the server. Available functionality includes:

- Service management and API invocation logging
- Core service registration and deregistration
- TPS registration and deregistration
- X layer registration and deregistration
- TPS monitoring service (to be implemented)

VE Server Administration Tool

This tool (presented using Java Server Pages) is a web-based environment for facilitating the configuration and maintenance of the Role A server. It is to be noted that this tool is a simple interface to relevant API calls for initialising a VE project. The basic functionality provided by this tool includes:

- Set-up, initialisation, and removal of VE projects
- Registration of organisations and employees
- Configuration of access to core and third party services
- Audit trailing and monitoring of usage patterns
- System backups

VE Project Administration Tool

This tool (see FIG. 12) is basically a subset of the *VE Server Administration* tool and is developed in particular to configure and manage any VE project. Once a project has been set up, control is passed over to a "VE Project Administrator" who then uses this tool to configure and manage it. Some of the basic functionality offered by this tool includes:

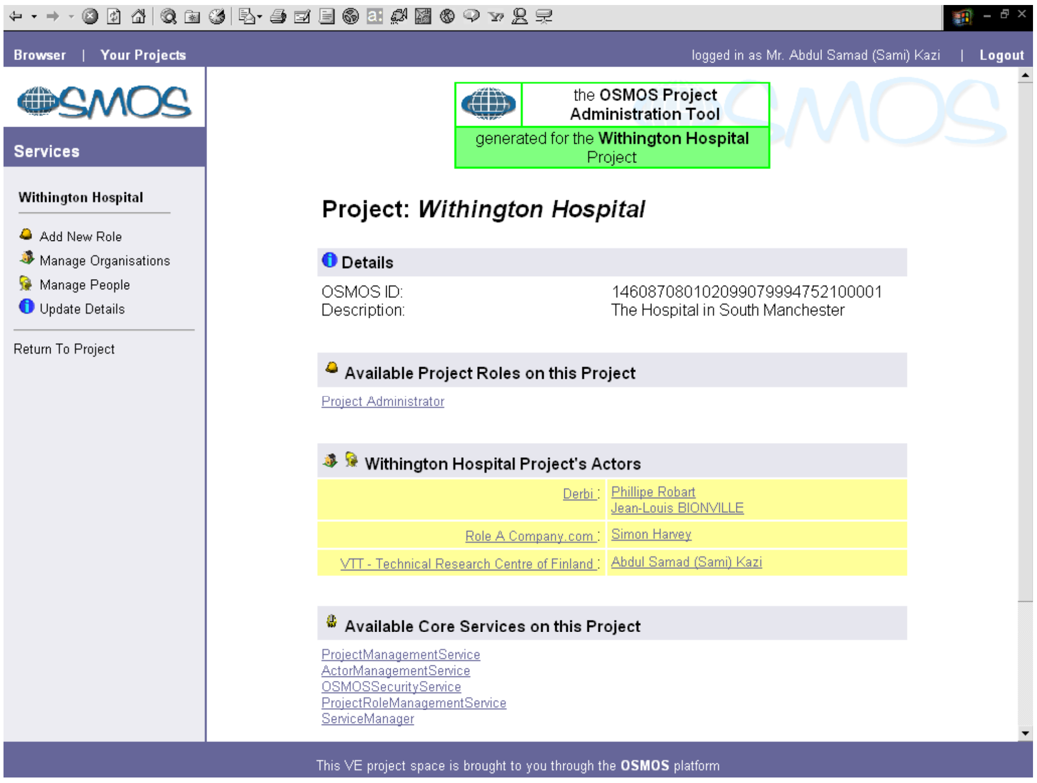
- Creation, definition, maintenance and deletion of instances of OSMOS objects such as: VE participants, Project roles, Technical classifications, Information metadata, Information repositories, etc.
- Creation, modification, and deletion of relationships between different project specific objects
- Management of VE participant organisation metadata, e.g. employees, roles, access rights, etc.

