## FORM

# IST-1999-10357

Engineering a Co-operative Inter-Enterprise Management Framework Supporting Dynamic Federated Organisations Management

Document Number:	IST-1999-10357/BRI/WP5/0230
Title of Document:	D10 Validation of Inter-Enterprise Management Framework (Trial 2)
<b>Restrictions:</b> (P/R/L/I)*	Р
Nature of the Document: (P/R/S/T/O)**	R
Contractual Date of Delivery to the CEC:	2002-01-31
Actual Date of Delivery to the CEC:	2002-03-08

Workpackage responsible for the Deliverable: WP5
Editor: Niamh Quinn (BRI)
Contributor(s): FORM Consortium
Reviewer(s): Soren Vejgaard-Nielsen (UHC), Jacques Brook (KPN)

#### ABSTRACT

This deliverable presents the setup of Trial 2. Trial 2 represents the second and last evaluation phase of the FORM project. Trial 2 assessed the overall results of the project including all the components of the Inter-Enterprise Management Framework. The documentation of the final project trial (Trial 2) provides information on the trial infrastructure, trial test cases and evaluation based on the operational requirements and the development aspects. The structure of Trial 2 is presented; the four test teams within the project demonstrated their management systems. These demonstrations were then assessed against the operational requirements for the Inter-Enterprise service, which were drawn up at the beginning of the project. The development aspects of an Open Development Framework in which management systems for an IES can be developed commercially were also evaluated.

#### KEYWORDS

Evaluation, Validation, Trial, Test cases, Requirement assessment, J2EE, XML, Security, Fulfilment, IES, VPN, Assurance, Accounting, Reporting, UML, Development Aspects, Methodology, Architecture, Technology

© 2001 by the FORM Consortium.

See http://www.uhc.dk/form/index.htm for further details

\* Type: P:Public, R-Restricted, L-Limited, I-Internal

\*\* Nature: P-Prototype, R-Report, S-Specification, T-Tool, O-Other

## IST-1999 FORM

## **Deliverable D10**

### Validation of Inter-Enterprise Management framework (Trial 2)

Editor :	Niamh Quinn (BRI)
Status – Version :	Version 9
Date :	2001-11-27
Distribution :	Public
Code :	IST-1999-10357/BRI/WP5/0230

© Copyright by the FORM Consortium. The FORM Consortium consists of:

Atos Origin Intégration (France) – Project Coordinator	
Broadcom Eireann Research Ltd. (Ireland)	Trinity College Dublin (Ireland)
DELTA Danish, Electronics, Light & Acoustics (Denmark)	UH Communications A/S (Denmark)
Fraunhofer Fokus (Germany)	University College London (UK)
LM Ericsson A/S (Denmark)	Waterford Institute of Technology (Ireland)
TDC Tele Danmark A/S (Denmark)	KPN Research (The Netherlands)

## **Table of Contents**

EXECUTIVE SUMMARY			
1	INTRODUCTION	6	
	1.1 PURPOSE		
	1.2         SCOPE           1.3         DOCUMENT STRUCTURE.		
	1.5 DOCUMENT STRUCTURE	0	
2	EVALUATION PLANNING	7	
	2.1 OPERATIONAL ASPECTS	7	
	2.1.1 Web-based Operational Requirement Tracking		
	2.1.2 Test Cases Description		
	2.1.2.1 Fulfilment – IES Test Cases		
	2.1.2.2 Fulfilment – VPN Test Cases	10	
	2.1.2.3 Assurance Test Cases		
	2.1.2.4 Billing Test Cases		
	2.2 DEVELOPMENT ASPECTS		
	2.2.1 Overview of Assessment of Development Aspects of Trial 2		
	2.2.2 Logical Architecture Evaluation Plans		
	2.2.2.1 Scope of Logical Architecture Evaluation		
	2.2.2.2 Assessment of Architectural Principles and Benefits		
	<ul><li>2.2.2.3 Assessment EIM integration</li><li>2.2.2.4 Evaluation of Trial Models</li></ul>		
	2.2.2.4 Evaluation of Trial Models 2.2.3 Development Methodology Evaluation Plans		
	2.2.5 Development Methodology Evaluation Plans		
	2.2.4 Technology Architecture Evaluation T tans		
•			
3			
	3.1 OPERATIONAL REQUIREMENTS		
	3.1.1 Test Cases Results & Conclusions		
	3.1.1.1 Fulfilment – VPN Test Cases		
	3.1.1.2 Fulfilment – IES Test Cases		
	3.1.1.3     Assurance Test Cases.       3.1.1.4     Billing Test Cases.		
	3.1.1.4 Billing Test Cases 3.1.2 Operational Requirements Tracking		
	3.1.2.1 Fulfilment-IES (Test Team 1) Test Cases		
	3.1.2.2 Fulfilment-VPN (Test Team 2) Test Cases		
	3.1.2.3 Assurance (Test Team 3) Test Cases		
	3.1.2.4 Billing (Test Team 4) Test Cases		
	3.2 DEVELOPMENT ASPECTS EVALUATION RESULTS		
	3.2.1 Logical Architecture Results		
	3.2.1.1 Assessment of Architectural Principles and Benefits		
	3.2.1.2 Clarity of Principles		
	3.2.1.3 Assessment of EIM Integration		
	3.2.1.4 Assessment of Trial Models		
	3.2.2 Development Methodology		
	3.2.3 Technology Architecture		
	3.2.3.2 XML for Contracts		
	3.2.3.3 XML for mediation		
4	CONCLUSIONS	60	
5	REFERENCES	61	
_			
6	ACRONYMS		

#### **Executive Summary**

FORM has enabled its industrial partners to exploit services, software systems and software components for the management of outsourced, co-operative Inter-Enterprise Services.

This deliverable presents Trial 2. Trial 2 encompasses the demonstrations of the Inter-Enterprise Management Systems developed during the FORM project and the validation of the Inter-Enterprise Management Framework. The Inter-Enterprise Management Systems were grouped into four test teams: Fulfilment IES, Fulfilment VPN, Assurance and Billing. Each of the four test teams trials consisted of a demonstration of their system and a evaluation of their system against the operational requirements as perceived by the End Customer and the Inter-Enterprise Service Provider which were collected at the beginning of the project.

The Open Development Framework (ODF), which was used to develop each of the trial systems, was also evaluated at this stage. The development aspects of the ODF were considered under the four high level quadrants of the Framework, i.e., the Logical Architecture, the Technology Selection Guidelines, the Development Methodology and the Logical Architecture. The development aspects were evaluated across the four test teams through the use of questionnaires and discussions.

#### **1** Introduction

#### 1.1 Purpose

The focus of this document is to present the results of Trial 2, the validation of the Inter-Enterprise Management framework. The objectives of the Trial 2 evaluation is to assess the different aspects of FORM results

- Operational (based on operational requirements assessed through test cases)
- Development (including Methodology, Logical Architecture and Technology Selection Guidelines)

The operational aspects have been evaluated by assessing operational requirements against test cases using a webtool. The development aspects have been assessed through the use of questionnaires and interviews.

#### 1.2 Scope

Trial 2 is the second and last phase of the validation of the Inter-Enterprise Management framework. In Trial 1, the focus was on the development requirements. Trial 1 was an assessment and overview of the results of the 'work in progress' of the development of the Inter-Enterprise management system and infrastructure at that time, where partners are using an iterative approach to design and build trial systems, which enabled them to test required system functionality. Some partners reused platforms and subsystems from existing implementations, while others built their trial systems and subsystem components from scratch.

In this deliverable, more attention to the operational requirements is given. Each of the test teams' test cases presented in this document has been analysed against the operational requirements which are addressed with it.

The development aspects of the ODF have also been evaluated across the four test teams.

#### **1.3 Document Structure**

Section 2 details the planning and structure of the evaluation. It consists of two main subsections, firstly covering operational requirements and secondly dealing with the development aspects.

Section 3 provides information regarding the evaluation results. It is also divided into subsections related to operational requirements and development aspects.

Section 4 provides conclusions on the evaluation and is followed by an acronym list and references.

Two annexes accompany this document. The first annex is concerned with the operational requirements and contains the test plans for each of the four test groups. The second annex contains the questionnaires used to assess the development aspects of the ODF.

#### 2 Evaluation Planning

In this section, an overview of the final FORM evaluation plans.

On the top-level the FORM evaluation has been divided between Development aspects and Operational Aspects. The Operational aspects cover integration events of developed Building Blocks supporting assessment of the Operational Requirements collected in the initial stages of the project. The Development aspects are assessed according to top-level structure of the Open Development Framework (ODF). Figure 2-1 illustrates the elements in the FORM evaluation approach.



#### **FORM Evaluation Overview**

Figure 2-1: FORM Evaluation Overview

#### 2.1 Operational Aspects

Trial 2 was planned as a set of *Final Integration Events*, which marked the end of integration work performed for WP5 assessment. Four trial teams conducted the integration events and the Trial 2 schedule was as follows:

F-IES Trial - Trial Team 1 (UCL, UHC, GMD): Week 49 Copenhagen

F-VPN Trial - Trial Team 2 (ATOS, LMD, DLT): Week 49 Copenhagen

Assurance Trial - Trial Team 3 (BRI, TCD, TDC): Week 50 Dublin

Billing Trial - Trial Team 4 (FOKUS, WIT): Week 46 Berlin

Trial 2 Operational Evaluation consists of auditing observed system operation against test cases and assessing observed system behaviour against Operational Requirements via a developed web tool

#### 2.1.1 Web-based Operational Requirement Tracking

As the captured requirements were only maintained on paper in a project deliverable it was decided to make them more 'alive', improve accessibility and to link them to the tests-cases in trial 2, through a simple web-based system.

Several COTS systems exist for tracing and maintaining requirements (e.g. Rational RequisitePro), but most are intended for production type systems/environments and are more extensive than needed.

