FP6 – Priority 2.3.2.9
Improving Risk Management
Integrated Project

Project Acronym	**OASIS**		
Project Full Name	**O**pen **A**dvanced **S**ystem for d**IS**aster and emergency management		
Contract Number	**004677**		
Starting date	**01/09/2004**	Ending date	**31/12/2008**

Deliverable ID	D-SP1_07		
Document Reference	OASIS_SP11_RPT_258_DSF		
Title of the Document	Summary of SP1 (COE) achievements and lessons learnt		
WP related to the document	SP11		
Type	Public		
Partner(s) Contributing	EADS DS		
Date	19/12/2008	Version	1.0

	Name / Function	**Company / Organisation**	**Date**
Managed by :	Martine Couturier Jean-François Gallet	EADS DS	19/12/2008
Checked by:			
Approved by:			
Released by:	Martine Couturier	EADS DS	19/12/2008

Project Coordinator

Company name	EADS DS
Name of representative	Martine COUTURIER
Address	1 Bd Jean Moulin - 78996 Elancourt - France
Phone number	+33 (0)1 61 38 52 52
E-mail	
Project WEB site	http://www.Oasis-fp6.org/

Document Status Sheet

Version	Date	Details
0.0	17/11/08	Creation of the document
1.0	19/12/08	Update of the document after POS2 and the final Oasis event

Table of contents:

1 Introduction 4

2 Reference documents 4

3 COE achievements 5

4 COE lessons learnt 12

4.1 Oasis Service Bus 12

4.2 Use of Web services 12

4.3 User and Application Managers 13

4.4 Geo-Spatial Managers 14

4.4.1 Map quality 14

4.4.2 Creation and edition of information related to the event 14

4.4.3 Provision of a map server 15

4.4.4 Technological solutions 15

4.5 Data Manager 15

4.6 Notification Manager 16

4.7 Logbook 16

4.8 GPS driver 16

4.9 The Event Manager 17

4.10 Forms tool 17

4.11 Navigation 17

4.12 User Directory 18

4.13 Security issues in Oasis 18

5 Conclusion 19

Table of the figures:

Figure 1: Oasis COE functional diagram 7

Figure 2: Web services triumvirate 8

1 INTRODUCTION

The objective of **Oasis** is to define and develop an Information Technology framework based on an open and flexible architecture and using standards, existing or proposed by **Oasis**, which could be the basis of a European Disaster and Emergency Management system.
Oasis is intended to facilitate the cooperation between the information systems used by civil protection organisations, in a local, regional, national or international environment. This Disaster and Emergency Management system aims to support the response operations in the case of large scale as well as local emergencies.
Oasis provides within this framework an initial set of applications which cover the main needs that are identified by the end-users.

The division of the **Oasis** project into Sub-Projects (SPs) was done according to the cutting of the architecture into big blocks. **Oasis** encompasses 4 SPs.

SP1 has designed and develop the Common Operating Environment (COE) that is the common software modules and tools for application developers, implemented on the host infrastructure in which the applications are plugged. The building blocks contain minimal technological research, but are innovative in the sense of the new concept it implements for civil disaster management systems. SP1 brings the IT backbone of the pre-operational system and enables interoperability across the applications. It is also defined and developed such that it is easy to plug external applications that are compliant with the architecture mechanisms defined within Oasis project.

SP2 has designed and developed the emergency communications needed for **Oasis**. It brings the deployable communication backbone to support the IT system.

SP3 has designed and develop the C3I modules to give the operators an effective support to Command and to Control the operations. These software modules are plugged into the Common Operating Environment through standard APIs libraries.

SP4 has focussed on the analysis of emerging and new technologies to develop advanced decision support modules which will be later integrated in the C3I modules from SP3.

This document presents a summary of the SP1 sub-project achievements and lessons learnt about the development of a Common Operating Environment

2 REFERENCE DOCUMENTS

All these reference documents are public and available on **Oasis** project web site.

[RD01]	Oasis Terms and Acronym
[RD02]	Oasis project Executive Summary
[RD03]	Oasis Target System Design Summary
[RD04]	Recommendations of choice of standards and COTS

3 COE ACHIEVEMENTS

The COE and its architecture are the key building blocks for Oasis. The COE provides support and services to the high level Oasis functions (SP3, SP4) and access to the communications services (SP2). Innovative concepts were employed to build an open, flexible COE that fully supports system integration and rapid integration of applications.

The Oasis COE provides the basic common and services which are needed by any information system (such as the authentication of the users) and defines a unique access to these services, such as new Oasis compliant components can be plugged easily in a system. It also simplifies the interconnection between systems which respects the rules promoted by the project.