Web-based Information Browser

This tool acts as a low-level entry environment to the OSMOS workspace. In simple terms, the objective of this tool is to present and expose to VE participants based on their roles and associated access rights, the different objects and their associated service methods to which the VE participants have access. In particular, this tool relies primarily on the Information Management Service (see section 3.3.2). Offered functionality includes:

- Creation of different classification schemas

- Browsing registered objects based on classification schemas
- Cross-referencing information objects (e.g. cross-referencing a document to an IFC object)
- Information metadata management, etc.
- Information versioning
- Etc.



Browser | Your Projects   logged in as Mr. Abdul Samad (Sami) Kazi | Logout

Services

Withington Hospital

- Add New Role
- Manage Organisations
- Manage People
- Update Details

Return To Project

the OSMOS Project Administration Tool
generated for the Withington Hospital Project

Project: *Withington Hospital*

Details

OSMOS ID:	146087080102099079994752100001
Description:	The Hospital in South Manchester

Available Project Roles on this Project

Project Administrator

Withington Hospital Project's Actors

Derbi :	Phillipe Robart Jean-Louis BIONVILLE
Role A Company.com :	Simon Harvey
VTT - Technical Research Centre of Finland :	Abdul Samad (Sami) Kazi

Available Core Services on this Project

ProjectManagementService
ActorManagementService
OSMOSSecurityService
ProjectRoleManagementService
ServiceManager

This VE project space is brought to you through the **OSMOS** platform

FIG. 12: VE Project Administration Tool

4.2. Implementation of Role B Services

Role B services may be integrated at any time into the OSMOS framework through TPS registration offered through the Role A Server tool. Here a TPS may be registered, and its available methods and associated parameters identified. FIG. 13 shows a screen shot of an early-developed TPS registration interface.

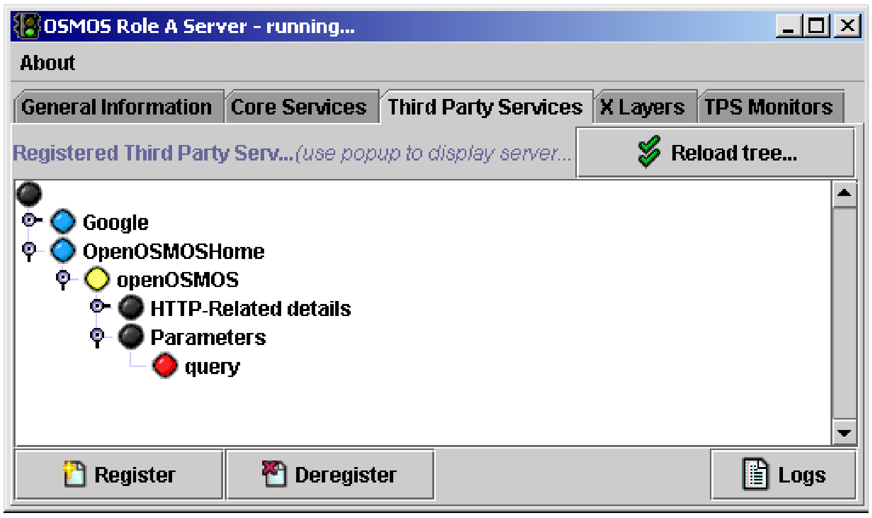


FIG. 13: Third Party Service (TPS) registration

Once a TPS has been registered the service is no different than an OSMOS core service from the perspective of the end-user (VE participant). The actual difference is that method invocation on the service takes place at the service provider (Role B) end rather than at the Role A. An example of the same is shown in FIG. 14, where the services developed by two OSMOS partners are “plugged in” to the OSMOS core.