Rational RequisitePro is a complete Requirements Management tool (requires an enterprise database; Oracle or Microsoft SQL Server) that comes with a Web interface (requires a Windows Server; Microsoft Windows NT Server 4.0 or Microsoft Windows 2000 Advanced Server and HTTP Server: Microsoft IIS).

The project preferred to develop a small web-based system - targeted only at the project needs - to maintain and link the operational requirements to the second trial test cases.

Apart from some obvious logistics advantages, this also has the potential of contributing to the dissemination effort, as non-FORM industrial actors in the IESP-value chain, could be granted online access to the requirements.

The system was to fulfil a few basic requirements:

- Enter/Update Requirement data
- Enter/Update Test-Case data
- Enter/Search/Update links between requirements and test-cases
- Search requirements and test-cases

The system was developed during the course of WP5 and will be made publicly available on the FORM web at the end of the project. The system has a public search part available at http://skinfaxe.delta.dk/reqsys\_public DELTA will maintain this web-based tool including the requirements database for minimum one year after the FORM project ends.

In the project the value-chain for Inter-enterprise services – and hence the different actors having requirements to the FORM Framework and results - was identified as: End-Customer -> IESP -> Management System Provider -> Management Component provider.

The captured requirements were grouped into Development vs. Operational Requirements, with the Developments Requirements assessed in Trial 1. The operational requirements were grouped mainly according to End Customer (EC) and IES Provider (IESP) and the sub-grouped into functional areas (identified by a *Functionality Block Label*):

- GE: General
- SA: Service Level Agreement
- IA: Information Service Assurance (QoS)
- QA: Network Provisioning Assurance (QoS)
- CB: Charging and Billing
- SC: Security & Outsourcing
- IE: Inter-Enterprise System Provider to Management System (Outsourcing)
- MS: Management System Provider to Management Component Developer System (Outsourcing)

These groupings are shown in Figure 2-2.



**Figure 2-2: Operational Requirements Groupings** 

The aim of the assessment was:

- Assess to which extent the trial 2 development addressed these requirements (both specification/concept-wise and implementation-wise)
- Assess to which extent the functional areas were addressed. (The functionality block label will be used when assessing how well the captured requirements are covered by trial implementations)

As the development efforts targeted prototyping, it was foreseen that strict adherence to the requirements, would not be pursued by the development teams. A simple 'Yes/No' answer for whether requirements were fulfilled by the trial implementation or not, would provide little information, therefore the following process and criteria was used:

During the Test Planning, the planner suggests a list of requirements to be addressed by the specific test cases targeting specific implementations.

As part of the actual test-case assessment, each associated requirement was assessed according to:

- *Concept-wise Fulfilment Degree* to which degree the requirement would be covered by the BB concept/specification if/when implemented.
- Implementation-wise Fulfilment Degree to which degree the requirement is covered by a actual trial implementation

For each of these two 'fulfilment degrees' the following groupings introduced were used:

- *Fulfilled*: This assessment indicates that the requirement has been fulfilled completely. The comment should specify what features of the test case satisfy the requirement.
- *Partly Fulfilled*: This assessment indicates that the requirement has been partially fulfilled. The comment should specify what features of the test case satisfy the requirement and what part of the requirement remains unfulfilled. If there are plans to address the unfulfilled part these may be mentioned.
- *Not Fulfilled*: This assessment indicates that the requirement has not been fulfilled, but implies that it is a target requirement to be addressed by the test case. Any plans to address the requirement should be mentioned.

• *Out of Scope*: This assessment indicates that the requirement is recognised as a valid one but has been judged out of scope of FORM. The reason for this assessment should be mentioned; typically this would be due to resource limitations or technology/cost restrictions.

The full set of responses is provided as part of the web-tool and is presented in Annex A. A summary of the assessment is given in section 3.1.1.

#### 2.1.2 Test Cases Description

The full trial 2 descriptions are contained in annex A. The following subsections contain short descriptions of the test cases of each of the four trial groups.

#### 2.1.2.1 Fulfilment – IES Test Cases

The main purpose of the Trial 2, Test Team 1 (TT1) was to validate the functionality and interoperability of the F-IES building blocks as well as the usability of their contracts.

This included evaluation of:

- The use of schema/DTD for SLA negotiation.
- Applicability of SLA negotiation sequence diagrams
- The scalability/usability of technology mediation in the SLA repository (Script based technology gateway)
- Platform support for SLA negotiation and order handling components
- Integration between SLA negotiation and SLA Handling

#### 2.1.2.2 Fulfilment – VPN Test Cases

As part of Trial 2, Test Team 2 (TT2) conducted *integration* and *basic functionality* tests of the three developed VPNS building blocks. A fourth BB - the GQIPS BB - was integrated in simulated mode to support the overall creation of a VPN service with security and bandwidth QoS. The test encompassed three main test cases and 10 subordinate cases. The mappings onto Operational Requirements were specified only for the main test cases.

The purpose of Trial 2 for TT2 was thus twofold:

- Integration of BBs from four FORM partners.
- Test of basic BB functionality for creation of a VPN Service, containing the virtual topology, and activation of the VPN connection between two endpoints in the virtual topology.

#### 2.1.2.3 Assurance Test Cases

• Customer Login and Validation

The selected web programming techniques must enable generation of dynamic client web-pages, where the information and menu options are generated dynamically based on customer ID.

The web application should enable dynamic retrieval of customer specific information both locally and external i.e. from other management -systems or –components.

The web application should enable that information regarding individual customers or customer types are stored in the XML format and are handled with 'open source' XML tools by the web application.

The web application must be able to identify the customer client browser during a session and secure that information generated or exchanged during a session will not be seen by be available to other sessions i.e. customers.

The web application must be able to identify the customer client browser type e.g. WAP phone and select type specific templates and information.

• Reporting Template Completion

Use customer id and customer menu selections to collect reporting data to fill out a customer specific reporting template with 'dynamic' data.

• Report Customisation

The customer specific reports are based on information that can be changed and adapted to individual customer needs. Configuration and customisation of reports should be controlled by updates of XML documents or by adding new XML documents local or external to the web application.

• Production of Assurance Configurations

When an SLA is submitted to the assurance system it is necessary to configure the various components of the system to support it. The purpose of this test case is to evaluate how the SLA is processed by the system and to ensure that the correct configurations are produced for distribution to the other components of the system.

• Service Monitoring

Once the assurance system has been configured in response to a new SLA the system will begin to monitor the service. This involves a number of different components distributed in the customer, provider and IES domains. Each of the components, called Server Monitors, in the customer and provider domains are responsible for collecting the statistics produced locally and processing them, if necessary, for use by the performance monitor. The performance monitor, in the IES domain, is then responsible for aggregating the statistics into metrics that match those specified in the SLA. The purpose of this test case is to evaluate how this process is currently supported by the system.

Service Violation

Reporting while monitoring a service the assurance system calculates the values of the metrics used within the SLA. However it must also compare the values of these metrics to thresholds specified in the SLA to ensure that it has not been violated. If a violation does occur then the event must be generated to indicate this to interested parties. The purpose of this test case is to ensure that these events are produced and that they correctly identify the parameters that caused the violation to occur.

• Workflow implementation of Business Processes

The workflow framework enables flexible management of business processes within a system. The purpose of this test case is to evaluate the implementation of assurance business processes using the workflow framework. The assurance client invokes assurance processes. The workflow framework implements the control flow and data flow for the Building Blocks to implement the processes. Two different assurance processes were tested for the configuration of the assurance Building Blocks to support an SLA.

• B3 Setting

The purpose of this test case is to show how a network operator sets the domain information, inputs and draws a Network Topology (NT) into the GQIPS sub-system, using the B3 console.

• Single domain service negotiation

The purpose of this test case is to check the formulation, formatting, sending, receiving, and answering of a service negotiation, considering only a single domain and a QoS request which is not to resource-hungry (i.e. whose request can be fulfilled).

• Events subscription and notification

The purpose of this test case is to show how an interested party can subscribe to B3's event notification service, in order to be informed latter, when events will occur onto the service. Because there is no use of routers in this first trial, and that therefore no user traffic will be routed, the word *event* in this document is used to represent issues happening when the service is expired, or cancelled.

• Multi domains RAR negotiation

The purpose of this test case is to test the inter-domain service negotiation protocol, through a GQIPS'B3 to GQIPS's B3 communication.

#### 2.1.2.4 Billing Test Cases

The trial is based on the execution of two trial scenarios. The first will demonstrate single service (VoIP) provision and rating/discounting for customer charging and service provider settlement. The second will demonstrate composed service provision and rating.

These scenarios will support the evaluation of:

- Preliminary test of InterdomainAcctMan contract: This was done in Trial 2 through the aggregation of multiple usage sessions.
- Support for convergence of services (voice and data): This was demonstrated in Trial 2 by a composite service called Online Collaboration Service.
- Evaluation of composite service information model: This is largely based on and makes use of IPDR information model. The usage information sent by Online Collaboration Service model is modelled on composite service information model.
- Mediation of the usage of multiple services in real-time and adaptable and practicability of federated mediation in multiple SP environment Fulfilment and Billing MCG sub-scenario
- Fulfilment and Billing MCG sub-scenario
- OSP (Open Service Platform) support for Federated Mediation Adaptor (FMA) Building Block
- 3 Phase service discounting usage discounting, periodic discounting and incentive discounting
- E-IPDR rating and discounting
- Aggregated rating for composed service usage
- IESP broker settlement rating
- SLA/SLS based rating algorithms
- VoIP service provision and usage mediation
- An E-IPDR recorder and Database
- SOAP/.Net webservices

#### 2.2 Development Aspects

#### 2.2.1 Overview of Assessment of Development Aspects of Trial 2

The original Development Requirements that were used as the basis for the Trial 1 assessment were taken from the TM Forum document GB909 [gb909]. Many of these requirements are not so directly relevant to the work now being done in FORM. In addition, these requirements are no longer being actively used in the TM Forum and are therefore of less industry relevance than before Trial 1.