The COE is the core of Oasis with several important consequences:

- It defines open core interfaces and standards so that specific application modules can easily be 'plugged' into Oasis.
- It also defines the technical design environment (tools, languages, operating systems, middleware) of the complete Oasis system.
- Its study and development has been ahead of the developments of other Oasis components. Therefore it paves the way for establishing the overall framework of the complete Oasis system, in terms of development, procurement and deployment. It provides methods and tools which allows to develop new components quickly and economically.

The COE is designed to support the easy integration of new and legacy applications for disaster management. The COE aims to define and develop/integrate the Oasis system backbone by providing a standard open platform including:

- hardware and software architecture,
- system administration services,
- middleware,
- generic services.

The main objectives of the Oasis technical platform are:

- to support an open, flexible n-tier architecture that is scalable for different COE configurations, from simple to more complex and advanced systems,
- to be adaptable to support the different levels of Oasis (from co-ordination to field level),
- to implement a flexible state of the art framework that allows the rapid and cost effective development of software including the usage and integration of third party components,
- to be fast and reliable through a well-balanced usage of system resources,
- to be cost effective and be based on COTS products.

Common Operating Environments have been developed in the military business sector to support network centric defence concepts. The consortium's expertise and background in these areas have been exploited to be effective in the development of the Oasis COE.

Generic operational services, decision support software and knowledge based applications are connected to the COE via standard Application Programming Interfaces (APIs). The COE is capable of integrating new services or new software modules without major re-design of its architecture. This is a critical feature as it enables the easier migration and integration of future third party developments to Oasis.

The COE is scalable and therefore is able to be installed on a simple single platform with reduced capabilities as well as on a complete networked operational site. Considerable customisation flexibility must therefore be provided so that only the segments required to meet specific application needs are present at runtime.

The Common Operating Environment (COE) consists of Core Components and of Additional Support Components.
The Core Components are:

- COE Service Infrastructure
  - Application Manager (Service Broker)
  - User Manager
  - Notification Manager
  - Data Manager
- Event Manager
- Messaging Manager and Directory Manager

The Additional Support Components are:

- Geospatial Manager
- Localization
- Navigation

The following figure presents an overview of the COE components and their embedding in C3I applications and communications.

The Logbook component is part of the SP1 of the project as it has been developed by the leader of SP1. But it is a specific application and it is more logic to integrate it in the C3I applications in the diagram.