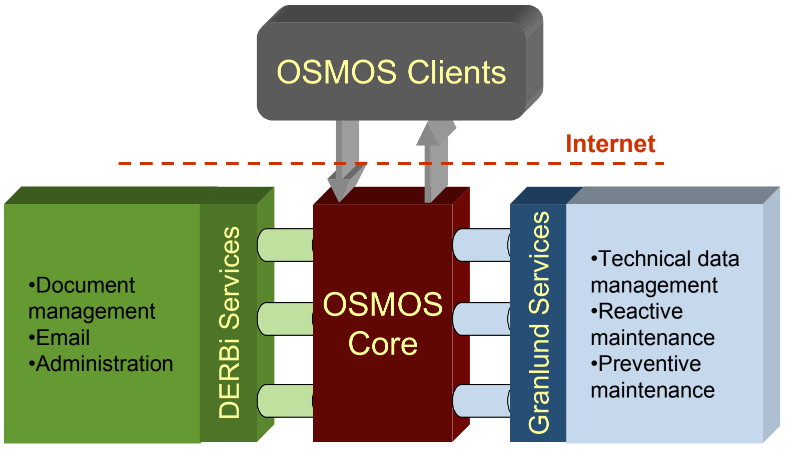


FIG. 14: OSMOS Core and OSMOS compliant TPS services provided by the OSMOS partners

Within OSMOS, two of the partners will be making their services available to OSMOS clients. The services offered are “OSMOS compliant” as they have a capability of invoking methods of the OSMOS API and vice versa. In a similar fashion, other services may also be adapted (through wrappers) and be made OSMOS compliant.

The TPSs (Role B) developed under and made available to OSMOS relate to facilities management and document management provided by Granlund in Finland and DERBi in France respectively. These services are briefly described below.

4.2.1. Granlund, Finland

Granlund offers OSMOS compliant services targeted for building users and maintenance companies to use for facilities management. Concentration is on those services that clearly have a need for location independent functionality and thus are suitable in the form of web-based services. Provided services target technical data management and both reactive and preventive maintenance. The main functionalities offered include:

Technical data management services:

- Add/Edit/Delete object
- Edit object attributes
- Edit object instructions

Reactive maintenance services:

- Add a service request
- Get service requests
- Confirm a service request

Preventive maintenance service:

- Print weekly work order

4.2.2. DERBi, France

DERBi offers OSMOS compliant services for electronic document management and email based communications targeted to the construction industry. The document management service allows documents to be stored on the basis of a defined codification mechanism that can be later used for easy document retrieval, while the email based communication service provides an interface for clients to communicate and share documents. The main functionalities of the services include:

Electronic document management service:

- Store documents on a server
- Search for documents using simple criteria
- Access document meta-information
- Retrieve documents
- Create document folders
- Manage document versions

Email based communication service:

- Email consultation
- Send emails
- Retrieve emails

5. TESTING AND VALIDATION OF THE OSMOS APPROACH

As mentioned in the introduction, the principle means for testing and evaluating the OSMOS approach is through field trials simulating working in a construction VE. Throughout the project the consortium has recognised that the integration of human, organisational and technical elements is a prerequisite for a successful specification of the strategy required, for each end-user involved in the project, to identify and implement the potential changes resulting from the proposed OSMOS approach. With this in mind, therefore, concomitant with the analyses carried out in the requirement capture phase of the project, and coupled with the field trials, the research has included analysis of factors often neglected in similar industry/business and ICT research efforts. This work included the completion of an IT and Construction questionnaire, which provided a profile of each of the companies in terms of the technologies currently in use; teamwork methods and the effectiveness of teamwork; the business environment; business processes and the effects of change; and knowledge management practices. The questionnaire survey was aimed at employees working at the operational/tactical level in each company. Further to the questionnaires, semi-structured interviews were also carried out with more senior people at the tactical/strategic level. Results from the research provided initial validation of the OSMOS approach and the requirements of the resultant system being developed and valuable information that will ultimately enable the formulation of valid and useful business recommendations. The global work to be undertaken in testing and validation includes tasks to:

1) Define and perform field trials based on the OSMOS infrastructure.
2) Test and validate the OSMOS infrastructure.
3) Analyse the likely benefits of adopting the OSMOS approach.
4) Define the migration path (including organisational recommendations) to using the OSMOS approach.