It is then suggested that the evaluation of the ODF is driven from the principles presented for the different parts of the Framework, primarily from the Logical Architecture [lewis] and the Development Methodology white papers [formD12].

It is first necessary to understand the relationships between the ODF principles, the Logical Architecture, the Development Methodology and the work conducted in developing the Trial 2 systems.

The ODF aims to address the goal of supporting the construction of management systems from separately sourced software components. This goal is refined by addressing the needs to the four generic stakeholders that represent the different types of organisations present in such a market: namely:

- Standards Bodies, which develop the industry agreements that underpin interoperability and integration of separately sourced software components
- Independent Software Vendors, which develop and market software components
- System Integrators, which develop management systems constructed from separately sourced software components
- Service Providers, which possess the business requirements for management systems and operate them.

The ODF therefore aims to:

- 1. Support common mechanisms for the communication of products between these stakeholders
- 2. Ensure the processes for developing the products exchanged between stakeholders converge with industry best practise.

Based on an analysis of the state-of-the-art in management software architectures and component based development, in particular the NGOSS architecture being developed by the TM Forum a set of architectural principles was established for the ODF which supported these aims by providing relevant benefits to the ODF stakeholders.

These architectural principles (contained in annex B) have driven the definition of the meta-model in the Logical Architecture. The model elements in the meta-model have been mapped to modelling artefacts that make up the Development Methodology. These modelling artefacts are what has actually been produced by developers of the Trial 2 systems and the components from which they are contracted. It is the models, and the experiences gained from developing these models, that constitute the basis for evaluating the ODF in this document, see Figure 2-3.

The development of the Trial 2 system and the associated system and component-related models reflects the guidance given to developers through the Development Methodology Guidelines. The Trial 2 development experiences, therefore, only indirectly reflect an application of the architectural principles and the related meta-model from the Logical Architecture. As a result, assessment of the experiences of developers in actually analysing, designing and implementing the systems is restricted to an evaluation of the Development Methodology.

The evaluation undertaken in this document can therefore be summarised as follows:

#### **Logical Architecture**

Evaluation of the Logical Architecture consists of the following:

- An evaluation of the clarity of the architectural principles and the relevance of their associated stakeholder benefits.
- An evaluation of the designs represented by the Trial 2 Models (both Management System Models and Reusable Element Models) consists of characterising these systems in terms of the meta-model and then assessing aspects of their alignment to the meta-model that does not result directly from following the Development Methodology.
- A specific evaluation exercise to assess the feasibility of generating an integrated External Information Model from the various information models generated for Contracts in Trial 2.

#### **Technology Selection Guidelines**

The ODF does not aim to define a specific technology platform to meet its goals and its Technology Selection Guidelines, therefore, it consists only of guidelines on the factors that should influence technology selection. The evaluation of the Technology Selection Guidelines addresses the technology selections that have been made in the Trial 2 systems.

#### **Development Methodology**

The Development Methodology is evaluated by assessing the experiences of developers in applying the methodological Guidelines during the Trial 2 development activities.

#### **Reusable Elements**

The Reusable Elements are evaluated by assessing their generality or specificity to the IES domain as part of the Logical Architecture Assessment. The *functional contents* of the Reusable Elements are assessed as part of Operational Requirements evaluation.



#### Figure 2-3: Summary of ODF element related to Trial 2 evaluations

The following sections provide the detailed plans for the assessment of the individual portions of the ODF.

#### 2.2.2 Logical Architecture Evaluation Plans

#### 2.2.2.1 Scope of Logical Architecture Evaluation

The ODF has been developed as a framework for the long-term. Therefore Logical Architecture contains principles and Architectural Model elements that are seen as important to the future use of the ODF, but which, due to limited resources and the relative priorities of other project objects, have not been addressed in the Trials and therefore have not been evaluated. This section uses the list of architectural principles to give an overview of what has and what has not been assessed from the Logical Architecture.

P1: Management Systems are software systems that perform some task related to communications or systems management in an operational environment. They are constructed partially or fully from Building Blocks (BB).

Several different Management Systems have been developed by the Trial workgroups, and the use of BBs in their construction can be seen in [formD11] and is assessed as part of the trial models.

P2: Building Blocks are pieces of software that are atomic units of deployment (one can be replaced in a running system without requiring other BBs to be replaced).

The trial consisted of initial functional prototypes, and not fully operational systems. Therefore, there has not been an opportunity to assess the replacement of a BB at run-time, nor the resources to develop a second implementation of a BB to act as a replacement.

P3: Building Blocks are atomic units of system management.

The system management of the management building blocks developed in the trials was not addressed, though system management of service components (which were not necessarily BBs) was conducted as part of the assurance trial.

P4: Building Blocks may support multiple interface types termed Contracts.

It emerged that all except one of the BBs implemented in the Trial were designed with only one contract each (Assurance Configuration has two). Therefore this principle was not strongly exercised or, therefore, assessed.

*P5: A Contract may support multiple business operations.* 

This is evaluated as part of the trial model assessment.

*P6: The Logical Architecture does* not *prescribe the technology to be used in implementing Building Blocks or their Contracts.* 

This is a statement about the form of the Logical Architecture itself and therefore not something that can be assessed through the Trials.

#### P7: Contracts may be defined in a technology neutral form or a technology specific form.

The definition of a suitable language for the technology neutral form of a contract remains a challenge as described in [formD9]. The Contract form used in the trial was therefore only partially technology neutral, with the information content defined in a common technology neutral manner, based on the DMTF's CIM, and the Contract interface signature being defined in a technology specific manner. The use of different languages for the latter is recorded in the Trial model assessment.

P8: A BB implements a Contract in a technology specific form. When mapped from a Contract specification in a technology neutral form, this must be performed using an explicitly described transform.

As the Contracts were not fully defined in a Technology Neutral form, no attempt was made to define transforms to any technology specific format.

P9: Different Contracts may support different interface definition paradigms, though one Contract specification can only support one such paradigm. Interface definition paradigms include, but are not limited to, model-centric, operation-centric and message-centric. Different paradigms are typically suited to specific ranges of technologies.

Though a technology neutral form was not used for the contract, the trial model assessment attempt to capture the interface definition paradigm used for each Contract.

*P10: The definition of the information that is passed via a Contract should be published separately to the Contract specification.* 

The information content of each Contract has been explicitly defined in the Boundary Information Models (BIM) for each Contract, as can be seen in the on-line Contract catalogue. A full integration of these models into a publishable FORM External Information Model has not been conducted, but an assessment the use of a common format. However, an assessment of how the use on a common format for generating the BIM may support such an integrated EIM has been conducted. The suggested structure of the resulting integrated EIM is given in [formD11].

P11: Functionally related Building Blocks can be grouped together for the purpose of software release.

The BBs developed in the Trials were not purposefully grouped for release, but an attempt has been made to identify possible groupings and this is presented in the assessment of the trial models.

P12: Building Blocks must be released with documentation describing: the related business context in which the BB is intended to operate and the analysis of this context that led to the identification of the BB and the design of the Contracts it uses.

This has been implemented through the format defined in the Development Methodology [formD12] for the Contract Specification, the results of which can be seen in the on-line Contract Catalogue. Full traceability between the sub-elements of a Contract Specification has not been fully implemented however. A key link between Contract Specification and their Business Requirements is captured in the mapping between contracts and Reference Points identified in a Business Role Model. This mapping has been evaluated as part of the trial model assessment.

P13: The behaviour of a Contract and interactions with the behaviour of other Contracts on the same BB may be modified at deployment- or run-time. Where this feature is offered it should use explicitly defined business-rules.

Techniques such a policies and workflow have been used to support changes in behaviour at run-time. However a full model of how such techniques, characterised as 'Business Rules' in the Architectural Model, relate to elements such as Contracts, Building Blocks and Management Systems, has not been achieved, as discussed in [formD9]. The use of different techniques for implementing such Business Rule is evaluated only on an ad hoc basis as part of the Technology Selection Guidelines assessment.

#### 2.2.2.2 Assessment of Architectural Principles and Benefits

The Architectural Principles and associated stakeholder benefits are used as the basis of the evaluation of development aspects of Trial 2. However, these principles and benefits have been developed alongside the Logical Architecture over the lifetime of the project. Therefore, the principles and benefits themselves need to be assessed against industrial needs so that the industrial relevance of the evaluations can be better understood.

The approach taken to this assessment is to publish the principles and benefits on the project web site as part of an on-line questionnaire. Different versions of the questionnaire will be presented for each stakeholder (not all principles provide a direct benefit for each of the stakeholders). Project participants, other members of partner organisations and people from external organisations will be solicited to complete the questionnaire.

Broadly, the questions will be:

- For each architectural principle: is the principle clearly explained?
- For each stakeholder benefit: are the benefits associated with the principle clearly explained?
- For each stakeholder benefit: to what extent do you agree with the desirability of the benefit for your organisation?
- For each stakeholder benefit: to what extent do you expect the principle to provide the benefit?
- For each architectural principle: are there additional benefits an architecture conforming to this principle may bring your organisation?
- In general, are there additional principle you would seem as desirable or essential before making used of the ODF?

As the answers are fairly qualitative, where respondents are available, group sessions were established to attempt to draw conclusions from the range of answers.

#### 2.2.2.3 Assessment EIM integration

The ODF principles aim to achieve a technology neutral specification format for Contracts (P6). This was not fully achieved with contract interface signatures being represented in a technology specific format. However, all Trial 2 Contracts do possess a Boundary Information Model, representing the information passed via that contract. The ODF principles also aim to make such information models available externally to location of suitable Contract using information model matching and to promote the reuse of information models (P10).

The aim of this evaluation is to assess the extent to which a common External Information Model can be constructed from the individual information models developed for the separate Contract Sets within the separate trial working groups in FORM. The approach of this assessment is:

- to assess the extent to which each of the existing information models followed the EIM modelling guidelines,
- to assess where identical information objects were used in separate BIMs,
- to assess where similar information objects were used in separate BIMs and to characterise their differences,
- to assess what problems were found in comparing and resolving differences between the separate information models.