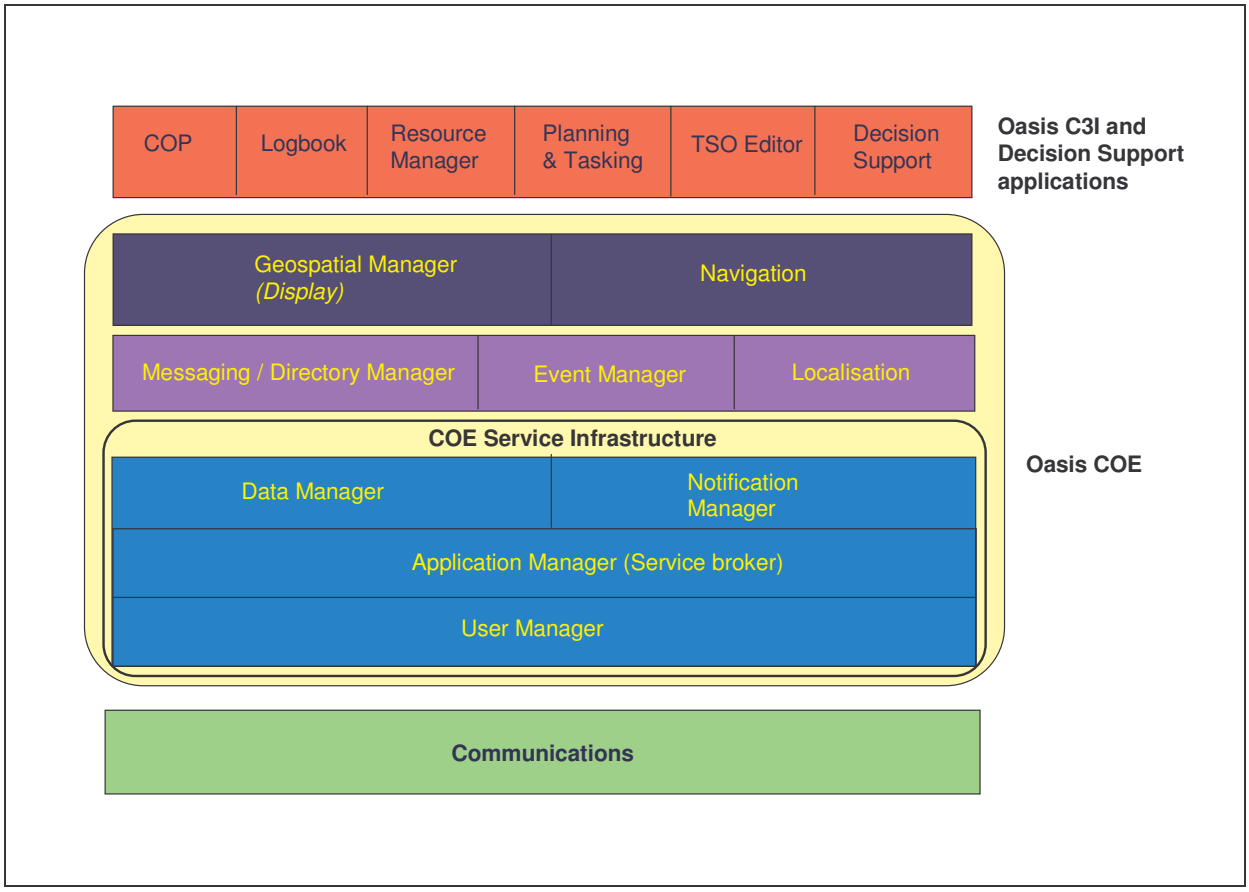

Figure 1: Oasis COE functional diagram

The architectural approach selected by the **Oasis** partners for the Common Operating Environment follows the paradigms of a Service Oriented Architecture (SOA). A general high-level overview is presented in the document [RD03].
A Service Oriented Architecture (SOA) is a way to define and provision an IT infrastructure to allow different applications to exchange data and interoperate regardless of the underlying operating systems or programming languages. The integration is realized via services that work (can be provided or consumed) on every application in the system. XML based service descriptions, messages and data models or schemas enable the interoperability between the participating services.

The major advantages are:

- Easy to integrate (legacy) systems,
- State of the art and will be used by more systems in the future,
- Meets OASIS interoperability requirements.

Its main disadvantage is its complexity: all frameworks implies some training or experience.

The used OASIS SOA solution comprises web-services as the implementation technology and an enterprise service bus (named **Oasis** Service Bus for this project) for the integration of all **Oasis**-internal applications and legacy systems. It comprises the components which build the run time infrastructure for services.

The Oasis Service Bus (OSB) is a set of infrastructure capabilities implemented by middleware technologies that enable a SOA. In other words, the OSB is not a product but a concept and a set of technologies and specifications that work together to enable interoperability and integration within a service environment. Beside the integration of legacy systems and external applications, the service bus approach is used within Oasis to ensure loose coupling between the different Oasis functional areas such as Mission Planning, Resource Management, Decision Support, and so on.
The participating components interoperate via XML based messages or service calls. The services are the only interface through which one component is accessing others. All services are linked to the service bus as illustrated and explained below.

The components "Application Manager" and "User Manager" are the infrastructure components which are needed that services could run.
The **User Manager** provides the functionality of user, roles and access rights maintenance which is needed to grant services access to each other and to the Oasis system.
The Application Manager provides the services to connect applications services.
The Notification Manager provides the basic services that application services could provide a subscription and notification service.
The Data Manager provides database functionality for persistent storage of data for application services.

As a basic approach for the SOA implementation, the Web Services concept has been chosen. There are three fundaments in this concept: consumer, service provider and service broker, which all communicate with SOAP messages.