Various commercial web-based electronic document management (EDM) systems are already available, offering document and workflow management services across the Internet, some of which are tailored specifically to the construction industry. Such solutions include offerings from Bricsnet (http://www.bricsnet.com), Buzzsaw (http://www.buzzsaw.com), BuildOnline (http://www.build-online.com) and Citadon (http://www.citadon.com). It is interesting to note that BuildOnline is the only offering aimed specifically at the European construction industry.

The services offered by these companies are based currently on technologies already offered by them. This is not unsurprising, as they are commercial ventures; for example Buzzsaw emerged as a spin-off from the design-software company Autodesk, and thus offers Autodesk software solutions. The OSMOS approach, however, differs in its objectives in this respect. These companies' offerings are basically closed, whereas OSMOS aims to integrate services from many organisations, possibly even making available alternative versions of the same service, in an open way (and not merely as a Web link to a vendor). This should also create a better environment that offers low barriers to entry and thereby supports SMEs. The plug and play provision of third party services thus emerges as a unique selling point of the OSMOS approach, key to which is the unique OSMOS API.

5.1. The OSMOS Field trials

Overall the field trials aim to address three high-level concerns: to ensure that the proposed system works (i.e. to evaluate its usability); to ascertain that it meets and achieves its intended business goals; and to ensure its acceptance by the intended users. In terms of usability, four aspects were taken into account: S*ystem functionality, Efficiency, User-friendliness,* and *Technical aspects*.

5.2. Field Trial Scenarios

The field trials carried out to date have been based on separate work scenarios within two of the end-user companies – Derbi in France and Granlund in Finland – and have allowed evaluation of the OSMOS VE Server Administration tool and the OSMOS VE Project Administration tool.

In the French field trial (Derbi), seven professionals from Derbi and OTH (the parent company) were selected to participate based on their experience and skills. Two separate sessions were organised and the OSMOS tools were manipulated to simulate the set-up and management of a construction VE. The first session involved professionals from the IT industry who were experienced in the management and use of electronic platforms supporting VEs. These professionals had participated in the development of Derbi's SGTi tools (EDMS) (http://www.sgti.com.fr). During the trials, they manipulated all the OSMOS tools with an emphasis on the OSMOS VE Server Administration Tool. The second session involved professionals from the construction industry with some basic knowledge of ICT. These people were experienced in construction projects management and teams both at the design and site works stages. They represented the OSMOS users or clients (OSMOS Role C), concentrating on the VE Project Administration Tool. Together, the individuals taking part represented the roles of VE Server administrator, TPS provider, VE Project administrator, and VE Project participants.

In the Finnish field trial the testing scenario involved using the OMSOS VE administration tools to set up a new VE and the use of Granlund's proprietary web enabled FM software 'RythiWeb' (http://www.granlund.fi). Three organisations (companies) representing a Building Owner, a Maintenance Company and a Facility Consultant were created as organisation actors in the VE. Employees were then registered to each organisation with usernames and passwords. A new project was then registered to the system and the three organisations within the

VE were assigned to the project, thereby allowing the existing employees also to be assigned. Finally six different project roles were created in the VE and subsequently assigned to the various actors, thereby controlling the services to which the employees have access within the project. Further to the testing of the administrative side of the VE, three of the users tested the Granlund tool over the WWW. After logging in to OSMOS, messages were sent and returned between OSMOS and the Granlund Web enabled FM tool.

5.3. Critique

Analysis of the results from the field trials showed generally satisfactory results overall according to the usability criteria. Some issues were noted regarding the intuitiveness of the user interface, and navigability issues, but these are not of major concern in terms of this paper, especially as the tools tested were reference implementations only. It is interesting to note however, that cultural differences were apparent between the results from the two countries involved.