#### 2.2.2.4 Evaluation of Trial Models

The Trial Models are examples of Reusable Elements, such as Building Block Groups, Contract Sets and Reference Points, and a set of Management System Models, which represent how Reusable Elements can be reused within different application domains (albeit within the same overall business scenario). This assessment first attempts to characterise the different types of Trial Models in terms of the architectural meta-model. On the whole this characterisation simply quantifies the extent to which the individual Trial Models adhere to the meta-model and thus to the architectural principles. The characterisation is also intended, however, to make more apparent to the developers the relationship of their models to the architectural meta-model rather than thinking of the models only in terms of the Development Methodology Guidelines that were followed in developing the models. In the light of the meta-model relationships revealed comments will be gleaned on how the models address aspects of the meta-model that are not directly present in the Development Methodology Guidelines.

The evaluation of Trial System Models will be conducted via two sets of questionnaires; one assessing issues at the level of each trial work group and requiring consensus between its members, and the other assessing issues related to individual BBs, and requiring more detailed design knowledge of each BB. The former will be completed by one representative of each of the four Trial System workgroups, in consultation with the other members of the group, the latter will be completed by individual BB developers.

To distil further insight into the merits or otherwise of the architectural principles and meta-model in relation to the Trial Models, a mediated group session will be run with the respondents to discuss their responses and to attempt to identify any such insights or consensus views.

The following subsections outline the questions to be asked for both types of questionnaire.

#### 2.2.2.4.1 Assessment by Trial System Teams

This assessment is to be completed by trial working group leaders in consultation, if necessary, with other workgroup members.

#### 2.2.2.4.1.1 Characterising Trial Systems

The following information is sought to characterise the trial systems developed by each working group, and thus to provide a context for the evaluation of those trial systems with relation to the Logical Architecture:

- Identification of the BBs used in each Trial System, the Contracts they support and the interaction technology used in their implementation.
- Identification of the non-BB subsystems also present in the Trial System, the individual interfaces they offer and the interaction technology used in their implementation.
- An estimation of the relative proportions of the overall system functionality that is provided by the BBs and the Subsystems respectively.
- Identification of which Contracts are used only by software within the System developed by the workgroup, which may be used by entities external to the System and which follow existing standards.

#### 2.2.2.4.1.2 Evaluation by Trial System Teams

The following information is sought to evaluate the level of adherence of the models developed to aspects of the Architectural Model that were not explicitly addressed by the Trial development. In this way, we attempt to get a better picture of how well aspects of the Architectural Model emerged, that were not used in the immediate system development, but which are important to later reuse of models and BB software.

The questionnaire therefore attempts to make the following evaluations:

- Evaluation of the extent to which the Contracts designed by the workgroup use the same information model for defining their individual Boundary Information Models?
- A Contract Set is a grouping of related contract definitions, published together for the purposes of reusing the contract specification in a useful way. The generation of Contract Sets was not an aim of the System and BB development work in the Trial, and is not explicitly addressed in the methodology Guidelines. Therefore, suggestions for Contract Sets are sought, with an overall description of their purpose, which Contracts they would include and which further, as yet un-designed Contracts, may be needed to make the Set more reusable.
- A BB Group is a collection of BB grouped together for the purposes of reuse either within the organisation which developed them, e.g. a System Integrator, or as a commercially sold package. The generation of BB Groups was not an aim of the System and BB development work in the Trial, and is not explicitly addressed in the methodology Guidelines. Therefore, suggestions for BB Groups are sought, with an overall description of their purpose, which BBs they would include and which further, as yet un-designed BBs, may be needed to make the Group more reusable.

#### 2.2.2.4.2 Assessment by BB developers

#### 2.2.2.4.2.1 Characterising BBs and Contracts

The following information is sought to characterise the BBs, and thus to provide a context for their evaluation with relation to the Logical Architecture:

- Identification of the type of analysis object from which the BB design was derived.
- Identification of the Reference Points and Business Roles supported by the BB's Contracts.
- Identification of the number of information objects supported by the BB's Contracts and the number of operations each Contract offers.

#### 2.2.2.4.2.2 Evaluating BBs and Contracts

The following information is sought to evaluate the level of adherence of the models developed to aspects of the Architectural Model that were not explicitly addressed by the Trial development. In this way, we attempt to get a better picture of how well aspects of the Architectural Model emerged, that were not used in the immediate system development, but which are important to later reuse of models and BB software.

The questionnaire therefore attempts to make the following evaluations:

- Whether the functionality offered by a BB's Contract is more generic than the functional scope represented by the Reference Point and Business Role it supports, and if so what functional scope would be a more accurate match.
- What further new Contracts would be needed in addition to the BB's Contract to fully address the functional scope of the relevant Reference Point.
- For each Contract, the number of information objects

#### 2.2.3 Development Methodology Evaluation Plans

The FORM Business Process driven guideline was evaluated in primarily two ways. The first was a questionnaire completed by the FORM development team. This team comprised of FORM partner managers, management system designers, and management system developers. Follow up interviews were then conducted where necessary to gain greater understanding of the questionnaire results.

#### 2.2.4 Technology Selection Guidelines Evaluation Plans

The Technology Selection Guidelines of the FORM Open Development Framework is presented in details in FORM deliverable D9, section 5 [formD9]. It focuses on providing mechanisms for selecting one or more technologies for the implementation of specific applications for management systems. Selecting a technology is not a trivial task and will constitute a strategic decision for most organisations. Because the ODF aims to achieve technology neutrality, the objective is not to present and compare technologies but rather to raise awareness about the different criteria to be considered in the technology selection process. The precognition of a Building Block based system architecture means that a number of quality requirements must be met. These requirements, expressed mostly by the TMForum Application Component Team, play an important role in the selection process. Criteria for the technology selection process can be divided into three different categories. The first criterion is purely functional and corresponds to the set of functionalities (business processes) that the management system is to fulfil. The second criterion is associated with the FORM Architecture principle and relates essentially to non-functional system requirements. Finally the last criterion in the technology selection process is more external to the technology features and focuses on corporate strategy, developer's training, technology trends, etc. An organisation should be aware that technology decisions must be part of a broader and long-term strategic plan. Performing a selection for implementation and integration technologies requires an organisation to marry a number of internal business requirements with a myriad of vendor attributes that relate to both performance as well as the ability to effectively provide long-term value to the organisation.

Within the FORM consortium, more than 80% of the partners chose J2EE technology framework components as their prime implementation technology (EJB, Servlets, JMS, etc...). Based on the technology architecture questionnaire [annex B], these choices can be explained by two reasons. First, functional requirements were often identified as playing an important role. For example, the need for inter-domain communication of distributed management systems. J2EE provided means to ease inter-domain communication, using RMI over HTTP for instance. The other main factor is a rather "non technology feature-based criterion" and correspond to the initial willing by partners to gain experience in this "trendy" emerging technology.

From the beginning of the project it was clear that partners wanted to get experiences with new and emerging and likewise 'trendy' XML technologies. The important XML Schema recommendation was approved as a W3C recommendation May 2001, i.e. one year after the FORM project started. In the same way XML tools was mainly available in beta versions. As mentioned above the choice of the J2EE technology and underlying Java Development Kit (JDK) resulted in partners selected XML tools that was recommended for the selected JDK or Java platform.

The following sections therefore aims at presenting various findings on two specific technologies that were used within the project, namely the lifetime of J2EE and XML. Again these two sections should not be seen as prescriptive in anyway, the aim is therefore not to dictate the use of J2EE and XML for implementing BBs but rather to present the experience gained in evaluating these two technologies.

#### 2.2.4.1 Evaluation of use of Policy-based Techniques

The use of policy-based techniques to provide run-time configuration of BB behaviour is suggested in the Logical Architecture, but not specific guidance has been provided on how to do this. Therefore the assessment of the use of policy-based techniques in Trial 2 is restricted to an assessment of the ad hoc use of policies by individual BB developers. This was conducted using the questionnaire in annex B.

#### **3** Trial 2 Evaluation Results

This part will present trial evaluation results (technology and test cases) and what forum work being impacted by this work and how

#### 3.1 Operational Requirements

#### 3.1.1 Test Cases Results & Conclusions

This section gives a summary of the findings for the test cases. Details regarding the test-cases can be found in Annex A.

#### 3.1.1.1 Fulfilment – VPN Test Cases

Initial integration was done and demonstrated at the M3 review in a distributed environment and all test cases were conducted successfully in Copenhagen in December 2002, where the four building blocks from four partners were integrated.

The main problems encountered during integration were of the following nature:

- Agreeing on naming: Each of the participating EJBs has a number of configuration and deployment files, which describe JNDI names of the EJB, which plug-ins to use, etc.
- Agreeing on location of naming server: The EJB's client locates the EJB using a JNDI naming server. The location (IP address and port) of the naming server must also be agreed.
- *Location of classpath environment variable:* On the machine on which the JBoss server is running, the classpath on location of naming server must be set-up correctly to avoid shadowing problems, i.e. that jar file A implements a certain interface and jar file B also implements the same interface. If the EJB wants to use the implementation in jar A, but if jar B is first in the classpath.
- *Test bed configuration:* Minor problems were also encountered in the complicated test bed regarding some of the test-tools for visualisation.

We consider therefore that most integration problems have been related to configuration of EJB server platform. It is also important to note that only one EJB server platform (JBoss) as been used by all partners. This fact certainly avoids platform interoperability problems.

However we estimate that correction of the integration problems encountered required less resources that would have been used if we had developed the functionality using a non component-based technology, e.g. C++. The J2EE platform delivers a number of services. Using a non component-based technology would have force us to agree on which third party libraries to use for each of the services or develop them ourselves. We did not encounter any problems, which needed further investigation, but the tests conducted were focused on a normal-path through the system and further testing would require examination of error-paths.