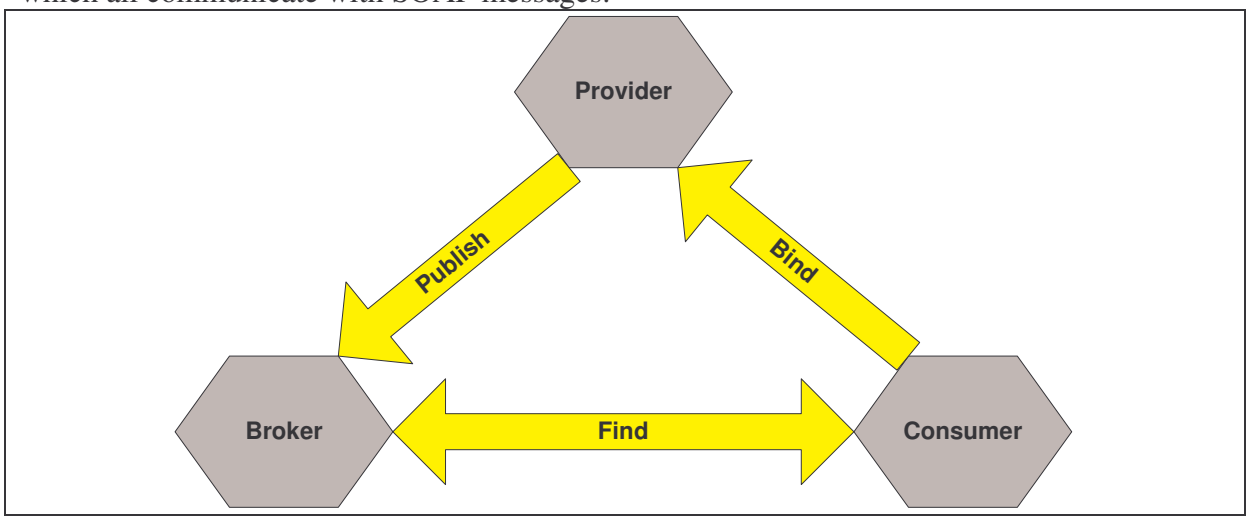

Figure 2: Web services triumvirate

It is a business of the **Application Manager** and its environment to function as:

- Service broker – it provides authorized consumers with information on required services. It maintains an instance of service registry, in which it keeps metadata on all services available in its node. If the node is connected with other node, it is possible to query also all external services provided by this external node.

- Service Proxy – the Application Manager is also a proxy, which means that consumers do not send their messages directly to the target services, but to the Application Manager which they are currently logged in. Then the Application Manager is responsible for the proper delivery of the messages to both parties.

- Session manager – during the session established between a consumer (client application) and an instance of Application Manager it keeps necessary information on logged-in user and his/her activities.

AM - Session Keeper

Just from the technological point of view there should be mentioned here a component (in effect a service) responsible for the keeping of the session context on the client side. Once a user is logged in and a session is established, the Session Keeper is instantiated as a singleton on a client side and lives here until the explicit end of session (i.e. logout). During its life this local service provides all its consumers (client applications) with the session context.

Data manager

The data manager is used by any other Oasis components.

The goal of this component is to allow storage of data of any kind, in a database environment. Data can be updated, deleted, queried upon, and fetched out from storage. Data can represent, but are not limited to:

- Emergency information
- Hazardous materials information
- Glossary
- Medical information
- geo-spatial information
- Persistent information from other components
- …

Interested components can subscribe to be notified when query-based events occurs (data creation, modification, deletion).

Messaging

A core element of Crisis Management is communication between people to exchange information, consult, command, etc.

Voice and video communication are two examples for real-time, synchronous communication. The quality requirements for the underlying communication network are high and therefore this type of communication is placed in SP2 ‘Communication’, where the communication infrastructure is owned.

Text based, asynchronous communication, as Email, has only very basic requirements for the communication network quality. Information is stored and forwarded when the communication network and the messaging infrastructure is available. Lost information is resend until the involved parties are satisfied.

This text based, asynchronous, store and forward communication is covered by the SP1 ‘Messaging’ component.

User directory Manager

The User Directory Manager provides address book functionality to hold information about:

- registered OASIS users;

- external contacts.

The amount of information held per entity will comprise the typical address book attributes and can be extended with additional information on request. The component is based on an open source software.

Event Manager

The "Event Manager" is to provide a centralised repository for the description of all the events managed by an Oasis node.

The Event Manager component provides the services to create, describe, merge, organise the information associated to events.

The Human Machine interface specific to this service is developed in the Geo-Spatial Manager. A HMI could be also provided by other existing components as the Logbook for example.

Logbook manager

The logbook manager tends at providing a repository for human-provided or system-provided logbook entries. A log may represent any past, on-going or future event or operations. A special type of log entry is also defined to handle alarms: they need to be acknowledged by the user.

The logbook stores information either provided by human operators or by automatic collection of information managed in the system. It stores these pieces of information in "books", allowing to retrieve any information easily. This is a useful information repository in many circumstances, for example when there are long events, needing to pass information concerning the event between successive users, or for the replay of what happened in order to learn from past events.

Geospatial Manager

The main objective of the "Geospatial Application Manager" is to provide a cartographic application where the features that are part of the crisis management (e.g. areas to protect or to evacuate, in-the-field teams, etc.) are displayed, in order to have a global view of the current situation during the management of an emergency.

