FACILITATING INTER-ENTERPRISE INFORMATION EXCHANGE IN ONE-OF-A-KIND SETTINGS

RECEIVED: April 2003
REVISED: August 2003
PUBLISHED: October 2003 at http://www.itcon.org/2003/24/
EDITORS: Robert Amor and Ricardo Jardim- Gonçalves

Abdul S. Kazi, Senior Research Scientist
VTT – Technical Research Centre of Finland
email: Sami.Kazi@vtt.fi http://cic.vtt.fi

Chotchai Charoenngam, Associate Professor
AIT – Asian Institute of Technology, Thailand
email: chot@ait.ac.th http://www.sce.ait.ac.th

SUMMARY: *The construction industry's norm of operation is that of one-of-a-kind production and delivery of products (e.g. a building). The results (products and/or services) are typically delivered through complementary competence sharing between different project participants. Such operational modes naturally have implications on the way that product related information is shared and exchanged between the participating organisations. Information and Communication Technology (ICT) infrastructures and applications while seemingly available are still far from meeting the real needs of inter-enterprise information exchange, and, service integration and management in one-of-a-kind settings. This paper presents the research findings of two recently concluded international projects (GLOBEMEN, and OSMOS) that have explored the operational norm of one-of-a-kind settings (in one case at a generic cross-sectoral level, and in the other in the construction sector). ICT architectures, components and various interaction scenarios are furthermore identified. The results are discussed from the perspective of two case studies. Key lessons learned and barriers to take-up of ICT solutions are presented.*

KEYWORDS: *information management, virtual enterprise, one-of-a-kind settings, interaction scenarios.*

1. INTRODUCTION

One of a kind product or service delivery (e.g construction of a building or facilities management) increasingly demands the one-time collaboration of different organisations to consolidate and synergise their dispersed competencies in order to deliver a desired product or service. This naturally has an implication not only in the way information (related to the to-be-delivered product or service) is exchanged and shared, but the way in which secure, quick to set-up, transparent (to the end-user) and non-intrusive (to the normal ways of work of an individual/organisation) ICT is used for this purpose. This has been a central research theme in various research initiatives (a listing of more than 100 European projects can be found by selecting "Virtual Enterprise" at http://cic.vtt.fi/links/euproj/index.html).

1.1 The Challenge

Clients today demand customised products and services that a single organisation may not be able to deliver. These specialised products and services can be delivered through competence sharing between several different organisations. A virtual enterprise (VE) is then formed to deliver the product or service. For organisations participating in a VE setting, the (ICT) challenge is information exchange between heterogeneous applications and services. This exchange of information should be in a transparent and controlled manner. How for example, can an individual organisation access and integrate information from multiple (external) sources while only exposing relevant parts of its own information.

1.2 Structure of this Paper

This paper starts with a short introduction to operational modes within one-of-a-kind settings, in particular, the virtual enterprise, followed by a discussion on the different mechanisms through which inter-enterprise information is exchanged. Using these as a foundation, the paper then introduces the research context and adopted methodologies of the projects; GLOBEMEN and OSMOS that are under investigation in this paper. The aim of the third section is to raise some of the main questions that have been the subject of investigation and resolution in these two projects. Both projects have developed a list of requirements (functional, architectural, and technical) that should be provided by the appropriate IT solution. These requirements are consolidated in the paper and later discussed in light of a potential ICT architecture, its components, and various interaction scenarios in section 5. Section 6 presents a discussion of the results from the perspective of two cases: one on distributed engineering in construction, the other on facilities management, and some key lessons learned.

2. BACKGROUND

2.1 One-of-a-kind Settings: The Virtual Enterprise

The virtual enterprise (VE) is quickly becoming the preferred organisational form for one-of-a-kind-settings to deliver a one-of-a-kind product. Needless to say, the term itself has been a popular one in recent research undertakings (Browne et al, 1994), (Charbuck and Young, 1992), (Rabelo and Camarinha-Matos, 1996), (Walton and Whicker, 1996). In fact, Filos and Banahan (2001) have coined the VE as, the *Smart Organisation*. Kazi et al. (2001 b) presented a collection of definitions as identified by others as follows:

- "A virtual enterprise (VE) is primarily an interoperable network of pre-existing enterprises with a common goal, that can function as a single real organization" (Afsarmanesh et al, 1997)
- "a Virtual Enterprise is a temporary consortium or alliance of companies formed to share costs and skills and exploit fast-changing opportunities" (NIIIP, 1998)
- "Virtual Corporation is a temporary network of independent companies - suppliers, customers, even rivals - linked by information technology (IT) to share skills, costs and access to one another's markets. It will have neither central office nor organisation chart. It will have no hierarchy, no vertical integration" (Byrne et al, 1993)
- "the Virtual Enterprise consists of a series of co-operating 'nodes' of core competence which form into a supply chain in order to address a specific opportunity in the market place" (Walton and Whicker, 1996)
- "Virtual enterprises materialize by selecting skills and assets from different firms and synthesizing them into a single business entity" (Camarinha-Matos et al, 1998)

Kazi et al (2001) made an attempt to consolidate these definitions, and defined the virtual enterprise as, "a temporary collaboration to exploit a business opportunity (with each business transaction being performed by a collaborating legal entity)". It is interesting to explore this further with regard to some core concepts involving the customer, product or service, enterprise, enterprise networks, and virtual enterprise. This is best understood through the illustration shown in FIG. 1.

An explanation of the main 15 steps (shown above) should help explain this illustration:

1. A customer requires a certain product or service. The customer needs are the input and the received product or service are the output.
2. The customer interacts with an enterprise to receive the required product or service.
3. Where and when possible, the enterprise delivers the product or service that is requested by the customer.
4. An enterprise may initialise a network based on a common or complementary need between network members. ENCORD (The European Network of Construction Companies for Research and Development) that allows members to share best practices and results of ongoing research and development efforts is an example of a network.
5. An enterprise could also lead/facilitate a network.
6. Or alternatively, an enterprise could participate in a network.
7. The main function of a network within the above context is to reach consensus. For example, this may involve research and development of procedures, standards, etc. The network prepares/agrees on

procedures, ICT platforms, standards, etc. to facilitate the collaboration of the network participants as they work together to deliver a product or service.

8. An enterprise that is a member of a network would implement, comply with and use the different standards, ICT platforms, procedures, etc. that have been prepared/agreed upon by the network of which it is a member.
9. The different standards, ICT platforms, procedures, etc. prepared/agreed upon by the network, are used to support delivery of the product or service.
10. When a customer requests a particular type of product or service that the enterprise alone cannot deliver, but that can be delivered through temporary collaboration with other organisations, then that enterprise would initialise a virtual enterprise (VE).
11. An enterprise could also lead a virtual enterprise (usually when it is the main contact point to the customer).
12. Or alternatively, an enterprise may be invited to participate in a virtual enterprise.
13. Once the VE is in operation, its participants collaborate and share information on the basis of common standards, ICT platforms, procedures, etc.
14. The VE coordinates the co-delivery of the product or service requested by the customer.
15. As a member of the VE, the enterprise would contribute towards the development/delivery of the requested product or service.

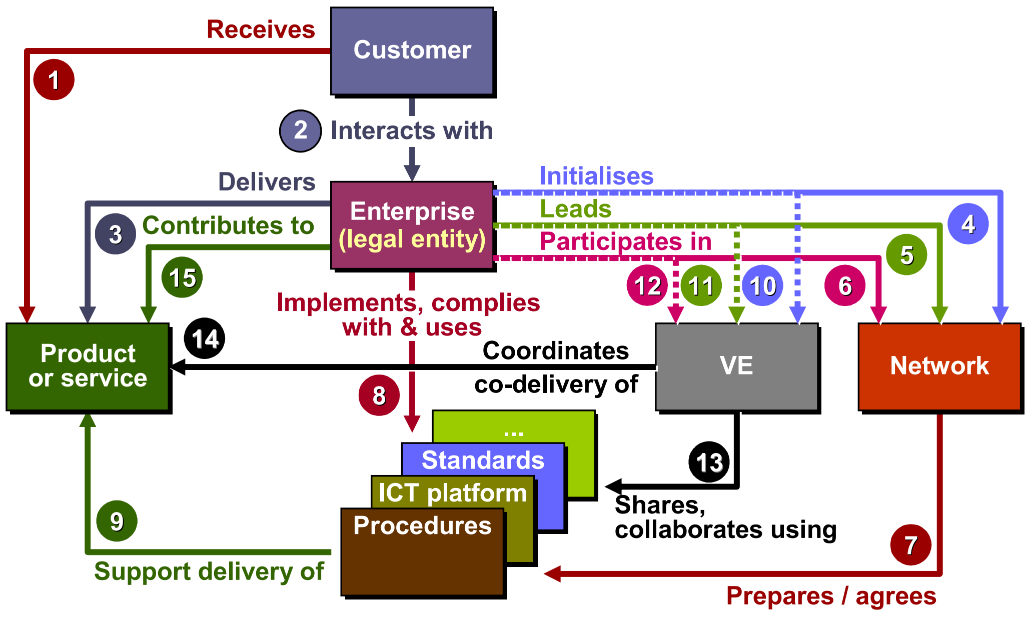


FIG. 1:Virtual Enterprise, core concept

The VE, despite being an operational norm for delivery of one-of-a-kind products and/or services is a complex undertaking. Hannus and Kazi (2000), Kazi et al (2001), Kazi and Hannus (2002), and Wilsonet al (2001) explored some of the salient characteristics of VEs:

- A VE is not necessarily a legal entity
- Some members are not known in advance
- Collaboration between participants in VEs is temporary
- Complementary competence is provided by distinct organisations
- Information flows are not necessarily covered by contractual relationships

- There is an absence of dominant actor to set up the rules of the game
- Members may participate in several other concurrent VEs

These characteristics can act as a departure point for VEs and may be seen as a basis for their complexity. This complexity is evident once a comparison is made between intra-enterprise and inter-enterprise settings as reported by Kazi et al (2001) and Kazi and Hannus (2002), and shown in TABLE 1:

TABLE 1: Comparison between intra and inter-enterprise settings

	Intra-enterprise	Inter-enterprise
Contractual coverage	No	Yes
Legal responsibility	No	Yes
Technology	Probably proprietary	Industry standard
Control of ICT solutions	High	Low/none
Set-up time	Variable (May take long to build and set-up proprietary solutions)	Typically short to allow for early operation.
Training	Probably yes	No (though some minimal or self-training possible)
Application user interface	Yes (~"bells and whistles")	Usually web access or hidden (~"Save as …")
Integration	Integrated applications and databases	Culture, ontology, standards, data warehouses, etc.
Target users	Own staff	May include unknown future partners
Coordination	Resource and workflow management	Deliverable management
Information updating	Synchronous	Asynchronous

2.2 Inter-enterprise Information Exchange Mechanisms in One-of-a-Kind Settings

The successful sharing and exchange of relevant information from one organisation to another are of paramount importance in any inter-enterprise environment: this is particularly so for one-of-a-kind-settings harnessed to deliver a one-of-a-kind product or service. Different mechanisms have been used over the years to enable this sharing and exchange of information (see FIG. 2).

File Exchange: The advent of email has been a powerful step towards rapid electronic exchange of data and information. This advance has not been without pitfalls, especially in inter-enterprise settings. For example, when point-to-point information exchange takes place, data/information redundancy and loss of control of the physical source of the information in addition to its original user is inevitable unless specific measures are put in place, as shown in FIG. 2 (a). At the same time, typically both information provider and recipient need to have access to similar application software or viewers to be able to read/work on the same piece of information.

Project Servers: One solution to the problem of data/information redundancy has been through the introduction of project servers. Here, information is stored in a central information repository that is accessible by the relevant information providers and users, as shown in FIG. 2 (b). Version control is one of the many salient functions used in project servers. The person-project server solution, while resolving the data/information redundancy problem and maintaining information centralisation as opposed to decentralisation in the file exchange approach, is still individual user centric. In simple terms, the interaction is through a user (appointed by the participating organisation) and not the organisation, which in fact is the legal entity involved. As such, when information is released by an individual, legal endorsement of that information by the enterprise is assumed though the enterprise may be unaware of what information was in fact released. Organisations prefer to at times only release "partial" information, while keeping and maintaining the whole "internally" (Hannus and Kazi, 2000).

Links between Enterprise Systems: The way forward as described by Kazi and Hannus (2000) would be through flexible links between enterprise systems, as shown in FIG. 2 (c). Here, an individual would communicate with the central repository of the enterprise, for which the individual is working, this would then release the relevant portion of this information to a shared project server. As such, enterprise specific systems/repositories would transfer and receive information packages on a periodic or per request basis to/from the VE specific project server. At the same time, this complies with legal requirements that information is released by an enterprise and not an individual. An enterprise would simply need to "plug-and-communicate" to the VE repository. Where the enterprise systems and VE repository do not share a common data format, a simple mapping/translation component may be used to enable interoperability between both.

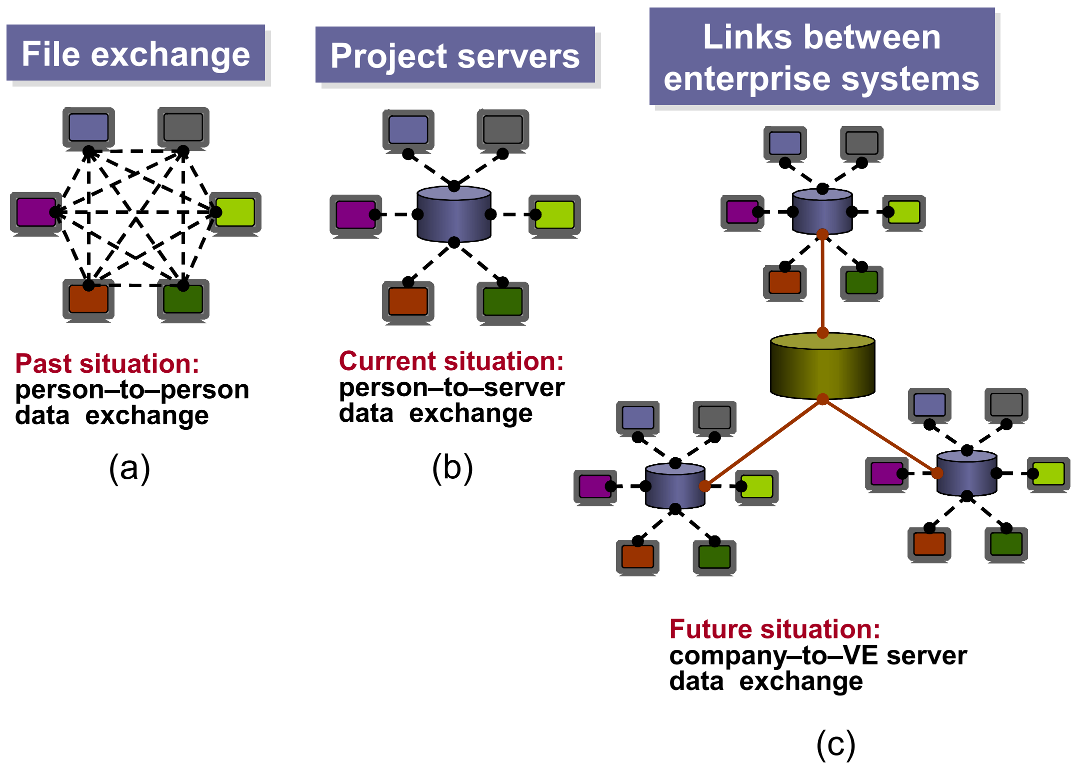

File exchange
Project servers
Links between enterprise systems
Past situation:
person–to–person data exchange
(a)
Current situation:
person–to–server data exchange
(b)
Future situation:
company–to–VE server data exchange
(c)

FIG. 2:Inter-enterprise information exchange in one-of-a-kind settings

3. RESEARCH CONTEXT AND METHODOLOGY

The aim of this section is to introduce the two projects under consideration, GLOBEMEN and OSMOS. An overview of each is provided followed by their respective objectives, description of work and adopted methodology. Both projects have identified a list of desired functionalities, supporting architectures, systems components, and developed both prototypes and in-use applications. In both cases, the focus has been on inter-enterprise information exchange in VE settings.

This section, mainly presents the context and approach of GLOBEMEN and OSMOS. Within section 4 a consolidated perspective of the requirements from both projects is presented. In section 5, based on the findings from these projects, and the consolidated requirements, an ICT architecture, system components, and various interaction scenarios (how the VE environment would interact with other applications) are discussed. Within section 6, two cases (one from each project) are presented in addition to some key lessons learned.

3.1 The GLOBEMEN Project

Global Engineering and Manufacturing in Enterprise Networks, GLOBEMEN (2003) was a three-year project consuming approximately 1000 person-months of effort. It was initiated to both define and harmonise ICT support requirements in various one-of-a-kind industries operating in various cultural environments. The main result of the project is a generic reference model for virtual manufacturing enterprises (VME). This was based on an exploration of the following three domains:

- Sales and services
- Inter-enterprise delivery process management
- Distributed engineering

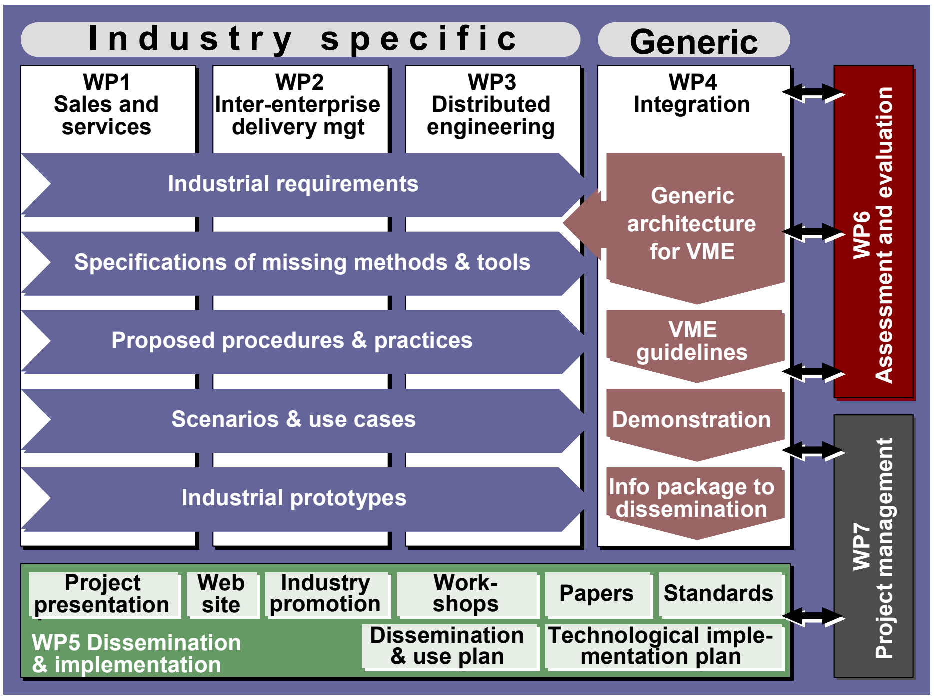
3.1.1 Objectives

The main objectives of GLOBEMEN were as follows:

- Definition of a reference architecture for virtual manufacturing enterprises
- Implementation of proof of concept industrial prototypes
- Demonstration of core features of the architecture
- Promotion of deployment by IT vendors, manufacturing industry, academia and standardisation

3.1.2 Description of Work & Methodology

The description of work of GLOBEMEN is presented in FIG. 3. A common approach was used for each domain considered (sales and services; inter-enterprise delivery process management; distributed engineering), a common approach was used. This was based on the identification of industrial requirements, specification of tools, procedures and practices (best cases), scenarios and use cases, and finally industrial prototypes. The results from these three domains were then consolidated to identify a generic architecture and some generic guidelines for virtual manufacturing enterprises (VME).