One main factor for successful integration in projects where components are developed by geographically distributed partners from different organisations is well-defined interfaces. Building Blocks and contracts have been developed in respect of the development methodology proposed by the ODF, which supports ODF principles. Therefore, main conclusion regarding integration process and use of ODF is that integration of complex component-based system can be eased using well-adapted methodology.

In the same way the development methodology supports a top down approach allowing integration of user requirement at the beginning of the development process. Trial 2 allowed to validate user requirements, captured at beginning of the project, have been fulfilled by developed Building Blocks and corresponding integrated system.

#### **3.1.1.2** Fulfilment – IES Test Cases

#### The use of schema/DTD for SLA negotiation.

As part of the F-IES trial system three DTDs were created representing each of the BB contracts (SLAHandling Service, SlaNegEng, and the SLARepository). These DTDs proved very useful as part of the individual pre-trial tests conducted by each partner. Test XML documents were exchanged between the partners and validated against the agreed DTD. Therefore most of the building block functionality testing was completed before the trial.

#### Applicability of SLA negotiation sequence diagrams

A prototype SLA Negotiation Engine was developed as part of the trial system to test the use of policies as part of the SLA Negotiation process. The SLA Negotiation Engine proved capable of validating the SLA requests send by the IES Customer using predefined policies.

#### The scalability/usability of technology mediation in the SLA repository

The SLA repository was implemented by using the Q3Ade Technology Gateway. XML DOM parser like functionality was implemented and scripts were written to convert the XML encoded SLAs. The script approach proved very useful during the implementation, as changes to the DTDs were easily adapted. Also during integration some handshake problems were quickly solved by minor changes to the scripts. As expected some functionality proved to be too performance critical for the use of scripts and would therefore have to be replaced by compiled code for a real-time environment. The Technology Gateway therefore now implements support for transparent migration of scripts to C++ code. In other words function calls can now be made to C++ code or scripts seamlessly.

#### Platform support for SLA negotiation and SLA handling components

SLA negotiation was a standalone component and therefore required no platform support. The SLA handling component is a service running on the platform. It needs to be registered in subscription and needs to be started by the user within an access session of the platform. Entries in the subscription database are required before the service can be used.

#### Integration between SLA negotiation and SLA Handling

The communication between SLA negotiation and SLA Handling Service is done via TCP/IP by transmitting XML documents which need to be parsed at both ends. To support this, valid schema definitions are necessary to ensure a common information model on both sides. XML DOM parser like functionality was implemented and scripts were written to convert the XML encoded SLAs.

Further investigations can be undertaken in the following areas:

- Enhancement of the functionality of the SLA Negotiation Engine so that several parameters can be negotiated flexibly.
- Administrative GUIs are needed to configure the building blocks on-line. In the trial this was done manually.

- The different components for subscription, SLA negotiation and accounting were not harmonised, for example, in their tariff usage. This needs to be made consistent across the whole functionality of the underlying information models inside the databases used.
- Integration with VPN-SC

#### 3.1.1.3 Assurance Test Cases

#### **Production of Assurance Configurations**

The "Production of Assurance Configurations" test was carried out using simplified SLAs containing just one or two metrics that could be easily calculated. The system successfully produced and distributed the configuration policies required by the other components of the system. No serious problems were encountered in the implementation of this test case although certain deficiencies in the information model being used were identified.

In particular it was found in certain areas that associations, necessary to traverse the information model successfully, were missing and had to be added. Another more difficult problem was the way in which the CIM Object Manager (CIMOM) used in the trial dealt with references to CIMOMs on other hosts. It was found that by default a CIM Object Path object does not have it's host name component set to any value. Also there is no way to transparently access multiple CIMOMs simultaneously. Therefore for the purposes of implementation it was found necessary to define the points in the information model where other hosts could be referenced. This area will need to be further addressed in the future.

#### **Service Monitoring**

The results of the "Service Monitoring" test case can be presented in two parts. After configuration the Server Monitors performed as expected refreshing metrics at specified intervals using the calculation method specified in the configuration. There was, however, a problem with the Perl DLL that was used to perform the calculation of the metrics that caused the system to stop working on several occasions. It was decided therefore that different methods of performing these calculations should be investigated to make the system more stable.

Unfortunately at the time of the trial the functionality of the Performance Monitor was incomplete. Therefore only the ability of the Performance Monitor to gather statistics and events from the various server monitors was evaluated. While statistics were gathered from the Server Monitors successfully issues did arise concerning the information models used to store them. In particular there was some discussion over the best method for storing previous metric values (for trend analysis etc).

#### Service Violation

For the purposes of the "Service Violation" test case an SLA was submitted to the system with a threshold that was known would be violated. For this test the workflow engine was also setup to start a process on the receipt of these events. In this way we hoped to simulate part of one of the MCG processes, namely the Billing-Assurance process.

As before the Server Monitor calculated the statistics successfully. When compared to the specified threshold the Server Monitor then threw an event (in the form of a JMS message). This message was then received and interpreted correctly by the workflow engine. During this test it was identified that further information may need to be stored in the event to properly identify the context from which it was raised. Also, although it did not have a direct bearing on the test case another area that needs to be addressed is the quantity of events produced and when notification should be suppressed.

#### Workflow implementation of Business Processes

The "Workflow implementation of Business Processes" test case successfully executed the two variations of the assurance business process and raised one important issue concerning the specification of the control and data flow in UML. The UML v1.3 activity diagram is not able to specify correctly the dynamic invocation of multiple instances of the same activity and these instances subsequent synchronisation. We partly specified this control flow requirement using the "Foreach Configuration in List" activity but the subsequent synchronisation of the instances was left unspecified in the diagram. This control flow requirement was hand coded into the control flow rule code.

There were no significant problems encountered during the test. Further investigation is necessary in the mapping of activity diagram to process control and data flow rules, in particular, how best to deal with cases where the activity diagram cannot specify the desired control flow. Future work will look into providing extensions to the UML activity diagram to enable specification of these more complex control flow structures. Also further implementation of coarser grained business processes needs to be carried out to demonstrate the workflow frameworks full potential.

#### GQIPS

The four GQIPS Test cases "B3 Setting", "Single domain service negotiation", "Events subscription and notification" and "multi-domain RAR Negotiation" were executed successfully. The GQIPS System was integrated with an existing policy server in the Broadcom Test Lab.

The GQIPS was able to extract a network topology of the testbed routers. Activation requests were received from the service level assurance components. These requests were processed via negotiations to another domain and requests were responded to correctly. The routing took place over existing test bed routers and no problems were encountered.

#### **3.1.1.4** Billing Test Cases

One of the objectives of this trial was to deliver IP services, mediate usage data for the service usages into E-IPDRs and perform rating and discounting on the E-IPDRs. Each of these objectives were fully achieved during the trial. However, the trial did raise the issue of guaranteed delivery of E-IPDRs to the Rating Bureau Service (RBS). The reliability of any such system would heavily depend on the assurance that all usage records are passed for rating and subsequent charging/billing. The failure to guarantee that all records are delivered could result in substantial revenue losses for the IP service providers. Other issues that were not addressed include:

- the security of the records passed between the relevant domains
- stress testing of the RBS for large numbers of records
- real-time capabilities of the RBS
- > measurement of overheads due to usage data collection, mediation and delivery
- ➤ the communication channel between the RBS and the SLA management system
- ➤ the traceability of the audit trail generated

These issues were identified prior to the generation of the trial test cases but were classified as out of scope for the trial. The main objective of the trial was to validate the use of the IPDR as a means for passing usage data between domains and the subsequent investigation of the validity of the addition of a Charge Element to the IPDR schema. The granularity of the information contained in the E-IPDR supported flexible charging encompassing accurate quality of service related discounting.

The Federated Mediation Adaptor (FMA) forms a key component in mediation of aggregated service usage and service value chain. The tests carried on it proved that it must be robust during operation and must maintain consistency of usage data.

Aggregation of a composite usage session and binding it to a set of E-IPDR documents proved to be more difficult than we had earlier envisaged. IPDR specifications do not address this requirement very clearly. Nor do they provide practicable means to implement aggregated service mediation. The means that were devised to meet this requirement were based on the idea of having **parentSession** in Master IPDR Schema. They were successfully tested and proved to be practicable.

#### 3.1.2 Operational Requirements Tracking

After the execution of the 20 test cases constituting Trial 2, by the 4 Test Teams, the following can be summarised on the linkage to the operational requirements captured in WP2.

In total 184 operational requirements were recorded at an early stage in the project of which 118 requirements (64%) was associated with a Trial 2 Test Case. Figure 3-1

**Coverage of Test Cases** 



Figure 3-1: Requirement Coverage of Trial 2 Test Cases

That the operational requirement coverage is ~65 % is not unexpected as the Building Block developments where to accomplish three elements: 1) Service as an experiment using the FORM Methodology and Framework; 2) Investigate supporting technologies and provide proof-of-concept implementations and integration; and eventually provide exploitable elements for industrial partners. Strict adherence to requirements where not prioritised by the development teams as the aim development-wise was prototyping and investigation.

Of the 118 associated requirements 72% were fulfilled/partly fulfilled by actual trial implementations and an additional 9% were concept-wise fulfilled/partly fulfilled - indicating that the BB designs and could have fulfilled/partly fulfilled the requirement if implemented. 19% of the associated requirement were not fulfilled by a test case, as shown in Figure 3-2.

Distribution of Requirements Covered by Test Cases



Figure 3-2: Distribution of requirement coverage

Figure 3-3 provides a more detailed look at the 'Fulfilment degree' for all test teams regarding both 'concept-wise' fulfilment (i.e. is the requirement covered by the BB concept/specification, but not implemented) and 'implementation-wise' fulfilment (i.e. is the requirement covered by a trial implementation). The majority of the requirements are implementation-wise Partly Fulfilled, which again is not unexpected. Going over the implementation-wise Partly/Not fulfilled requirements indicate that, often only fragment of the full requirement has been implemented. This relates to two factors: 1) Limitations in time and resources during implementations; 2) implementation teams have prioritised 'breath' to 'depth', i.e. do more experiments than implement for completeness. That a large number of requirements are 'Partially Fulfilled' implementation-wise relates to the fact, that several requirement were written in 'high-level' terms and in spans multiple more detailed requirements.