At the time of writing the field trials carried out were limited to testing the OSMOS infrastructure and approach in terms of validating the GVEPM and the administration tools that had been developed at that stage. This is, however, an important step before the full field trials, which will include integration of third party and other services to be used by OSMOS Role C companies. The results validated the GVEPM in terms of the processes involved in administering the VE at both the server and project levels. The activities within the GVEPM that relate to providing, maintaining and using TPS and other services are logically straightforward and do not impinge greatly on the model overall. It is expected that once field trials have been carried out to test the complete OSMOS approach the GVEPM (and therefore the underlying functionality of the OSMOS tools) will prove to be accepted throughout. The approach taken in OSMOS will inevitably lead to some business process re-engineering, and as the processes around which the system is based are generic, there may be some need for training within companies wishing to enable their construction VE through OSMOS.

The trials also validated the OSMOS Roles (A, B, and C). The roles were accepted as a strong and logical underlying concept within the OSMOS consortium, and it is an excellent result that the users who took part in the field trials – who were not previously associated with the project – found agreement in these concepts.

In terms of the OSMOS API, again the evaluation will only become truly apparent when the OSMOS platform is tested with a complete set of services available. During the French trials Derbi were able to register and deregister SGTi from the framework very easily, which also augurs well for the rapid set-up of a VE in which Role C SMEs can take part.

6. FUTURE DIRECTIONS

The field trials so far carried out in the project have served as a good interim evaluation for specifics of the OSMOS approach. However, in order to test the true added value of the OSMOS approach, and also to enable objective business process recommendations and a strategy for implementing the OSMOS method, a final full-scale field trial is required. The critical success factor required for this field trial is full integration of TPS, accessible over the Internet to OSMOS Role C actors. Such a field trial is currently being planned, involving all three OSMOS end-user organisations, based on the following tentative storyboard:

Scene 1

- A Role C company (JM) requests from the Role A company (Derbi) the set-up of a VE project environment for the construction and management of a business complex
- Requested third party services (Role B) for the project are:

– Document management services from DERBi

– Email services from CSTB

– FM Services from Granlund

- TPS provide a list of service specific objects and the different methods available to them.

– Documents: upload, download, view, show meta-data, show historic record

- Messages: send, receive, archive, fetch
- FM data management: add, edit or delete objects, edit attributes, edit instructions
- Helpdesk: add request, manage requests, manage workorders
- Maintenance: weekly workorders

Scene 2

- The Role A company is informed of a designated VE Project Administrator at the Role C company
- A VE Project Environment with subscribed services is initialised and the VE Project Administrator is registered
- The VE Project Administrator elaborates and registers a list of organisations, agreed roles, and access rights
- Participants involved in organisations are registered and roles delegated to them.

Scene 3

- The VE Project Administrator launches the VE project environment and participants are notified
- Participants start using the VE project environment and associated services
- Partners will discuss their experience in using the VE Project environment created and hosted through OSMOS

It was possible to make some preliminary business recommendations specific to the companies involved in the project from analysis of the interim field trials results, combined with the research into human and organisational issues. It is the intention of the OSMOS consortium to be able to make business recommendations relevant to the construction industry as a whole in terms of adopting the OSMOS approach, following the final field trial.

Further to this it is envisaged that time/cost analyses will also be included in the final trials. Results from this will provide a good indication of the validity of the OSMOS approach in terms of efficiency improvements over the processes currently employed. From these results it should be possible to propose a general migration path, including potential training requirements, for potential users outside the consortium.

7. CONCLUSION

The OSMOS project aims to provide a generic open platform to enable construction enterprises, including SMEs throughout the European construction industry to enhance their current capabilities. This will be facilitated through the ability to plug in services from third parties as well as from member actors within a construction project, quickly and at a low entry-level. Through an incremental and iterative development methodology, the project has developed, and continues to refine, a model-based environment supported by tools to set up and maintain the construction VE and projects according to the specific agreements of the actors involved. Interim testing of the OSMOS approach has shown encouraging validation of the concepts used. The work is ongoing, and concomitant with the technological development, human and organisational issues are being addressed to ensure that the OSMOS consortium can define the likely benefits of adopting the OSMOS approach and recommend a migration path to that end.