This work package has developed 2 geospatial applications having a human-machine interface using 2 different existing cartographic tool; one is based on UDIG and the second on Map Object, an ESRI product. Standard cartographic functionalities are provided and, in addition, the "Geospatial Application Manager" provides mechanisms to integrate other Oasis components into own "graphic user interface" (GUI), allowing a high level of interoperability in order to compose an application useful during operations of search & rescue.

All Oasis users are concerned by this component, due to the fact they have to manage a large quantity of data related to men and assets that are linked to the territory. In general, a GIS-based application allows a faithful monitoring of the situation in a real-time environment in addition to a powerful instrument for the decision support.

The Geospatial Manager Application is composed by a set of layers or "thematic plans" containing geographical information represented by points, lines and polylines. Usually, from the cartographic application point of view, the objects displayed into the graphic user interface can have any meaning. Each object contained in a layer is identified by a coordinate system and have an associated set of attributes (alphanumeric information). In the Oasis framework, operational objects are stored in sets of structured information silos, allowing the

responders to select the information that they need for their activity (activities such as managing the resources, preparing possible scenarios, informing their hierarchy or the media, etc.).
The map can be moved (pan) or zoomed and the user can interact with it using the buttons contained into the toolbar and the menu shown when the mouse button is pressed. Other functionality have been developed in order to allow to the user a simple interaction with the map, like “zoom to a selected object” or “print map” or “measure a distance”.
Every object from every plan can be selected and its details can be visualised. In addition, the Geospatial Application Manager host functionalities implemented by the other components of ***Oasis*** project. These functionalities will operate using the currently selected object as input parameters. For instance, functionality like “send message” is offered by Messaging component and is hosted into the Geospatial Application Manager. The user can select a team displayed on the map and activate the “send message” functionality: a window will appear and the user is prompted to insert the text message to send to the team.

4 COE LESSONS LEARNT

4.1 Oasis Service Bus

The Oasis framework includes the capability to connect several control rooms together. This connection protocol does not allow the connected partner to access to all applications and information sets, but defines strictly the perimeter of the services which may be used by this partner. In principle, this is a very convenient mechanism, which can be used permanently (for control rooms which are always cooperating) or during a limited time period (for some activities during a specific event). This mechanism was used successfully during the 1st trial for permanent connections between several control rooms. However, the "on demand" connection has not been used (except for system tests) due to some problems in the software.

During POS2 a legacy system has been connected to an Oasis backbone. The C3I system from IMASS, used in many fire services control rooms in UK, has been slightly enhanced in order to be able to exchange information with an Oasis system. This allowed to setup a "coordination room" which displayed information from several organisations (owning Oasis as well as non Oasis information systems), and to provide to these participating organisations information from other rescue services without the need to build as many ad-hoc gateways as there were different systems. Another demonstration of the integration between a non Oasis and an Oasis system was shown during the final Oasis event: one Oasis system was connected to a Norwegian military system, in order to share some information between both systems. This connection was developed quickly (only a few weeks) and performed perfectly.

4.2 Use of Web services

There were several possible technical solutions for the implementations of the COE and more precisely for the User and Application Managers. We used the Open Source modules from the apache Foundation (Tomcat, and Axis) for several reasons:

- Commercial modules are very expensive and are not tailored for relatively small systems (commercial systems are targeting very large systems – for example information systems used by banks which thousands of on line users at a given time, which results in large infrastructures which are not consistent for systems used in control rooms with maximum 30 or 50 users)
- A lot of useful experiences, support and advices can be found in forums on the web, with very reactive skilled people.

These tools performed well (no crash, no performance issues) but we found serious problems due to the Operating Systems (OS). As the technology which is used is based on Web technologies, there are a lot of threats which are taken into account at the level of the operating systems (such as Linux and Windows). One of the counter-measures of the DOS (Deny Of Service) threat which is applied to the OS is the limitation of the number of connections to the HTTP protocol at a given moment. This is a very nice feature to protect Web servers against attacks… but this is a problem when the normal situation is to receive a large number of simultaneous requests. In the case of the COE, this was a normal situation, but we had a lot of trouble for finding a solution for removing this limitation (for example for the Windows XP OS, the limitation is part of the SP2 pack, but there is no "official" solution

to remove it). As a consequence, we have been obliged to disseminate our servers on all machines of the network, in order to share the connections on all these machines, which was not the optimal solution. Even if we had relatively large implementations for some trials, we could imagine that the problem could become serious in very large control rooms with tenths of operators.