Industry specific
Generic
WP1 Sales and services
WP2 Inter-enterprise delivery mgt
WP3 Distributed engineering
WP4 Integration
Industrial requirements
Specifications of missing methods & tools
Proposed procedures & practices
Scenarios & use cases
Industrial prototypes
Generic architecture for VME
VME guidelines
Demonstration
Info package to dissemination
WP6 Assessment and evaluation
WP7 Project management
Project presentation
Web site
Industry promotion
Work-shops
Papers
Standards
WP5 Dissemination & implementation
Dissemination & use plan
Technological imple-mentation plan

FIG. 3: GLOBEMEN: Description of work

One of the major issues in GLOBEMEN was the identification of inter-enterprise interfaces (mechanisms through which enterprise systems could exchange data) and applications. The challenge was to capture the requirements from various domains, consolidate them, and then present them in a meaningful form for software developers to build corresponding interfaces and applications. The approach used in GLOBEMEN and advocated by Kazi and Hannus (2000) was based on a combination of different modelling methods; GERAM (GERAM, 1998), IDEF0 (NIST, 1993), and UML (OMG, 1999). A summary of the approach is presented below and illustrated in FIG. 4.

1. The starting point is the identification of the different enterprise, network, virtual enterprise, or product life cycle stages of relevance and their mapping onto the GERAM lifecycle views (extended version in GLOBEMEN).
2. For each of the relevant lifecycle stages identified in the previous step, the main (target) process is identified and modelled at a high level of abstraction using IDEF0.

3. Each process modelled in the IDEF0s from the previous step is discussed in detail with end-users for validation. Once validated, each process is detailed as a set of use cases (It is worthy of mention, that in most cases, these use cases were textual descriptions provided by the end-users).
4. Through a noun-phrase analysis on the use cases, core system modules are identified.
5. One of the main elements of inter-enterprise systems is the interaction norms of different system modules, i.e. the flow of information (message exchanges) from one to the next, etc. These are defined using sequence diagrams.
6. The results from the previous step are now translated to pseudocode to represent the different required method calls. These method calls form a basis for the definition of the API (Application Program Interface). The API is to serve as the main channel for inter-enterprise interfaces and applications to share and exchange information and related services.

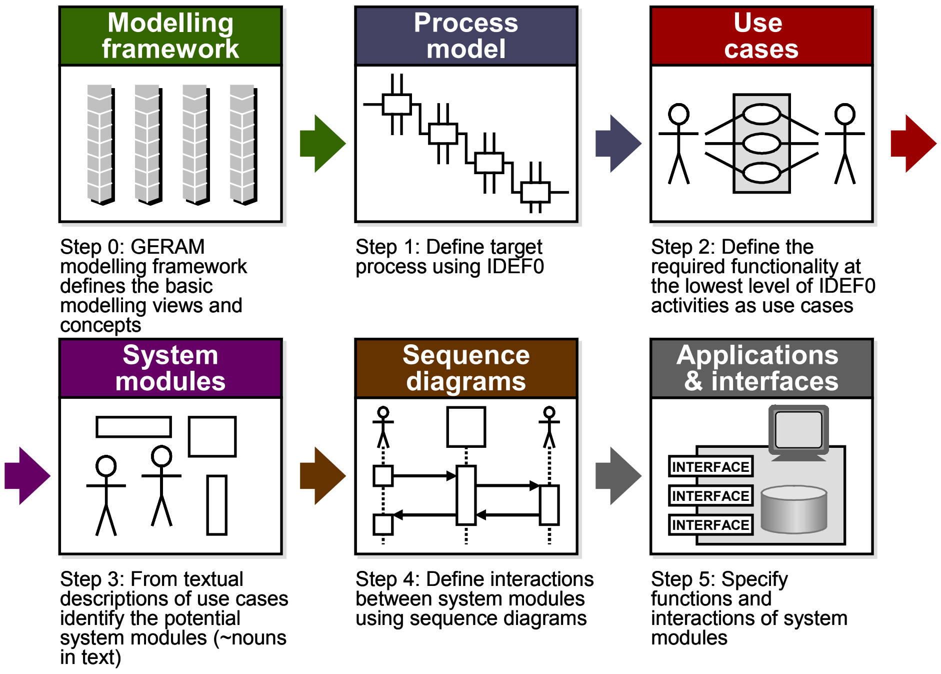

Modelling framework
Step 0: GERAM modelling framework defines the basic modelling views and concepts
Process model
Step 1: Define target process using IDEF0
Use cases
Step 2: Define the required functionality at the lowest level of IDEF0 activities as use cases
System modules
Step 3: From textual descriptions of use cases identify the potential system modules (~nouns in text)
Sequence diagrams
Step 4: Define interactions between system modules using sequence diagrams
Applications & interfaces
INTERFACE
INTERFACE
INTERFACE
Step 5: Specify functions and interactions of system modules

FIG. 4: GLOBEMEN: Modelling methodology

3.2 The OSMOS Project

Open System for Inter-enterprise Information Management in Dynamic Virtual Environments, OSMOS (2002), was a 27 month project consuming 236 person-months of effort. Its main aims were to:

- Enhance the capabilities of construction enterprises, including SMEs, to act and collaborate effectively on projects, and
- Specify, set-up and promote model-driven, value-added, Internet-based flexible services that support team work.

3.2.1 Objectives

The main objectives of OSMOS were to:

- Specify a model-based Internet platform with the following services and characteristics:
    - Team work services
    - Information management services
    - Tools must be cheap and user-friendly to allow SMEs to act and participate in VE
    - Tools must allow end-users to use their proprietary and commercial applications.

- Set up two OSMOS prototype services patterned after these specifications and ensure their take-up, as commercial offerings, after the completion of the project.
- Define the migration path to OSMOS and analyse the likely benefits of its use.

3.2.2 Description of Work & Methodology

The work in OSMOS was based on an iterative refinement approach. Starting from teamwork and requirements analysis, a supporting architecture was specified (in the form of models and API) and a supporting infrastructure was implemented. This was then subject to testing and evaluation by end-users. Overall, three iterations were conducted within OSMOS. Within the second and third iteration, the development was based the results of testing and evaluation by the end-users.

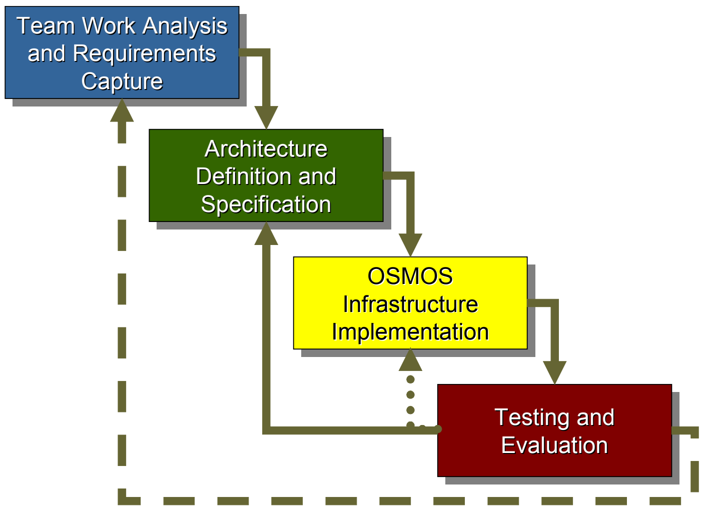

Team Work Analysis and Requirements Capture
Architecture Definition and Specification
OSMOS Infrastructure Implementation
Testing and Evaluation

FIG. 5: OSMOS: Description of work

As compared to GLOBEMEN, a much more detailed analysis was conducted within OSMOS for the construction industry. Whereas GLOBEMEN covered all aspects of the VE as shown in FIG. 1, OSMOS focussed on the VE itself and provision of relevant services for the construction industry.