Figure 3-3: Details on concept-wise and implementation-wise distribution

Two of the main groupings - introduced early in the project - of the operational requirement were on:

1) 'Expressed-By', i.e. who expressed the requirement. This could be either End Customer (EC) or Inter-Enterprise Service Provider (IESP), but also a few requirements by Management System Provider (MS) to Management Component Developer is included.

2) Functionality Block.

GE: General

SA: Service Level Agreement

IA: Information Service Assurance (QoS)

QA: Network Provisioning Assurance (QoS)

CB: Charging and Billing

SC: Security

IE: Inter-Enterprise System Provider to Management System (Outsourcing)

MS: Management System Provider to Management Component Developer System (Outsourcing)

Examining these groupings the following picture emerges. Figure 3-4 shows the requirement coverage by 'expressed-by' category. A similar picture as for all requirements emerges, but with a slight difference between IESP and EC expressed requirements. Approx. 50% of the EC requirements were considered in a test case and slightly less actually implemented and approx. 70% of the IESP requirements were of a more technical nature and often directly related to the areas of interests for the development teams, wherefore a higher percentage was covered.



#### Requirement Coverage pr. Expressed-By Catagory

Figure 3-4: Requirement Coverage of Trial 2 Test Cases by 'expressed-by' category

Figure 3-5 shows the requirement coverage by 'functionality block'. More CB (Changing & Billing) requirements was collected but mostly the requirements relating to cores elements in a federated Billing/Accounting service was pursued for the trials and not many of the secondary requirements (which would be covered in a production system). The same applies for other categories as well.



**Requirement Coverage pr. Functionality Block** 

Figure 3-5: Requirement Coverage of Trial 2 Test Cases by 'functionality block's

All in all a large number of the initial requirements were addressed by a test case and slightly fewer were validated by an actual trial implementation. The following paragraphs conclude on the requirement impact from the individual test teams.

#### 3.1.2.1 Fulfilment-IES (Test Team 1) Test Cases

SLAs are becoming increasingly significant in selling services and so flexible on-line negotiation was an important requirement for the trial. The functionality required led to the generation of the trial test cases so that on-line dynamic negotiation of SLAs could be tested and validated in the trial test cases. The ability to offer service subscriptions via on-line SLA negotiation is a critical element in today's competitive market and the trial showed that it is possible to build individual building blocks that can operate together to meet this requirement. This addresses requirements EC-II.02, EC-II.03, SA.II.04, SA-II.05, etc.

In an e-business environment, such as that represented by FORM, it is important that services can be rapidly and flexibly subscribed to on-line. It is also important that customers can select the QoS parameters they desire for the service. The trial showed that the QoS of the service being ordered could be selected by the customer. In this case bandwidth, and thus the usage performance, could be determined by the customer. This addresses requirements EC-II.08 and QA-I.01.

The trial showed also that SLA negotiation and conclusion can be undertaken in a one-stop-shopping environment and that various services can be ordered via negotiation of one SLA. This addresses requirements EC-II.07 and SA-IV.08.

#### 3.1.2.2 Fulfilment-VPN (Test Team 2) Test Cases

Of the originally collected requirements only few were specific to VPN, many functionally addressed a generic Inter Enterprise Service concerning outsourcing, security or establishment. The requirements were very broad and basic such as dynamic establishment of service, dynamic modifiability of service, etc. Main reason is that the project started from a very broad vision of Inter-Enterprise service to be supported by an IESP.

The system design of the Trial 2 system and the BBs within the Trial 2 system was based on concepts and ideas from standards: from ITU-T [M.3108.1], [M.3108.3], [M.3208.1], [M.3208.3]), IETF [IPSec Configuration], [IPSec Policy]) and the QBone ([Internet2 QBone]) initiative. During the analysis-phase the requirements served as guidance on the functional level, whereas in the design and implementation phases strict adherence to requirements was not prioritised.

The planned test cases for Trial 2 dealt with the most basic VPN functionality as integration had focus, and it was needed, to enable testing of the functionality in the individual EJBs. However this simple functionality designed and partially implemented for Trial 2 was traced back - using the web-tool - to many of the original requirements of which a large number were addressed.

Anyway, operational requirements, as defined initially, allowed assessment of provision and activation of service in a B2B context.

#### 3.1.2.3 Assurance (Test Team 3) Test Cases

• Production of Assurance Configurations

For the most part the requirements to be met by this use case were addressed. In particular those relating to the SLA and Service Architecture information models, IA-I.02 and IA-I.04, and the requirement for Policy Construction, IA-II.06, were fully addressed although it was identified were further improvements can still be made.

This leaves two requirements that were only partly validated by this trial. The first of these, IA-II.05, relates to the subject of SLA Admission. Currently the SLA is only checked to ensure that it is well formed and relates to a known server. Further work needs to be done in this area to deal with multiple SLAs and the conflicts that can arise between them.

• Service Monitoring

Three requirements were addressed by this trial. The first, that the management service should be able to cover several geographical locations (EC-II.22), was addressed by the production of the Server Monitor components and the extraction of statistics from them by the Performance Monitor.

The other two requirements, EC-II.30 and IA-II.10, both relate to the production of service statistics and notifications. These requirements were only partly fulfilled as although the statistics relating to the service were collected and could be presented in debug format they were not placed into a proper format. This issue will be addressed with the finalisation of the report generator component.

Service Violation

The most important requirement to be met by this test case was the "QoS Monitoring" requirement, IA-II.07, and this was indeed fulfilled by the trial in the way in which both the Server Monitors and the Performance Monitor could calculate/collect statistics relating to the service performance and determine when thresholds had been broken.

The other requirements, EC-II.29, IA-II.08 and IA-III.12, all relate to the production and reception of asynchronous notifications or events. Mechanisms for fulfilling these requirements, through the use of JMS, have been identified and tested to a certain degree although further testing needs to be carried out before these requirements can be considered properly fulfilled.

• Workflow implementation of Business Processes

As the workflow framework is seen as a platform service and not a Building Block, the requirements it addresses belong to the Generic Framework Requirements. These requirements are general software engineering requirements and are related to the separation of Process from Entity. The workflow framework encourages a clear separation of process from entity, which produces a more flexible and scalable system. These generic requirements did not change throughout the project.

• Customer Login and Validation. Customer specific information is XML formatted, and the web application uses 'standard' XML techniques to make such information available for the application

To save programming resources by implementing an open source XML parser software, it was required to update the selected open source Web server to the latest beta version, which resulted in lack of portability of the Web service to older Web servers. One problem which came up was that the Web server itself included an XML parser which it uses for configuration, so trying to install a new one resulted in class conflicts between Java classes. The problem was later recognised by the Web server provider and a solution was provided in a later beta version.

Requirements impact: *It should be possible to define counter actions for security violations or violation attempts.* This was a very general requirement but also very important since the client Web page returned to the customer after successful login, was customer specific and should only be shown to the right customer. It was addressed to examine the benefit of using XML to mark-up information about customers in the IESP customer base.

• Reporting Template Completion.

'Standard' web programming techniques are used to define 'browser specific' dynamic client pages (templates) to present customer specific information and selected report data. 'Standard' web programming techniques are used to enable generation of 'browser specific' dynamic menu pages with customer specific information and menu options, and customer menu selections are used to initialise search filters for collection of reporting data. Web application detects missing customer client input, and adds help-messages to the customer menu page.

The use of Java Server Pages enabled changes in the client web pages to be implemented with very little programming effort. Again to take advantage of this new technique required that the latest versions of the Java platform used by the Web server was used, so again there was a lack of portability to older Java platforms.

Requirements impact: End-customers can receive information about performance and usage of endcustomer equipment. This was also a general requirement and it was seen as a requirement for a very flexible solution that could be customised to specific end customer demands with very little programming effort.

Status and statistics. IESP requires management functions, which enables end-customers to get information regarding status or statistics from managed equipment on-demand and on schedule.

Since most existing reporting services support on schedule reports this was addressed by focusing on on-demand reports where the customer uses a menu to select only that information which are required here and now by the customer.

Report Customisation

Information used for customisation is XML formatted, and 'Standard' XML techniques are used to add or change information. Translation to customer client browser mark-up language, can be done by web application or by client browser.

The use of open source software for translation of XML documents also required the use of the latest beta version of the Web server. The selected XSL Translation technique enabled translation of XML documents into other XML documents with very little programming effort, but it did not support the use of variables to add dynamic information like e.g. counters to be used in reports. As a result of that all dynamic information had to be added to the XML document before the translation, or handled by using temporary XML documents.

Requirements impact: Presentation of service information. MSP requires components that support presentation of service information on end-customer terminals.

Since most existing reporting services support presentation on Web browsers this requirements was addressed by focusing on the Web services ability to detect if the customer browser supported Wireless Application Protocol WAP or XHTML which are used in the latest version of WAP. In addition to adapting to the protocol version used by the customer, an attempt was made to take advantage of a new technique to create scaleable graphics that can be scaled to match the dimension of the customer terminal display. The use of Scaleable Vector Graphics (SVG) also enabled reuse of the XML translation software since SVG is an XML based specification language. The test showed that XML to SVG translation might be handled by the customer's own browser if it has the correct SVG software plug-in installed. A major advantage of SVG is that it reduces the size of the graphics data that needs to be transmitted to and stored on the customer terminal. Another advantage is that SVG enable graphics to be interactive e.g. a customer can zoom in on graphics e.g. on a small display on a mobile terminal display.