As mentioned in the previous sections, the Oasis trials implied a large number of work stations and of operators, but none of the control rooms can be defined as a "large" control rooms. The setup of the trials was more the juxtaposition of several small control rooms collaborating together. As a consequence, we have not tested the COE in an environment with a large number of work stations and operators. Such an environment shall multiply the number of application messages which are exchanged (for example, if there are 50 work positions, each operator using a Geo-spatial manager, a logbook and one additional tool, that will result in several hundreds of TSO channels opened permanently). As we have seen in the previous section that there are some limitations in the operating systems which do not allow a large number of connections during a short period, the system shall have problems to cope with the load.

However, even if we did not found good solutions for this problem during the Oasis trials, it is a fact that such solutions shall exist (else no large systems would exist on the web).
Another problem shall be the capability of the servers to respond quickly to the requests. On Oasis we used very standard computers (they were chosen for their physical characteristics – dimension, weight, etc… as we needed to move them from one country to the other for our trials). For improved performances, there are servers with more advanced characteristics (fast quad core CPUs, fast disks, large memory footprint, gigabit connections, etc.) which are more tailored for such systems.

4.3 User and Application Managers

For the "normal user", the user manager is a very simple application: it allows the user to enter its login identifier and its password. The application manager is completely transparent for the "normal user". For the administrator, the use of these managers is slightly more complex. This complexity is not due to the tools themselves, it is the consequence of the need to prepare the list of all of the resources which are managed (Web services provided by Oasis components), and to associate users, their profiles and resources. It also requires to think very soon to the profiles which shall be ready for providing limited access to the resources of the system to external organisations, in order to set up these profiles even before they are required (setting up these profiles the day of the disaster demonstrates a very poor organisation!).

However, these mechanisms were not used at their maximum during the trials, due to the fact that there were some limitations in the connection mechanism between control rooms. The connection worked correctly, but the disconnection was problematic (some "ghost" connections seemed to remain, resulting in poor performances). The mechanism itself is good, but the implementation shall be improved and tested in more depth. The HMI for allowing the connection shall also be reviewed.

Whole AM architecture is built upon open source components, such as Apache Tomcat, SOAP message protocol and its implementation - Apache Axis. All these components proved reliable and sufficient in all trials where AM was used.

In case of use the AM in operational system it would be improved in these areas:

- user documentation
- flexible configuration
- robustness and security

Although current documentation is complete, it is not targeted for users not familiar with Oasis documentation format. It should be enhanced with more illustrations, examples and step by step tutorials as well.

Current configuration is complex, it covers all configurable features, but lacks user friendly configuration tool. In many cases the raw configuration files has to be edited directly using the text editor.

Up to now the AM architecture has been proved only in rather small scale scenarios of Oasis user trials. As a core component of Oasis system it has to be tested not only in largest system configurations, but also for various security threats.

4.4 Geo-Spatial Managers

4.4.1 Map quality

After POS1, the project received remarks about the quality of the maps: only vector maps were used, and some users are used to raster maps, which was a problem for them. The vector maps are more precise and more up to date, but the display is less user-friendly. On POS2, all types of maps were available. Users had the choice to use:

- several types of raster maps (we digitalised some of their maps for the trials),
- aerial images (we purchased Spot images and got some aerial photos)
- vector maps

Then for the Oasis event, we added the support of information gathered on the web (raster and vector maps from Google and Microsoft).

Switching between different types of maps was very appreciated by the users, allowing them to choose the best maps according to the location of the event, to their task, and to their habits.

However, even if this was a good demonstration, it shall be remained that the use of maps from the web (Google and Microsoft) is not free of charge for commercial and operational applications. Access to these data is relatively expensive.

4.4.2 Creation and edition of information related to the event

The Geo-Spatial managers have been improved in order to facilitate the creation and the edition of the TSO files describing the event. During POS1, it was not possible to use these capabilities, resulting in the need to use several tools for some very simple tasks. The capability to define templates was also appreciated, allowing to create very quickly the description of a new event with pre-defined fields.

The separation between background information and operational data has been added for POS2, in order to respond to some remarks done in POS1 (working on operational data was not easy as this was mixed with the static background information). This modification needed to take some "distance" from a pure GIS tool, as these GIS tools only see the world as a set of layers which can be superimposed in order to make a map. The GSM separation between background and operational information puts the focus on the operational information which is the centre of the preoccupations of the responders. In addition, some specific features were implemented on the operational information, allowing to define the type of information which is relevant to the operator (focus on a given event, or focus on the management of resources, etc.). It also implements the various hierarchies which may exist during events (event / sub event hierarchy, or event associations).