The main modelling methodology used in OSMOS is shown in FIG. 6. Detailed analysis of the end-user firms was conducted both with regard to their practices within VEs, and within the enterprises themselves. This was deemed necessary so that the developed solution would be as non-intrusive as possible with regard to internal methods of work (while the tools may vary when participating in VEs, internal work practices may remain the same). The analysis also included a detailed investigation of current end-user applications to understand how they may be potentially extended and as to what interfaces would be necessary for them to share information and even services with other applications or a central VE environment in an inter-enterprise setting. As an output of the analysis, a generic VE process model was developed and later broken down to constitute one or more use cases for each identified process (from set-up of the VE to its de-commissioning). The use cases served as the main link (development of conceptual models) between the requirements and the specification of the solution. They, along with the analysis of end-user applications were furthermore used to identify the potential deployment infrastructure. Using UML (OMG, 1999), the conceptual models were used to define the detailed system models, API and underlying core services that the system would offer. Different system tools were developed and end-user applications extended. A set of field trials and evaluations was then used to validate both the generic VE process model and the API. The process was repeated in three iterations. Harvey et al, (2001) and Wilson et al, (2001) provide a more detailed presentation of both the modelling methodology and the consequent field trials.

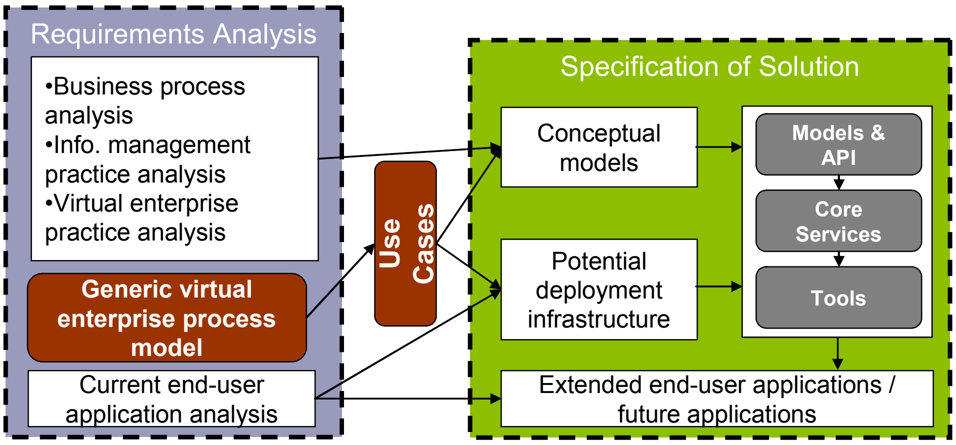

Requirements Analysis
•Business process analysis
•Info. management practice analysis
•Virtual enterprise practice analysis
Generic virtual enterprise process model
Current end-user application analysis
Use Cases
Specification of Solution
Conceptual models
Potential deployment infrastructure
Models & API
Core Services
Tools
Extended end-user applications / future applications

FIG. 6: OSMOS: Modelling methodology

4. REQUIREMENTS FROM IT SOLUTION

ICT requirements may significantly vary from one VE to the next, yet certain commonalities do exist (Harvey et al, 2001, Wilson et al, 2001, Kazi and Hannus, 2002, and Kazi and Hannus, 2003). Within this section, some of these common requirements (as identified within GLOBEMEN and OSMOS) are presented. These requirements are mainly for the shared VE environment.

4.1 Functional Requirements

- Quick set-up: Once a VE is initiated, it is necessary to have in place as quickly as possible and as easy to configure as possible an ICT environment for the VE. This should preferably take a matter of hours and at most some days. Where possible, configuration of the environment could be on the basis of pre-defined templates from previous projects.
- Reliance on standards: Reliance on standards can never be over-emphasised. Standards enable the true sharing of data between not only different applications, but also different organisations. Where proprietary systems do not provide data structured in accordance with current standards (e.g. the Industry Foundation Classes by the International Alliance for Interoperability), interfaces/components need to be provided that provide a translation/mapping mechanism.
- Ease of use: The VE environment must be easy to use. This is necessary, as there is little, if any, time to train the users of the environment (many coming from different organisations). Where possible, users should be in the position to be able to customise the interface to their own personal needs and requirements.
- Integration with legacy tools: Individuals use their own legacy tools for data creation, analysis and decision-making. They use the VE environment mainly to share common data and services. Integration with legacy tools (through appropriate interfaces) can enable users to automatically send relevant data to the shared VE environment rather than having to upload/download data from the VE environment. This furthermore makes the VE environment as non-intrusive as possible to the individual. The individual could then seamlessly interact with the VE environment through his/her own legacy application(s), rather than the other way around. This is an important need, especially when individuals are involved in multiple VEs at a given time.
- Security: Security control is of paramount importance in VE environments, especially when data flows occur between partners who may not necessarily be covered by legal contracts. Transactions need to be carefully monitored and access rights efficiently moderated. It is advocated that during configuration of a VE environment, access rights are not configured at an individual level, but at a participant organisation level. Thereafter, the organisation (or its authorised representative) would delegate all/some of its rights to individuals within the organisation. This supports the concept that individuals may change not only from project to project, but also from project phase to project phase.

- Information and transaction archiving: Once a project is completed, it is necessary to access and trace all transactions and data from the project. Bearing in mind that once the project is over, only the organisation or ISP hosting the VE environment would have access to it (based on typical licensing conditions). Mechanisms need to be provided for other participants to be able to browse the transactions and data too. The simplest mechanism to do this is to do the archiving on a CD that can be browsed and accessed through a simple browser (e.g. Windows Explorer).

4.2 Architectural Requirements

- Platform independence: Most off-the-shelf solutions today operate on a given platform and typically persistence is limited to one main (or a few) database types. This in itself leads to problems on many occasions. The preferred solution would be for a solution (both client and server) to be platform (and as far as possible, hardware) independent. Most, if not all, databases should be supported (most are configurable through a standard SQL initialisation script), and no stored procedures should be used. It is these stored procedures that typically force the use of a particular database.
- Distribution support: The system should by default, provide a set of basic services to support the registration, invocation, and transaction monitoring of different concurrent distributed systems and their exposed services. This should include, e.g. lookup facilities (naming and/or trader), transactions, control of concurrency, remote method invocation, etc.
- Communication flexibility: The system should be as flexible as possible to be in a position to support new types of messages and data flows. It should automatically (or at least with very simple ease) update its generic repositories to support the administration, control and transmission of these messages.
- Communication protocols: The system architecture should provide support for various communication protocols and not rely on a single one alone, e.g. TCP/IP (Transmission Control Protocol/Internet Protocol). TCP/IP though, may be the default and preferred option.
- Communication types: The system must support different types of communication. In simple terms, it should allow for both synchronous and asynchronous communication. Referring to the discussion presented under section 2.2, in many cases, enterprise systems would communicate with the central VE repository. This communication, while in some cases could be real-time (synchronous), in others would be periodic (asynchronous), e.g. twice a day when bulk uploads/downloads are done.
- Code portability: The code used should be portable. It should be possible for it to be edited (extended), recompiled and run on top of another environment without any change in the main code.
- Security hierarchies: The system should support multiple levels of security. At a minimum, it should provide services for identification/authentication and authorisation/access control. Other services to be offered should include confidentiality and message integrity, audit control and non-repudiation.
- Legacy application integration: The system architecture should provide a standard framework for the integration of legacy applications and services through both invasive and preferably non-invasive application adapters.

4.3 Technical Requirements

- Handling of any data types or specific types: The environment should have the ability to access/exchange information that is of any “strong” type (integers, floats, Boolean, arrays, unions, etc.). Where this information is in the form of data flows within strings, the environment should be able to manipulate these and then do appropriate mappings to relevant types.
- Object oriented model handling: The environment should be able to allow the integration and management of the application object model (potentially at any level of object granularity) without

any specific adapter layer. This should constitute the capability of the environment to deal with various application interfaces and invocation semantics.

- Dynamic service integration and method invocation: The environment should be able to dynamically integrate "external" services and present mechanisms for method invocation on these services. At the same time, it should provide mechanisms for these external services to invoke methods on the VE environment itself. Service integration and method invocation should be possible at runtime.
- Language independence: It should be possible for the client/server to be implemented using any language. As far as possible, a neutral or "language independent" API should be provided. Where this is not possible, middleware components that facilitate translation of method calls in one language (e.g. COM objects from legacy applications) to others (e.g. Java objects of the VE environment) should be provided. The system should consequently allow requests and/or method invocations no matter the requesting language and the implementation language.
- Security: The environment should possess a capability for dealing with firewalls. This should include mechanisms to ensure the protection of the data/information that is exchanged through various communication protocols.

5. ICT ARCHITECTURE, COMPONENTS AND INTERACTION SCENARIOS

The aim of this section is to specify the ICT architecture and system components of a potential solution that will satisfy the requirements presented in section 4. Some scenarios illustrating various forms of interaction between the VE environment and external services/applications are also presented. A subset of these scenarios will be demonstrated in the discussion of results (section 6).

5.1 ICT Architecture

A simple seven layered ICT architecture (FIG. 7) was defined by Kazi et al, (2001), and Kazi and Hannus (2002). Its main purpose was to act as a mapping template upon which organisations could map their in-house applications, interfaces to shared VE environments, communication protocols, and shared services used. This architecture was validated and used by all end-users in GLOBEMEN.