• GQIPS Test Cases

The GQIPS Trial 2 Test cases consisted of "B3 Setting", "Single domain service negotiation", "Events subscription and notification" and "multi-domain RAR Negotiation". For trial 2 the GQIPS system was integrated with an existing policy server and mediation device. These test cases took place on real router with the Broadcom test lab. The Assurance Trial 2 (in Dublin, December 2001) demonstrated the integration of the GQIPS with the Assurance group and the GQIPS test cases demo on the test network. GQIPS was also integrated in the VPN (test team 2) trial in Copenhagen, December 2001.

Of the operational requirements which were originally collected, all remained relevant for the test cases. It was noted that nearly all were fulfilled at design phase. The requirements were used as guidelines at design phase. The ideas of the Qbone initiative were another driving factor at the design phase of the GQIPS. Those that were not fulfilled at implementation phase, were due to resource limitations.

#### 3.1.2.4 Billing (Test Team 4) Test Cases

Standards (e.g., TMForum, ETSI, IRTF/IETF, IPDR) and recent trends in the industry (Billing solution vendors, etc) were studied to capture the initial requirements. Study of billing business process in general also helped greatly in capturing requirement. Difficulties did arise when we attempted to use useful concepts from organisations such as IRTF/IETF and then tried to provide feed back. Here the research goals and background appears to be the main reasons. IRTF/IETF saw billing processes from point of view of the Internet engineering, and terms such as use cases and business model did not play as important a part as it was in FORM project.

The main driver for addressing the subset of requirements that were met was current telecommunications industry requirements for IP service accounting. As the new IP-based services market expands rapidly, the upsurge in the level of B2B interactions creates new service requirements in the areas of customer service access, security, billing and Quality of Service (QoS). An important feature of the new environment is the creation of composite services (service sets) created from the integration of services provided by ISPs, Virtual Private Network (VPN) and application service providers, as well as backbone operators. In such an environment B2B requirements can no longer be economically met through the provision of non-dependent standalone services. A critical factor in the growth of this environment is addressed by the requirements CB-I.01, CB-I.04, CB-I.08 etc. These requirements describe the need for a standard for the accurate exchange of usage data between these various providers. The trial was concerned with validating the IPDR against these and other related requirements.

Requirements captured and listed under **Abnormal Conditions** did not have considerable impact on the design of BB. However, many of the requirements captured under **Dynamic Functionality** proved to be important and influenced BB design.

Those requirements that did not have considerable impact on BB design did play the role of a set of guidelines with which we tracked the course of development work and later on evaluated the result.

#### 3.2 Development Aspects Evaluation Results

#### 3.2.1 Logical Architecture Results

This section provides the evaluation of the development aspects of Trial 2 against the ODF's Logical Architecture.

#### 3.2.1.1 Assessment of Architectural Principles and Benefits

The evaluation of this questionnaire was intended to help improve the presentation of the ODF's architectural principles as presented in [formD9] and to structure them to better reflect relative priorities of respondents.

The results presented here reflect the responses from 8 of the partners in the FORM project. The respondents characterised their organisations as the following ODF stakeholders:

- Independent Software Vendor 2 respondents
- System Integrators 3 respondents
- System Integrator and Independent Software Vendor 2 respondents
- System Integrator and Service Provider 1 respondent

The evaluation presented here is the result of the responses to the questionnaire, and follow-up group discussion conducted at Waterford Institute of Technology on the 5<sup>th</sup> February 2002.

Though it had been intended to get some external respondents to the questionnaire, there was strong feedback from, the internal respondents that the questionnaire in its current form was too long and complex to be completed by external people. The questionnaire on principles and benefits has not, therefore, been completed by anyone outside of the project, and the mechanism for getting such feedback is being re-evaluated.

#### **3.2.1.2** Clarity of Principles

In response to whether or not the respondents found the principle statement clear (see Figure 3-6) only three of the principles were found to be clear by everyone, namely P4, P6 and P11, while two (P3 and P13) were not clear to the majority of respondents.



#### Figure 3-6: Percentage of respondents that found principle statement clear

Of the principles that were not found to be so clear the following recommendations were captured.

- A clear definition of 'Building Block' needs to be given, in particular with relation to P1, P2 and P3.
- A clear definition and perhaps examples needs to be given of the term 'system management' with relation to P3.
- The language used to described the use of business rule in P13 is too terse, and needs more explanation.
- A clearer definition of what is meant by 'technology specific' and 'technology neutral' needs to be given with relation to P7.
- A clearer definition of what is meant by an 'explicitly defined transform' needs to be given in relation to P8.
- A clearer explanation of what is meant by 'atomic units of deployment' needs to be given in relation to P2.
- A clearer explanation of what constitutes a 'business operation' is needed for P5.

#### 3.2.1.2.1 Importance of Principles

Figure 3-7 gives an overview of the relative importance attached to the different principles.



**Figure 3-7: Importance of principles** 

If we unscientifically assume a linear scale from 0= 'irrelevant' to 3='essential' in order to get a rough picture of the 'average' importance between the principles, we come up with the graph in Figure 3-8. From this we can see that P6 can be interested as the most important principle, followed by P10, while P13 can be interpreted as the least important, though this may reflect the lack of clarity revealed for this principle.

Consideration will be made to re-ordering the presentation of the principles in future publications.



#### Figure 3-8: 'Average' importance of principles

Further feedback was obtained on disadvantages perceived for particular principles, the major ones are summarised below:

- The development overheads related to following these principles when developing green field management systems, including the introduction of new architectural concepts (P1, P6)
- Loss of control of source code when buying BBs off the shelf, especially for debugging (P1, P4)
- The difficulties in designing BB without dependencies (P2, P3)
- Problems related to the interworking of technologies are non-trivial (P6)
- Difficulties in designing and implementing with the technology neutral contract format (P7)
- Potential complexity of technology transforms between technology neutral and technology specific form, and identifying the ROI involved in taking this approach (P8)
- Problems were envisaged if there is not good coupling between Contract Specifications and their information models (P10)
- Costs involved in fully documenting BBs plus the exposure of critical IPR when publishing this documentation (P12)
- Increased complexity in using explicit business rule to modify behaviour of BBs (P13)

Based on these comments the following recommendations are made to any future revision of the Logical Architecture:

- Test specifications and test certificates need to be packaged with BBs and their documentation.
- Clear models for determining the ROI needed to justify additional costs of developing BB based system, publishing BB documentation and controlling behaviour via Business Rule.
- Improved design guidelines for assessing and reducing the dependencies between BBs.
- The development of a technology neutral contract specification needs to be shown to be practicable and cost-effective. It must be designed:
  - to enable strong management of the link between contract interaction specifications and contract information specifications,

- o to support simple mapping to target technology specification,
- to support the development of gateways that separate technology interworking from model interworking.

#### 3.2.1.2.2 Clarity of Benefits

In response to whether or not the respondents found the benefit statement clear (see Figure 3-9), 25 of the 40 benefits statements were found to be clear by everyone and none were unclear to the majority.

Comments related to the clarity of benefit statements include a better explanation of 'interface type' needed for P4-B2 and the similarity of benefits for P6, P7 and P8. The comments will be taken into account in future revision of the principles text, and consideration may be given to restructuring the principles when presenting them against the benefits.


Figure 3-9: Percentage of respondents who found benefit statements were clear

### **3.2.1.2.3** Importance of Benefits



Figure 3-10 gives an overview of how the respondents rated each of the benefits in importance.

**Figure 3-10: Importance of benefits** 

If we assume a linear scale from 0= 'irrelevant' to 3= 'essential' in order to get a rough flavour of the 'average' importance between the principles, we come up with the graph in Figure 3-11. This shows that no benefit is thought to be less than 'important'.

			P13-B2(SI	3-B3(SI) )
				3-B1(ISV) 2-B1(SI)
			P11-B1(ISV)	<b></b> P10-B6(
			P10-B4(SI	D-B5(ISV)
			P10-B3(IS	V)
			P10-B2(IS	
			P9-B3(SI) P9-B2(ISV	
			P9-B1(SB)	) ·B5(SB)
			P8-	·B4(SB) ·B3(SI)
			P8-B2(SI)	
			P7-B4(SI)	)
			P7-B3(SI)	
			P7-B1(SB)	)
			P6-B2(SI) P6-B1(ISV)	
			P5-B3(SI) P5-B2(ISV	·)
				B1(SB)
			P4-B3(SI)	
			P4-B1(ISV)	
			P3-B1(SI)	
			P2-B2(SP) P2-B1(SI)	
			P1-P2(SI) P1-B1(SI)	
0.5	1	1.5	2 2.5	3

Figure 3-11: Relative 'average' importance of benefits

The respondents were also asked for each benefit if the thought the benefit would be delivered by adhering to the corresponding principle. The results are shown in Figure 3-12 as the percentage of those who expressed an opinion who thought that the benefit would be delivered.



# Figure 3-12: Percentage of those who thought the benefit would be delivered via using the principle

Several comments were provided where a respondent considered that the benefit would not be delivered by the principle. In most cases the comments indicated that the principle would be insufficient by itself to deliver the benefit, and that other conditions would nee to be considered, such as:

- organisational and cultural attitudes towards software reuse within organisations,
- the maturity of the market for reusable components
- the lack of maturity of some of the concepts and model of the ODF

Some additional benefits were suggested by the respondents for some of the principles, notable ones being:

- The BB principle (P1) enables small, specialised ISVs to operate in the management software market.
- Finer grained management of systems would be possible than at present (P3)

- Domain knowledge is de-coupled from specific technologies (P6)
- Potential for re-packaging legacy systems as BBs

#### 3.2.1.3 Assessment of EIM Integration

This section provides an analysis of the various Boundary Information Models (BIMs) that were provided within the on-line Contract Catalogue prepared for FORM deliverable [formD11] as part of the system model for Trial 2 of the project. The aim of this analysis is to assess the extent to which a common External Information Model can be constructed from the BIMs developed within the separate trial working groups in FORM. The approach of this assessment is:

- 1) to assess the extent to which each of the existing information models followed the EIM modelling guidelines provided for D8
- 2) to assess where identical information objects were used in separate BIMs
- 3) to assess where similar information objects were used in separate BIMs and to characterise their differences

A suggested structure for an integrated EIM covering all the Trial 2 Contracts is given in [formD11] and reflected in Figure 3-13.