4.4.3 Provision of a map server

Even if some common services were available (such as the provision of map servers), these services were not used by all components which needed to display maps, which resulted in some visual differences between these tools and in the capabilities for manipulating the maps in these tools (for example in the decision support tools, or for the planning and tasking module). There were multiple reasons for not using this service (too many changes in the module, or specific features required which were not available in the service). As these services are very standard (they are defined by the Open GIS Consortium), it is difficult to implement them with very specific features. Other solutions shall be found, requiring sometimes relatively big changes inside the component itself.

4.4.4 Technological solutions

The geospatial manager is a standalone application which can be compared with viewer client in a web browser. In the Oasis project, both technologies were used, as the TSO Editor was developed as a "thin client" application.

Both technologies have advantages and disadvantages. The main reasons why the GSM were developed as a stand alone application are:

- It works even if the network is not operational. This was important for the users in the field, as they were using stand alone PC, connected only through a very slow and unreliable network (the network itself is robust, but the user could be in a place – tunnel, rural area – where he does not receive the signal)
- Even if the "thin client" are more and more sophisticated (cf. the TSO Editor), the development of a stand alone client provides much more capabilities
- The performances are better, due to the fact that there is no communication overhead in the middle (allowing for example the real time display of the position of the resources).

4.5 Data Manager

The Oasis data manager was one of the cornerstones of the TSO channel mechanism, and was also used by the User, Application and Event managers. But it was almost not used by the SP3 and SP4 modules, which were relying on their own databases. So definition of such a component is an actual question.

As a central module of the COE and for the COE, the DM was very useful. Applications preferred to use their own repositories as it seemed to be less complex and less "expensive".

This is a fact that the use of the DM is a little bit more complex when developers are used to implement a direct access to a standard SQL database.

However, all the applications which used their own repositories did not implement the access control on their data. If they had to implement it, this would have been complex for them. For an application which is based on the DM, this is straightforward, as the application can delegate this management to the DM.

As a conclusion, providing a data manager is almost useless for the simple storage of information, as there are many tools which are easier to use and faster. If some central access control mechanism needs to be used, this becomes the easiest solution for implementing and controlling that these mechanisms are correctly used.

4.6 Notification Manager

The Notification was another cornerstone of the TSO channel mechanism. It was very useful for disseminating automatically changes in the situation to all the tools in charge of presenting it to the responders.

On the bad side, the notification mechanism was the component most impacted by the limitation of the web services, as explained in the section of the User and Application managers section. We had to implement some simplification mechanisms (direct connections with the data manager, not using Web services) in order to limit the number of simultaneous connections.

Such a mechanism is mandatory. There are now standards for implementing the subscription / notification services (cf. the Web Services Base Notification standard, proposed by the Oasis Open consortium). However, this standard was made public only at the end of 2006, which was too late for the Oasis project (first implementation arrived months later). This specification and the available implementation shall be reviewed in order to check that it covers the needs of the emergency and emergency management applications.

4.7 Logbook

During POS1, the logbook was appreciated as it provides a “text only” view of the information describing the events and the activities. However, it was said that it shall include automatically the information which are already provided by other means, mainly the information already contained in the TSO files.

For POS2, the logbook was improved and was added as a subscriber of the TSO channels. Other modifications were needed consequently, in order to display the TSO information (which is more complete than the information normally acquired by the logbook).

The logbook was very appreciated in the Romanian trial as it offers an efficient way to store and retrieve information with a friendly HMI.

4.8 GPS driver

For POS1, a first component was developed with very ambitious specifications. It was very useful, but extremely complex, compared to the service that it provided. Installation is a pain

(set-up of an IIS server, declaration of the service, 3 levels of modules!). And even once installed, it needs to be watched all the time.

For POS2, this was largely simplified, being seen as a local service with 2 outputs:

- Locally, several applications were allowed to subscribe in order to receive in real time the time, latitude and longitude of the attached GPS receiver. This was used by the Geo Spatial Manager in the "terrain mode", in order to display the current position of the equipped users in the field
- Remotely, it was able to send an email to a pre-defined address. This was also used for updating the position of the resources. The GPS driver was configured with the email address of the Resource Manager. This was very convenient, as the email protocol (SMTP) is supported by many communication systems, including the TETRA network used in Romania during the POS2 trial.

In this simplified configuration, it was successfully used during POS2, providing fast and accurate positions of the in the field resources.