1. *Presentation layer:* This layer provides the user interface through which individuals gain access to VE information depending on their roles in the VE partner organisation. (Example: web page providing access to the projects in which a person participates). This layer is (in an ideal case) configured by the organisation which the individual represents.
2. *Application layer:* This layer consists of applications that a user needs to perform tasks for specific VEs. It essentially includes application software of the VE partner organisation. (Examples: ERP, CAD, cost estimation, etc.). This layer is generally quite static and independent of a specific VE.
3. *Interoperability layer:* This layer acts as the data/information mapping and translation mechanisms between an organisation's applications and the shared VE environment. It may provide semantic and syntactic mapping to VE standards, release management, assigning VE level access rights internally within the partner organisation, etc. (Examples: conversion software that translates proprietary data format to a standard format, making officially released information available to the VE). This layer usually needs to be configured for each specific VE.
4. *Communication layer:* This layer enables communication between (the ICT systems of) an organisation and the shared workspace of the VE. Thereby, this layer addresses both the geographic and organisational distribution of VE partners. (Example: the Internet, data exchange and communication protocols, data formats etc.). This layer relies mostly on standards and commonly available technologies.
5. *Access layer:* This layer controls the access to the shared VE workspace and information. It provides/regulates management of VE member roles and profiles. (Example: user identification, access rights management).

6. *Service layer:* This layer provides access to and management of shared information and services to the VE. (Examples: document management, product model management, inter-enterprise workflow management, notification of new/changed information etc.).

7. *Storage layer:* This layer hosts the main system registry and repository. (Examples: databases, backup procedures, etc.).

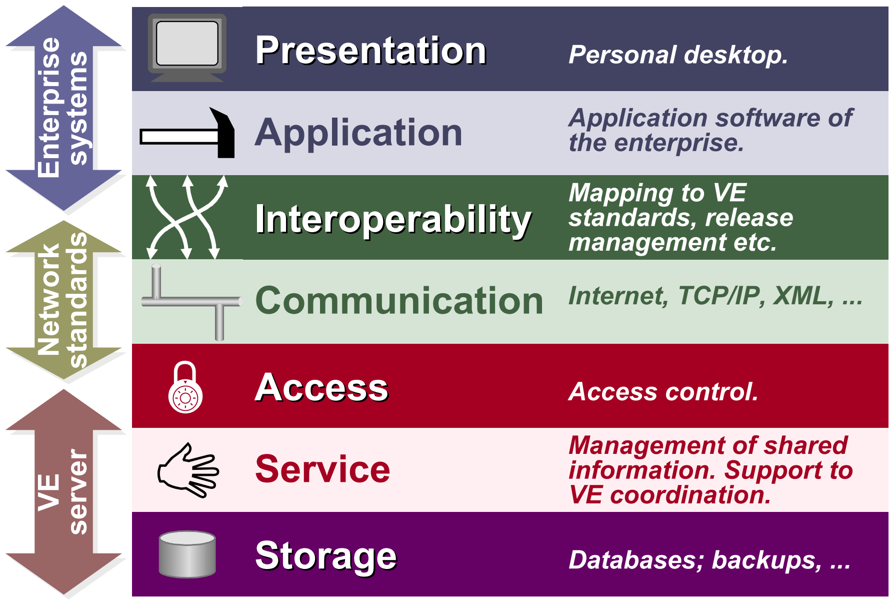

Enterprise systems
Network standards
VE server
Presentation
Personal desktop.
Application
Application software of the enterprise.
Interoperability
Mapping to VE standards, release management etc.
Communication
Internet, TCP/IP, XML, ...
Access
Access control.
Service
Management of shared information. Support to VE coordination.
Storage
Databases; backups, ...

FIG. 7: Seven layered ICT architecture

When mapping onto this architecture (shown in FIG. 7) it should noted that certain elements (applications/services) may span across more than one layer (see example in FIG. 14).

5.2 System Components

The envisioned environment for the VE (FIG. 8), contains several distinct components. A short description of the components and their mapping onto the seven layered architecture (FIG. 7) follows:

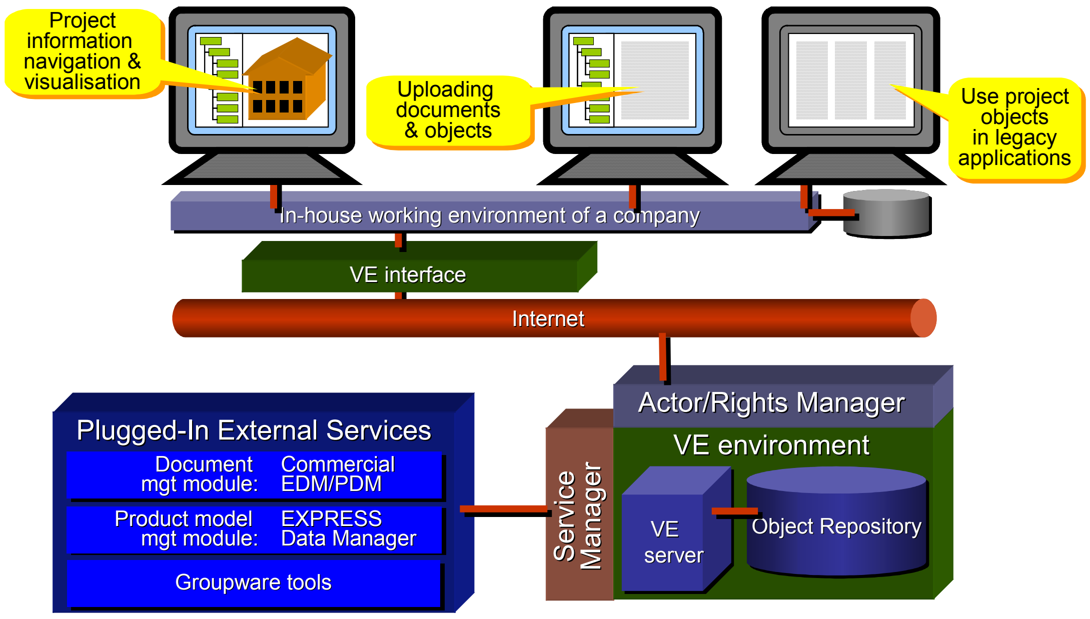

Project information navigation & visualisation
Uploading documents & objects
Use project objects in legacy applications
In-house working environment of a company
VE interface
Internet
Actor/Rights Manager
VE environment
VE server
Object Repository
Service Manager
Plugged-In External Services
Document mgt module:
Commercial EDM/PDM
Product model mgt module:
EXPRESS Data Manager
Groupware tools

FIG. 8: System components

- *Personal Desktop:* This is the main interface through which an individual interacts with both his/her organisation's in-house working environment and other sources (e.g. the VE). Some applications may reside on the desktop itself, whereas others may be accessible in the form of online services. The personal desktop maps well onto the "presentation layer" of the seven-layered architecture. Of course, certain applications on the desktop may also extend to other layers, e.g. application and/or interoperability layers.
- *In*-house *working environment of a company:* This is where most of the internal applications and services of the organisation reside. The corporate intranet, would be one such example. This component, maps mainly onto the "application layer" of the seven-layered architecture. It should be noted that "within" the enterprise this component could also act as a "local" VE environment. In such cases its mapping onto the whole seven layered architecture would be justified.
- *VE interface:* This component mainly acts as a mapping agent/translator between an organisation and the shared VE environment that it uses when participating in a VE. It provides for release management and regulates what information is sent by the organisation to the shared VE environment. The VE interface may be mapped onto the "interoperability layer" in the seven-layered architecture.
- *Internet:* The Internet here acts as the main communications channel between the enterprise and the shared VE environment and associated services. It may be mapped onto the "communication layer" of the seven-layered architecture.
- *Actor/rights manager:* This component deals with basic access control and provides access to those services that an individual (or organisation) have rights. It can be mapped onto the access control layer within the seven-layered architecture.
- *VE environment:* This contains the mechanisms to manage access control, plug-in services, management of shared information, etc. In many cases, this acts as the shared "project server" that many are familiar with. The VE environment maps onto the "Access", "Service", and "Storage" layers of the seven-layered architecture. It makes available and delivers the functionalities identified earlier in section 4.
- *Service manager:* This component handles the registration of, access to, and management of external services. It can be mapped onto the service layer of the seven-layered architecture.
- *Plugged-in external services:* The IT needs may vary from one VE to the next. This variation is primarily in terms of the shared services that are required for the VE in question. In most cases, they do not need to be developed from scratch, but may be plugged-in. In some cases however, adaptors may be required to enable the "plugging-in" of some specific external service. Some examples of such services are indicated in FIG. 8. This system component (plugged-in external services) may be mapped onto the "Service" and "Storage" layers of the seven-layered architecture. In most cases, access to these services will be managed by the service manager and regulated through the VE environment.

5.3 Interaction Scenarios

One interesting aspect of VE environments are the norms through which interactions take place with other applications/services that are plugged-into the VE environment. Several such interaction scenarios are presented in this section. Note that the scenarios are not exhaustive and hence only attempt to illustrate some possibilities. Different variations of these are, of course, possible. (FIG. 9 should be used as a key or legend to understanding the scenarios presented).