To summarise, FIG. 15 concludes the paper by showing a comparison between a traditional and an OSMOS approach to doing work in a construction VE. In essence it describes the “OSMOS impact” as one migrates from a traditional approach to the OSMOS one.

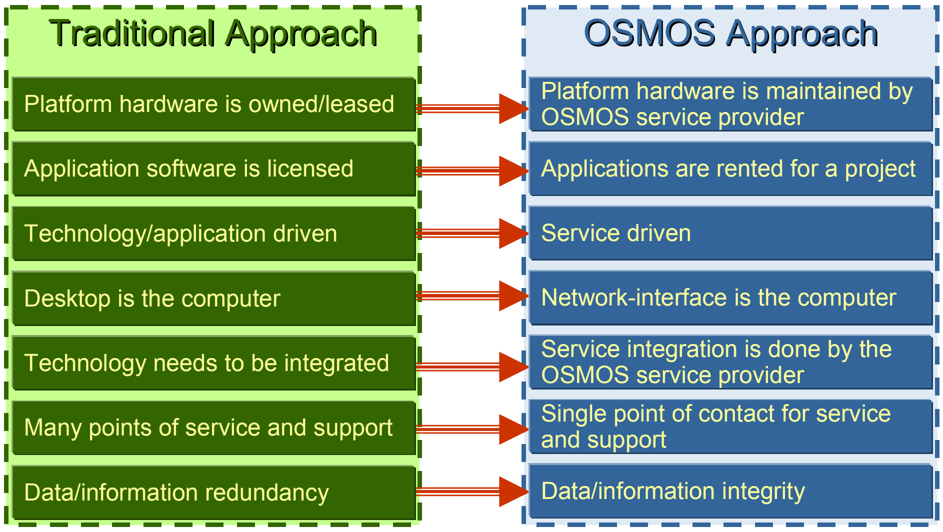

Traditional Approach
Platform hardware is owned/leased
Application software is licensed
Technology/application driven
Desktop is the computer
Technology needs to be integrated
Many points of service and support
Data/information redundancy
OSMOS Approach
Platform hardware is maintained by OSMOS service provider
Applications are rented for a project
Service driven
Network-interface is the computer
Service integration is done by the OSMOS service provider
Single point of contact for service and support
Data/information integrity

FIG. 15: A comparison between the traditional approach and the OSMOS approach

8. ACKNOWLEDGEMENTS

The OSMOS Consortium would like to acknowledge the financial support of the European Commission under the IST programme.

The authors would also like to thank the members of staff from OTH/Derbi, Granlund and JM who have assisted in the research through questionnaires and interviews and by taking part as users in the field trials and completing evaluation forms.

9. REFERENCES

Barker, J., (2000). *Beginning Java Objects*. Birmingham: Wrox Press.

Bennet, S., McRobb, S. & Farmer, R. (1999). *Object-Oriented Systems Analysis and Design using UML.* London: McGraw-Hill.

Brown, A., Rezgui, Y., Cooper, G., Yip, J. & Brandon, P., (1996). *Promoting Computer Integrated Construction Through the Use of Distribution Technology*. Electronic Journal of IT in Construction. Vol.1: p56. Also available on the World Wide Web at http://www.itcon.org/1996/3/.

Dutton, W. H., (1999) *Society on the Line: Information Politics in the Digital Age.* New York: Oxford University Press Inc.

Goranson, H. T., (1999) *The Agile Virtual Enterprise: Cases, Metrics, Tools*, Westport, CT: Quorum Books.