4.9 The Event Manager

The Event Manager was designed only for the POS2, as there was a limitation in the POS1 module which allowed the management of only one event.
The Event Manager implements the hierarchy of events, a major event being split in sub-events, which provides a simpler view of the on going activities for the operators. The Event Manager was used as the "coordinator" for some SP1 and SP3 components, such as the Geo Spatial Manager and the Resource Manager which are organising their IHM according to the hierarchies defined in the Event Manager.
This a simple module, connected to the subscription mechanism. The definition of the hierarchies between events / sub-events was used in the POS2 trials and was appreciated, as it allows to separate different levels during the management of a large event (for example, the main event being a flood, with sub-events describing the flooding of a village requiring its evacuation, another sub-event being the destruction of a bridge requiring to repair it, etc.).

4.10 Forms tool

The form tool is a simple application that allows to prepare simple forms to be filled in the fields; its goal is to replace the paper forms with a more convenient solution which can be used over various networks. It was originally developed for a previous European project (EGERIS), but appreciated as it is simple and "light weight"; forms are easy to fill with a tablet or a rugged PC having a small screen.

For Oasis, it was used in the trial in Romania: the forms were created by the Romanian users based on their own paper forms (Situation reports for example). They were largely used during the trial. For these users, they were very easy to understand and to complete.

4.11 Navigation

The Navigation service was successfully integrated in UK for POS1, but it was not demonstrated to users. There were some limitations to this service, for example it did not provide a "gazeeter" service.

There are now several services which could be used:

- The OpenGIS consortium has developed the "OpenLS" specification which could be used for such a service
- Another interesting track is the use of the specification provided by organisation like Google for its Google Map service. Such a specification is very simple and the service itself could be integrated in a system, however with some important limitations:
  - The use of such a service is not free for commercial use
  - (more important): the service cannot be customised: it works well in normal circumstances, but it cannot take into account important modifications during a disaster, such as the fact that roads or bridges cannot be used due to the incident!

4.12 User Directory

The User Directory service relied on the LDAP standard, which is commonly used in many organisations. The Open Source Open LDAP software was used, which is also used in many operational systems. The main criticism was about the administration client software which was not very friendly (this tool is not used by "normal" users, only by the administrator for managing the users.

It was only used in the POS1 trial.

4.13 Security issues in Oasis

There were no strict security requirements in the Oasis specification, except the fact that the organisations want to keep the control of their data and applications. This requirement was implemented by the User, Application and Data managers. As such it worked well.

Such a system will not comply with more strict security rules, as those which are used in a military environment. However, as it was demonstrated during the Oasis final event, it shall be possible to build a gateway between an Oasis system and a military information system in order to share some non restricted information.

5 CONCLUSION

If we consider the different point of views, the COE provided a good level of services for the different actors of the Oasis project:

- For the end users, most of the tools were appreciated. For POS1, the use of most of them was criticised mainly for a lack of friendliness. This was mainly due to the very short specification and development times. As a consequence, the project decided to limit the modifications for POS2 (very few additional services), but to focus on those modifications which are improving the user friendliness and for the modifications which were needed to take into account the methods or the protocols used by our POS2 users. As a result, we saw that these improvements were well received, even if there are still some additional improvements which were required (some of them were demonstrated during the final Oasis event)
- For the developers, the COE components were also improved during the POS1 and the POS2, for a better stability and additional performances (the same computers were used for both trials). A few additional services were developed, some were abandoned. Even if there was not a "big trial with everything inside" as it was done for POS1, all POS2 trials were ambitious and supported a large diversity of users and a large number of responders with their specific requirements and expectations. Due to this "inflation" of users, we found new problems which were difficult to solve (mainly the problem due to the Web Services overflow). These problems were circumvented so that the trials were successful, but it shows that these technologies are still not completely mature.

A last question remains: are the mechanisms defined in the target system design sufficient for obtaining a good COE? After this long experience (4 years), it seems that all basic services were provided. There are additional services which exist "traditionally" in military systems which could be added in order to improve the Oasis COE. They are not strictly mandatory, they are only improving the security aspects, and but they would allow to progress again of one additional step. These services are:

- A formal authentication service for the access control, based on a standard protocol such as Kerberos. This would replace partly the User and Application managers, but at the price of a much complex development process (note that there is no standard for using Kerberos in a Web service environment, only some implementations developed by a few big organisations)
- The use of some common services, such as synchronising all computers through the use of the Network time protocol
- A more formal solution for the supervision of the equipments and applications, allowing the administrators to react immediately when something goes wrong in the information system when important events are in progress.