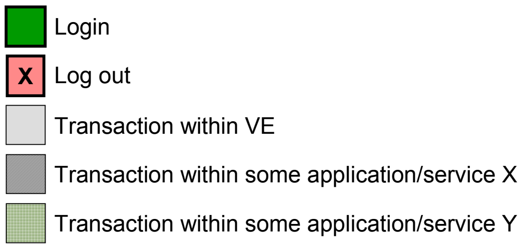

Login
X
Log out
Transaction within VE
Transaction within some application/service X
Transaction within some application/service Y

FIG. 9: Key to understanding scenarios

5.3.1 Scenario 1: VE environment passes a dynamic query to an external site

A user would log on to the VE environment and do some work. Based on the work he/she is doing, the VE environment can offer additional sources for more information. As an example, if the user was looking for a particular type of "beam", the VE environment could configure a dynamic query (containing the relevant keywords, e.g. beam) and pass it to the relevant information source/service. The user would be logged out of the system and directed to the service to which the query was sent. In most cases, though not necessarily, the VE environment would present some logon credentials for the user to access the requested service. There is however, no feedback from the requested service to the VE environment.

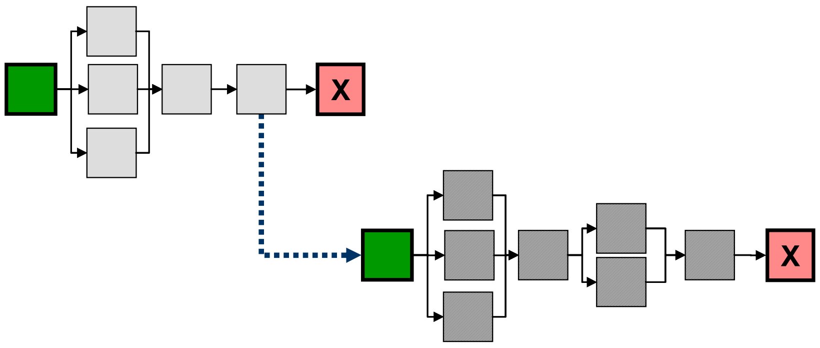


FIG. 10: Scenario 1: VE environment passes a dynamic query to an external site

5.3.2 Scenario 2: Application/Service calls VE environment

In this scenario, a user would log on to his/her in-house working environment. Assuming the user needs a particular piece of information, the environment would pass on logon credentials and the request to the VE environment, get the requested information, and then present it to the user. The user would continue to work normally and when done, log out from his/her in-house environment. It should be noted that in this case, the interaction and information exchange that took place between the in-house working environment and the VE environment is transparent to the user (he/she may not even know that it took place). This scenario corresponds to "links between enterprise systems" that was discussed in section 2.2 and illustrated in FIG. 2 (c).

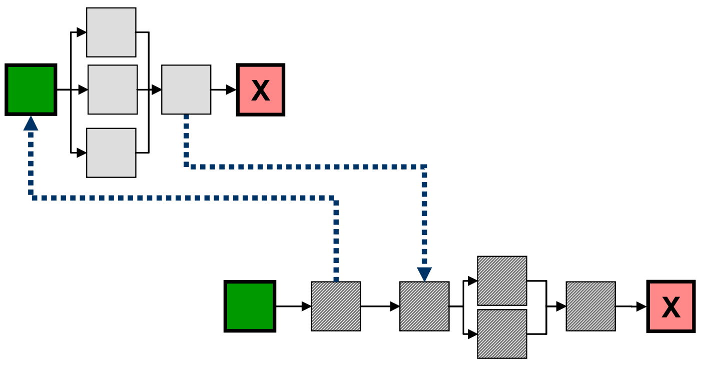


FIG. 11: Scenario 2: Application/Service calls VE environment

5.3.3 Scenario 3: VE environment retrieves information from external application/service

This scenario is very much similar to the previous one, only it is in the reverse direction. A user logs on to the VE environment and for example wishes to upload a new document. The VE environment itself does not contain a document management system, but one is plugged-in to it. As such, upon the user's request, the VE environment would logon to the document management system, upload the document, and in return, capture the "id" of the document. This captured "id" could then be used later on for example, when the need arises to retrieve the document.

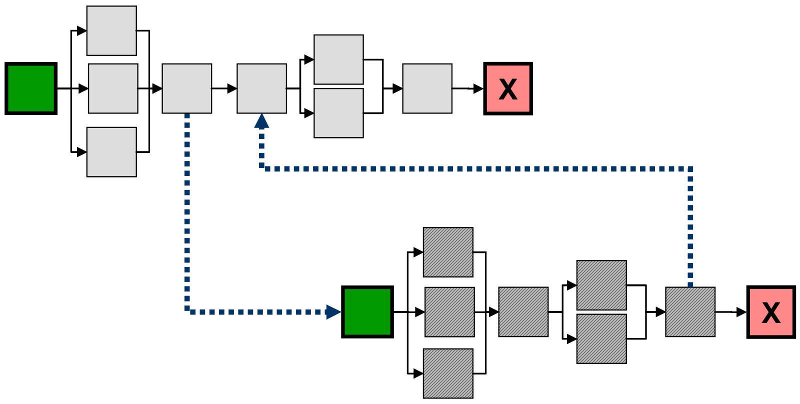


FIG. 12: Scenario 3: VE environment retrieves information from external application/service

5.3.4 Scenario 4: VE environment invokes several external applications/services

In this scenario, service-to-service (application-to-application) interaction through the VE environment is demonstrated. A user logs on to the VE environment and then wishes to perform some actions related to a registered "object" (e.g. a product model of a building), which in fact points to an object located elsewhere (e.g. product model server). The user is sent to the object "service provider" (product model server). The VE environment passes some credentials and requests to the service provider, which then makes available to the user some specific "allowed" methods provided by the service (e.g. extract partial model, edit object, etc.). For this example, we'll assume that a user would like to modify the details of a given beam. Once the details are updated, the user would be returned to the VE environment. At the same time, the service/application could send a message to the VE environment informing that a user has not only browsed the requested object, but that an updated version of the same was created. The VE environment would, in turn receive and register this information within itself. The user continues on, and wishes to upload a document that contains some information related to the beam he/she just updated. The VE environment consequently sends the user to another service/application (e.g. a document management system), which offers document management facilities. As in scenario 4, the document is uploaded and confirmation of the same ("id" of document) sent back to the VE environment. This time, the VE environment not only creates a new information object describing this new document, but furthermore creates a relationship between this document (object) and the new version of the beam (object) to which the document was attached (related). In this scenario, in addition to what was in previous scenarios is the fact that there is two-way communication between the VE environment and two mutually independent services/applications. It was possible to associate an object held in one service/application with an object held in another application/service. Last but not least, the VE environment has an idea of "what happened", i.e. what the user did with the applications/services.

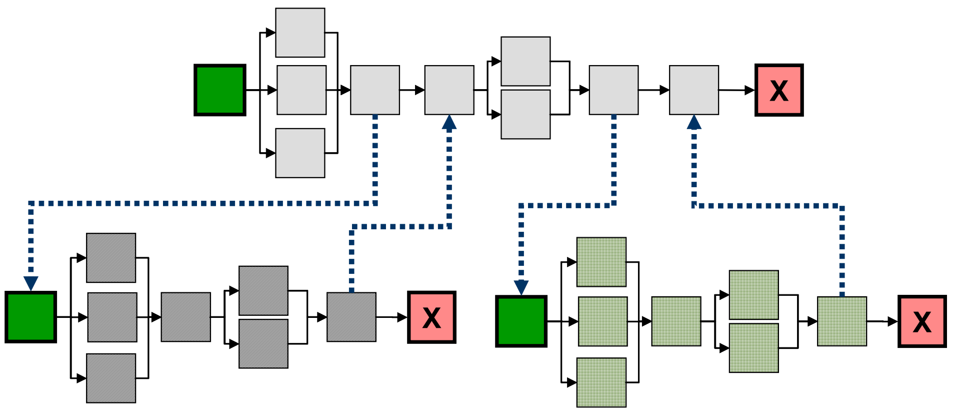


FIG. 13: Scenario 4: VE environment invokes several external applications/services

6. DISCUSSION OF RESULTS

GLOBEMEN and OSMOS yielded many results in the form of technical specifications, models, prototypes and commercial offerings to enable VEs in the construction and related industries. A detailed discussion of these is out of the scope of this paper and the reader is referred to GLOBEMEN (2003) and OSMOS (2002) for more information. In this section however, a brief outline of two cases is presented: distributed engineering in construction (from GLOBEMEN) and facilities management (from OSMOS) is presented. Some key lessons learned are also presented thereafter.

6.1 Case: Distributed Engineering in Construction

6.1.1 The problem

The main problem was characterised as follows:

- Most building drawings were done using 2D CAD by architects.
- Most of the design and project planning and control had to be done on a case by case basis and at times manually.
- Updating also led at times to design errors as it was difficult to ascertain what, when, where and by whom changes were made.

To maintain some form of consistency, the contractor (YIT construction) transformed the drawings to a product model. Where possible, the semantic richness of the product model was exploited and certain tasks in the process automated. However, whenever there was a change in the drawings, the product model had to be updated manually. The product model was not necessarily shared as it contained confidential information belonging to the contractor. There existed no easy means to make available only relevant portions (partial models) of the model. Last, but not least, when a project ended, no as-built model was created.

6.1.2 The solution

The solution was based on shared product model made available through a product model server. The product model server provided support for partial models. In the VE environment that was set-up, in addition to access control and transaction monitoring, two main services were provided: product model management, and document management. The architecture from the contractor's viewpoint, is illustrated in FIG. 14. Note, from other viewpoints, e.g. architect, the top three layers of the architecture (FIG. 14) may look different, though the bottom four will be the same as for the contractor.