Harvey, S. C., Wilson, I.E., Rezgui, Y. & Cooper, G.S. (2001). *A Comprehensive Description Of The Human, Process And Technical Requirements Of A Construction Virtual Enterprise*: The OSMOS Project. In Proceedings of the 1st International Conference on 'INNOVATION IN ARCHITECTURE, ENGINEERING AND CONSTRUCTION (AEC)'. 18-20th July 2001, Loughborough University, UK.

Marache, M., Harvey, S.C., Vankeisbelck, R., Zarli, A., Rezgui, Y., & Kazi, A.S., (2001). *Specification of the OSMOS services and modelling infrastructure*. OSMOS Project Deliverable (D2.2 Iteration 2).

NIST (1993) Draft Federal Information Processing Standards Publication 183: Announcing the Standard for *Integration Definition for Function Modeling (IDEF0)*. Available on the World Wide Web at http://www.idef.com/Downloads/pdf/idef0.pdf

Object Management Group (1999). *OMG Unified Modelling Language Specification*. Version 1.3. Available on the World Wide Web at http://www.omg.org/cgi-bin/doc?formal/2000-03-01.

Rational Corporation (2001). *The Rational Unified Process*, cited 16 July 2001. Available on the World Wide Web at http://www.rational.com/products/rup/index.jsp.

Rezgui, Y. & Cooper, G. (1998). *A Proposed Open Infrastructure for Construction Project Document Sharing*. Electronic Journal of IT in Construction. Vol. 3: 1998. Also available on the World Wide Web at http://www.itcon.org/1998/2/.

Rezgui, Y., Zarli A., Bourdeau M. & Cooper, G. (2000). Inter-Enterprise Information Management in Dynamic Virtual Environments: The OSMOS Approach. *Proceedings of CIT2000 – The CIB-W78, IABSE, EG-SEA-AI International Conference on Construction Information Technology*, 2000, 28-30 June, Reykjavik, 731-741.

Tapscott, D., (1996) *The Digital Economy: Promise and Peril in the Age of Networked Intelligence*, New York: McGraw-Hill.

The OSMOS Consortium (2001*a*). Proposed intra-company information process model. OSMOS Project Deliverable (D1.1 Iteration 2).

The OSMOS Consortium (2001*b*). Proposed inter-company interaction process model. OSMOS Project Deliverable (D1.2 Iteration 2).

10. APPENDIX A

The OSMOS consortium

The consortium comprises the following six organisations:

Derbi – Derbi is a subsidiary of the French consulting engineering firm Groupe OTH. Founded in 1970, DERBi employs twenty people and has an annual turnover of 3M Euro. The company is engaged in three main fields of activity: Research, software and application design and development; implementation of solutions for computerised data exchange; and network management consultancy.

CSTB – The "Centre Scientifique et Technique du Bâtiment" (CSTB) is the French national public research establishment in the construction field. CSTB's activities cover four major fields: research, technical consultancy, quality assessment and knowledge dissemination.

Granlund – Granlund is the largest engineering company in building services consulting in Finland. Founded in 1960, it employs 240 people, and is privately owned (mainly by the employees). The company's main activities are building services design, facilities management consulting, and development of design and facilities management software.

JM – JM AB (publ) is the fifth-largest construction firm in Sweden and one of the larger property management companies in the country. Founded in 1945, it employs approximately 2,000 people, and is a public company. Restricting most of its activities to the particular niche of housing construction, JM undertakes the complete building project from land acquisition to the development of detailed development plans.

ISI-USAL – The Information Systems Institute (ISI) was founded as a partnership between Salford University and leading businesses in the United Kingdom. The ISI research centre involved in the OSMOS project, one of the most successful in the UK and internationally recognised, is focused around information and management systems applied to the built and human environment.

VTT – Technical Research Centre of Finland, is the largest research establishment in the Nordic countries and one of the largest in Europe. The research group of VTT involved in the OSMOS project focuses on four main activities: design methodology, project management, information networking and product data technology.