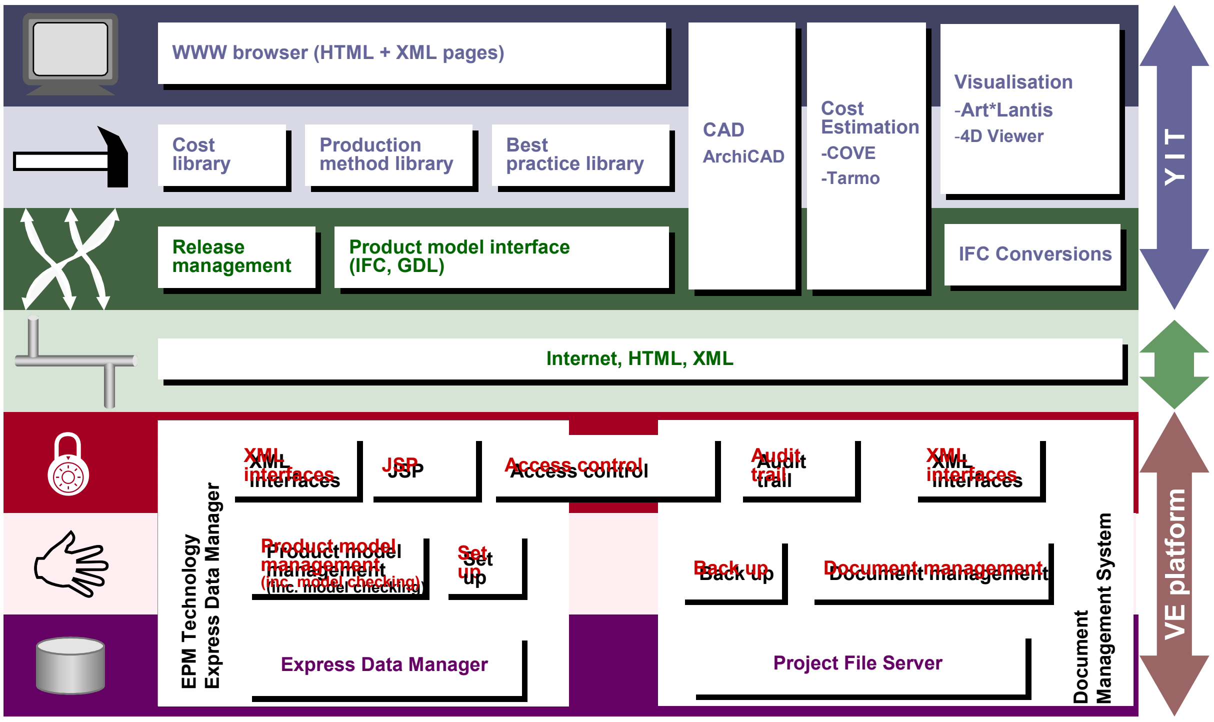


FIG. 14: YIT's mapping on to the seven layered architecture

The solution is best understood through a real scenario:

- The contractor makes available some preferred structural types in the form of pre-configured templates.
- The architect, where feasible, utilises these templates in the design and exports the results as a product model.
- The product model is uploaded into the product model server (made available through the VE environment). Other project participants (e.g. client, structural engineer, HVAC engineer, contractor, supplier, etc.) may now access the model in its entirety of in parts.
- Other participants download the product model (or parts of it that concern them) and import the results into their proprietary applications.
- Each participant works on the product model, enriches it with information, and uploads it (or relevant parts of it) to the product model server. It is to be noted that, only those parts of the product model may be exported that are either owned or controlled by the particular participant.
- All participants can now access and view the "merged" product model in its entirety or in parts. The merging is based on partial models uploaded by the various participants.
- The process continues in an iterative manner during the course of the project.
- Once the project ends, the product model is saved as the "as-built" model and enriched with relevant information (or pointers to information).

It is worth mentioning that access to the product model server and document management system was handled through the shared VE environment. Participants were able to use a single-login to access these systems. Furthermore, links between objects in the product model and documents stored in the document management system were maintained in the shared VE environment.

The interactions within this case are similar to interaction scenarios 2 and 3 identified in section 5.3.

6.1.3 The benefits

Several benefits were achieved:

- All sketches were available through the product model and in most cases, importable to the applications in use by the various participants.
- It was possible to navigate the product model in 3D. The 3D rendition was done on the fly.
- It was possible to automate certain design and project planning and control tasks (e.g. cost estimation).
- All project participants shared information through a shared and consistent source, the product model.
- It was easy to identify design changes and to know as to what, where, when, and by whom the changes were made.
- The semantic richness of the product model was exploitable in the form of automatically generated reports and graphs.
- Once the project ended, it was possible to save it as the "as-built" product model. Which was then usable for facilities management during the building life-cycle.

6.2 Case: Facilities Management

6.2.1 The problem

A state owned enterprise manages in excess of 2500 state facilities. Of these, the details of more than 2000 are stored in a facilities management solution (*Ryhti* software from Granlund, Finland) for facilities management purposes. The enterprise however has limited personnel (200) and utilises the service of other business partners for the management of its facilities. A recent addition has been a document management server that provides access to various documents related to the managed facilities. However configuring the document management server with access rights and controls for all of its business partners is becoming a tedious task. Furthermore, there is no mapping between the facilities management tool and the document management server.

6.2.2 The solution

The solution was built around the new VE tools developed by Granlund within the context of the OSMOS project. Of these, the *GranlundWeb* browser is a web-based tool that acts as an entry point to a VE environment. It is designed to handle requests and responses by *Ryhti* and other external "plugged-in" tools. The Granlund *VeManager* on the other hand is an administrative tool used for the basic instantiation and configuration of a VE, its participants, and used services.

The following scenario (FIG. 15) describes the solution (step 0 relates to system configuration, whereas the rest relate to operation in the VE setting):

1. Initially, the system is configured in terms of actors, roles, access rights, projects, available services and service methods, etc. using the Granlund *VeManager*. The configuration details are stored in the VE environment (registry).
2. The user logs on to the system through the *GranlundWeb* browser. This sends a login request to the VE environment.
3. The VE environment checks the login request and if valid, performs a role to project and available service mapping. The outcome of this mapping is then forwarded to the facility management server.
4. Based on the information it receives from the VE environment, the facilities management server presents to the end-user the projects' facility data and services to which he/she has access rights.
5. A user may invoke certain facilities management services or request a document related to a particular building object. The document request is in the form of sending a Building ID to the document management server.
6. The document management server cannot directly act on a Building ID alone and needs more information. To get this information, a request is made to the VE environment in the form of an API call.

7. The VE environment receives the request from the document management server and checks the credentials of the requesting user. Once the credentials have been verified, the VE environment sends access verification in addition to the Document ID for the document to which the user has access to back to the document management server.

8. The document management server releases the document to the user.

In this solution, all transactions took place through a single user interface. Furthermore, access rights based on predefined roles were defined and accessible through the VE environment and not the document management system.

The interactions within this case are similar to interaction scenario 4 identified in section 5.3.

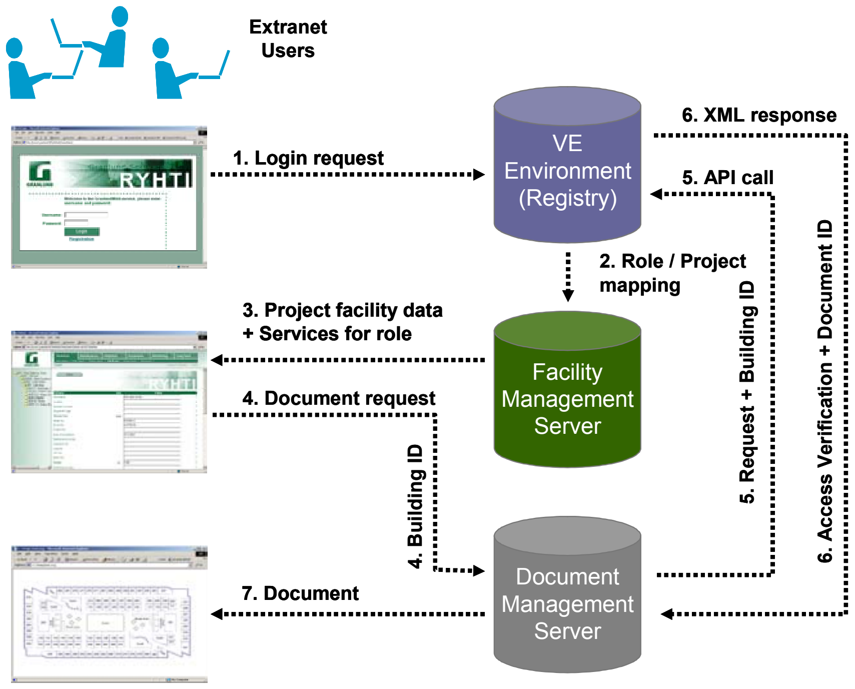

Extranet Users
1. Login request
VE Environment (Registry)
6. XML response
5. API call
2. Role / Project mapping
3. Project facility data + Services for role
Facility Management Server
4. Document request
4. Building ID
5. Request + Building ID
6. Access Verification + Document ID
Document Management Server
7. Document
RYHTI

FIG. 15: Facilities management case study scenario

6.2.3 The benefits

Within this solution, several benefits were realised:

- There was single log on and uniform user interface. As far as the user was concerned he/she was using a single application.
- It was possible to have personal user profiles.
- There was no overlapping of data. Documents were stored in the document management system, and building objects in the facilities management server. Links between documents and objects were stored in the VE environment.
- There was centralised definition of access rights and roles.
- It was possible to have data stored on different servers and in different applications. The VE environment acted as the central access point.
- Secure access was possible.

6.3 Lessons Learned

Several lessons were learned during the implementation, deployment, and use of the solutions developed with both GLOBEMEN and OSMOS. Some have already been covered in the form of requirements under section 4. Others have been solicited from the end-users of the developed solutions:

- Most end-users prefer Internet enabled applications. Many are familiar with the Internet and use of web-based applications. Furthermore, the menus in such applications are less complex than those on desktop versions. Where possible, the user should not have to go through more than *3 clicks* to get to the information he/she is seeking.
- There is a need to move from standardised applications to personal workspaces. The user can configure these environments (and user interfaces) based on individual needs and preferences.
- Data and information exchange should as far as possible be in a form that is transparent to the user. The user need not specify in each case from where exactly to retrieve the information.
- There is a need to ensure compliance with used standards (e.g. the Industry Foundation Classes). Where compliance is not possible (e.g. different countries use different building standards),interfaces (mapping/translation) to these standards should be provided.
- Inter-enterprise collaboration platforms should be set-up quickly and efficiently. Where possible, configuration should be done through pre-configure templates.
- Means should be provided for the capture, retention and further propagation of knowledge. Information should be accessible across projects.
- Solutions should provide support for the total life cycle management of facilities, and not just one phase (e.g. design).

Several barriers restraining the take-up and use of advanced ICTs were identified. The main barriers were:

- Organisational Barriers: lack of business incentives, poor IT strategies, lack of training, lack of appropriate ICT support
- Legal Barriers: risks for liability, lack of legal support for use of ICT, security of ICT transactions, Intellectual Property Right (IPR) issues for electronic information and documentation
- People Barriers: lack of personal incentives, lack of education/training/continuous professional development, cultural issues, reluctance to changes in business processes
- Technology Barriers: difficulties in using new technologies, lack of support from ICT providers (or IT department), incompatibility/interoperability problems: lack of (use of) standards

7. CONCLUSIONS

Client demands for one-of-a-kind-products and services are fostering competency sharing and collaboration between different organisations to deliver these products and services. Each organisation is typically involved in the delivery of one or a few components of the requested product or service. To deliver the complete product or service, organisations need to rely on each other for information completeness, as all product components are inter-related. This is giving rise to demands for inter-enterprise collaboration tools that allow an organisation to transparently share and exchange relevant product/service related information with other organisations.

In response to demands for inter-enterprise collaboration tools, a significant amount of research and development has been done. The research from GLOBEMEN and OSMOS reported within this paper presented just two out of many similar past and ongoing efforts. Despite calls for open and interoperable solutions, most developments are closed in some way or another. If one exchanges one component for another, a significant amount of re-work and programming needs to be done. As an example, the lead players in the construction industry are backing the use of the Industry Foundation Classes (IFCs). For their efficient use in distributed environments and to avoiding data redundancy, there is a need for a product model server. There are a few commercial offerings available, yet, each has its own API. So if in one VE, one product model server is used, and in another VE, another product model server, one encounters certain difficulties. Are product models and product model servers really the correct way forward? Many believe so, though with a word of caution that there still remains much to be done.

Several issues mar the take-up and use of advanced ICT solutions in the construction industry. Many organisations are not willing to take the next step forward, in the belief that once they have taken this step, another new (and costly) one will follow. The hunt and wait for the ideal solution goes on, but with no end currently in sight. Where some organisations do take a bold step forward, there is at times resistance from individuals working in these organisations. Several organisational, legal, human, and/or technological barriers still restrain us from taking the big step forward.

In the past, we had applications that were "built to last". In today's dynamic work environment however, we need to have applications that are "built to change" when necessary. This change should be possible through flexible interfaces and rely on commonly used data exchange standards.

8. ACKNOWLEDGEMENTS

GLOBEMEN (2003), Global Engineering and Manufacturing in Enterprise Networks, is an RTD (Research and Technology Development) project that was been initiated to define and harmonise ICT support requirements in various one-of-a-kind industries operating in various cultural environments. This is an IMS (Intelligent Manufacturing Systems) project involving 19 partners from different IMS regions. The European Commission within Framework V of the IST programme funds EU participation in GLOBEMEN: IST-1999-60002.

OSMOS (2002) is funded by the European Commission within Framework V of the IST programme: IST-1999-10491. The consortium includes Construction IT Service Providers: DERBi, JM, Olof Granlund, and leading European Academic and Research Centres: Centre Scientifique et Technique du Bâtiment (CSTB), the Information Systems Institute of University of Salford, and the Technical Research Centre of Finland (VTT).

The section on "Interaction Scenarios" is part of an ongoing research study at the School of Civil Engineering, Asian Institute of Technology (AIT), Thailand.

9. REFERENCES

Afsarmanesh H., Garita C., Hertzberger L.O. and Santos-Silva V. (1997). Management of Distributed Information in Virtual Enterprises – The Prodnet Approach. *Proceedings of the 4th international conference on concurrent engineering*, Nottingham, UK, 241-255.

Browne J., Sackett P.J. and Wortmann J.C. (1994). The System of Manufacturing: A Perspective Study, Report to the DG XII of the CEC.

Byrne J.A., Brandt R., Port O. and bureau reports (1993). The Virtual Corporation, *Business Week*, February 8. (http://www.businessweek.com/@@J1JaMoUQtnJSthEA/archives/1993/b330454.arc.htm)

Camarinha-Matos L. M., Afsarmanesh H., Garita C. and Lima C. (1998) Towards an Architecture for Virtual Enterprises, *Journal of Intelligent Manufacturing*, Vol. 9, No. 2, 189-199.

Charbuck D. and Young J.S. (1992). The Virtual Workplace, *Forbes*, Vol. 150, No. 12, 184-190.

Filos E. and Banahan E. (2001). Towards The Smart Organization: An Emerging Organizational Paradigm And The Contribution Of The European RTD Programs, *Journal of Intelligent Manufacturing*, Vol.12, No. 2, 101-119

GERAM - IFAC/IFIP Task Force on Architectures for Enterprise Integration. (1999). GERAM: Generalised Enterprise Reference Architecture and Methodology, ISO/DIS15704. (http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html)

GLOBEMEN (2003). Public Website: http://globemen.vtt.fi

Hannus M., and Kazi A.S. (2000). Requirements for Distributed Engineering. *Proceedings of ECPPM 2000: Product and Process Modelling in Building and Construction*, Lisbon, Portugal, 41-78.

Harvey S., Rezgui Y., Zarli A. and Kazi A.S. (2001). Service for inter-enterprise information management in dynamic virtual environments. In Asko Sarja (ed.), *Information and communication technology (ICT) in the practice of building and civil engineering; Proc. of ECCE 2001, Helsinki, 6-8 June 2001.* Helsinki: Association of Finnish Civil Engineers (RIL).

Kazi A.S. and Hannus M. (2000). Functional Requirements for Inter-enterprise Intranet Services. *Global Engineering, Manufacturing and Enterprise Networks* (Mo J.P.T. and Nemes L., editors), Kluwer Academic Publishers, 54-60.

Kazi A.S. and Hannus M. (2002). Virtual Enterprise Reference Architecture and Methodology, *eWork and eBusiness in Architecture, Engineering and Construction* (Turk Z., and Scherer R., editors), Balkema, pp. 43-48.

Kazi A.S., Hannus M., Laitinen J., and Nummelin O. (2001). Distributed engineering in construction: findings from the IMS GLOBEMEN project. *ITcon* 6: 129-148.

Kazi A.S., and Hannus M. (2003). Interaction Mechanisms and Functional Needs for One-of-a-kind Production in Inter-enterprise Settings, *Global Engineering and Manufacturing in Enterprise Networks* (Karvonen I., et al., editors), VTT Symposium Series, 301-312.

NIIIP (1998). Introduction to NIIIP Concepts, Book 0, NIIIP Reference Architecture (http://www.niiip.org/public/home.nsf/fProjectHome?openform&PID=NIIIP_TRP)

NIST (1993). Draft Federal information processing standards publication 183: Announcing the standard for integration definition for function modeling (IDEF0). (http://www.idef.com/downloads/pdf/idef0.pdf)

Object Management Group (OMG). (1999). OMG Unified Modeling Language Specification, version 1.4. http://www.omg.org/technology/documents/formal/uml.htm

OSMOS (2002), Open System for Inter-enterprise Information Management in Dynamic Virtual Enterprises (http://osmos.vtt.fi)

Rabelo R. and Camarinha-Matos L.M. (1996). Towards Agile Scheduling in Extended Enterprise, *Proceedings of BASYS'96: Balanced Automation Systems II- Implementation Challenges for Anthropocentric Manufacturing* (Camarinha-Matos L.M. and Afsarmanesh H., editors), Chapman & Hall, 413-422.

Walton J. and Whicker L. (1996). Virtual Enterprise: Myth and Reality, *Journal of Control*, Vol. 22, No. 8, 22-25.

Wilson I.E., Harvey S., Vankeisbelck R., and Kazi A.S. (2001). OSMOS: Enabling the construction virtual enterprise. *ITcon* 6: 85-110.