Figure 3-13: Suggested Integrated EIM for Trial 2 Contracts

Overall, the following can be observed about the how the information contents of the various Contracts developed by each Trial workgroup is reflected in the structure of the EIM:

- The IES Fulfilment Contracts all use the same SLA model, but otherwise contain mutually exclusive information models.
- The IES VPN Contracts all have largely mutually exclusive information models, except for some common IOs shared by VPNProvisioning BIM and the VPNService BIM (not shown in EIM figure).
- The Assurance Contracts exhibit a lot of sharing of common information models.
- The Billing Contracts all use exactly the same model.

# 3.2.1.3.1 Conformance to EIM modelling guidelines

Most BIMs conformed closely to the modelling guidelines, with the following exceptions:

- One of the assurance model group's BIMs contained classes with attributes with complex types, which should be remodelled a separate classes.
- Several BIMs had classes that included operation/method definitions. Such definitions should be confined to the contract interface specification with the information model only containing specification of information passed over a contract. However, for the purposes of BIM comparison and development of an integrated EIM, the operations definitions can simply be ignored.
- The attributes types used were not wholly consistent and in some cases were absent, in which case some form of string is assumed. A better defined type set is required.

Little use was made of constraints applied to associations between classes or to values ranges for attributes, this is important information for comparing and resolving semantic issues between information models.

#### **3.2.1.3.2** Use of Identical Objects

Several BIMs used identical information object, this being a favourable outcome of close co-operation within the trial working groups and also the outcome of information modelling work in the subscenario teams which spanned working groups.

As a result the same SLA definition is used for the Fulfilment IES Contracts and the Billing models when analysing potential Fulfilment-Billing scenarios. Interestingly however, there are two different UML representations of what is essentially the same SLA information within the Fulfilment IES Contracts. The differences centre on how explicitly the associations between Information Object, including aggregations, are modelled. For the SLA Negotiation Contract the model reflects the structure of the XML document used to exchange proposed SLAs and therefore 'container' objects are defined as information objects which have aggregation association to their constituent parts. The essentially identical SLA model used in the SLA Repository Contract reflects the structure of the underlying GDMO information model use to generate the interfaces and its internal representation. Here, there were few objects that only represented containment, with most objects represented specific information as well as possessing aggregation associations to other information objects. Modelling information was not lost however, since the aggregations themselves were labelled, replacing the need for explicit container object in a UML diagram, and resulting in clearer, more readable diagrams.

It is therefore recommended that where possible, information object that merely represent relationships between other objects should not be used in UML class diagrams, with the association being labels with a suitable identifier instead.

It is also noted that contracts in the Billing groups all refer to exactly the same BIM, showing good use of commonly defined information objects, is this case based on existing IPDR standards. Similarly the information models widely shared by the Assurance Contracts are based on information models taken from the DMTF's CIM. This shows how existence of a standardised information model assists in parties agreeing on new, extended information models.

### **3.2.1.3.3** Use of Similar Objects

This initial analysis has revealed no specific inconsistencies between information objects in different BIM. However there are cases where this is due to weak typing in one BIM compared to another. In particular, the SLA definition used in the IES Fulfilment group is aimed to be very flexible, since it must accommodate SLA terms that cannot be known in advance. This is handled by supporting later binding of additional SLA template and SLA negotiation policy information in the SLA Negotiation BB, where the SLA information is defined. This SLA therefore uses name-type-value triples to support these terms at the SLA negotiation related Contracts. Other Contracts that use terms from the SLA, however, have more strongly typed definitions. Though these are not inconsistent with the SLA definition, due to the latter's weak typing, they must be reflected in business rule models accompanying the SLA negotiation BBs.

#### **3.2.1.3.4** Summary of the EIM assessment

An initial analysis of the BIMs produced for the Contract Catalogue in D8 shows that the project seems relatively successful in avoiding incompatibilities between information objects used in different Contracts. This due to close team working within the Trial 2 groups, but also as a result of MCG scenario activities between those groups.

The analysis also has shown some shortfalls in the process of developing information models in a common format. This points to the need for clearer guidelines and some worked examples.

It is also clear how the use of standardised Information Objects is important. Even when they are used in a model that also contains customised extensions, the standardised status of these objects facilitates collaborating developers in coming to an agreement on the final model.

The benefit for interoperability of having groups of related Contracts refer to a common EIM is clear, and practised in some of the Trial 2 workgroups. However, the analysis shows that a mechanism for selectively identifying the subset of information objects from an EIM used in a Contract must be defined, otherwise over specification of the information content of Contract may result.

#### 3.2.1.4 Assessment of Trial Models

#### 3.2.1.4.1 Characterisation of Contracts and Systems

The following table characterises the Contracts designed in Trial 2 using the responses to the Workgroup and BB Development questionnaires.

Contract Name	Ref Points : Business Role supported		Number of separate operations supported	Interaction technology		Standards Conformance
SLARepo sitory	Internal: IES Provider	5	7	TCPIP	XML DTD	None

SLAHan dlingServ ice	IES-CM: IES Provider			XML RCP	XML Schema	OMG TSAS
SLANeg Reg	IES-CM: IES Provider	21	3	XML RCP	XML Schema	None
Ipsec- pContract	NA: VPNS Provider	5 (20)	7	EJB RMI passing XML data	XML Schema	IETF Ipsec Configuration Policy Model and Policy Information Base (partial)
Ipsec- pCOPSP RContrac t	VPNS-CM: VPNS Provider	20	3	EJB RMI passing XML data	XML Schema	None
VPN- Provision ing	Internal: VPNS Provider	13	11	EJB RMI passing XML data	XML Schema	ITU M.3208.1 and .3 (partial)
VPNServ iceConfig uration	VPNS-PM: VPNS Provider	12	11	EJB RMI passing XML data	XML Schema	ITU M.3208.1 and .3 (partial)
Assuranc e Service	GQIPS-PM: IES Provider	22	6	EJB RMI	Java Interface	
Assuranc e Configur ation	IES-CM: IES Provider	33	1	EJB RMI	Java Interface	WBEM (partial)
Performa nce Monitor	Internal: IES Provider	23	1	EJB RMI	Java Interface	WBEM (partial)
Server Monitor	AS-PP: IES Provider IES-CM: IES Provider	22	1	EJB RMI	Java Interface	WBEM (partial)
Resource Allocatio n Manager	GQIPS-PM: GQIPS Provider	3	20	EJB RMI	Java Interface	Internet2Qbon e (partial)
Customer Reporting Service	IES-CM: IES Provider AS-PP: IES Provider (GUI)	This is a GUI so IO difficult to assess	2	НТТР	Java Interface	WML (partial), SVG (full)
AQuEXC tr	Internal: IES Provider	1	2	RPC J2EE	Java Interface	IPDR (partial)
BillingInt	IES-CM: IES	1	1	RPC J2EE	Java	IPDR (partial)

eraction Managem ent	Provider					Interface	
RBSCtr	IES-AS: 1 Provider	IES	2	1	RPC COM/COR BA-SOAP	IDL, WSDL	IPDR, SOAP (full)
IPDREng ineCtr	IES-AS: I Provider	IES	1	6	RPC J2EE- SOAP	IDL, WSDL	IPDR, SOAP (full)
Interdom ainAcctM gmt	AS-PP: 1 Provider	IES	1	1	CORBA/II OP	IDL	IPDR (partial)

This table yields an average number of information object per contract of approximately 11 and the average number of operation supported per contract at approximately 5. This gives a rough guide to the complexity of Contracts that result from the approach taken in FORM and granularity of the corresponding BBs.

These figures should be used with caution, however, as further discussion revealed the interpretation of these figures depended on the developer's approach to information modelling and their interpretation of what an 'operation' was. Concerning information modelling, the use the complex types could potentially hide a large number of IOs that would result if the information guidelines had been followed more closely. Equally, as discussed in the EIM assessment, even fundamentally identical information models were open to representation with varying numbers of UML class objects. Concerning the definition of Contract operations, this was not clearly defined and was thus interpreted differently. For instance, a Contract interface signature may have a single method called something like 'process message' but then the message structure contains elements, e.g. message name, for which different values imply different operations at a business level. A clearer definition of a Contract operation is required that ties it to individual use cases and thus onto business level operations.

The Management system developed all had the majority of their functionally implemented through BBs, 80% on average. Non-BB subsystems to one related to user interaction or service delivery components.

# 3.2.1.4.2 Reference Point Assessment

Figure 3-14 depicts the use of Contracts in making up Reference Points in the IES Business Role Model. It can be seen that Contracts are provided to address all of the management related Reference Points (AS-CP represents application traffic and DS-CP and DS-PP represent IP diffserv level traffic). Further Contracts shown in square brackets are ones suggested to complete the functional scope of the Reference Point but which remain unaddressed in FORM.



#### Figure 3-14: Mapping of Contracts to Reference Points

Respondents indicated that the following Contracts could be applied more generally than the functional scope of the Reference Point to which they are mapped in the IES Management Framework.

- AssuranceConfiguration
- Customer Reporting Service
- InterdoaminAcctMgmt
- SLAHandlingService
- SLA Negotiation

Figure 3-15 is a UML Package diagram that depicts some suggested Contract Set Specification (CSS) and Building Block Groups (BBG) suggested by the workgroups. In the diagram a CSS is represented by a package with stereotype <<CCS>> and a BBG is represented by a package with stereotype <<BBG>>. Additional suggested BBs or Contracts are shown in grey.



Figure 3-15: Suggested Contract Sets and Building Block Groups

The suggested CCSs and BBGs provide a representation of how the Contracts and BBs produced might best be packaged for future reuse or release as a commercial product. The IES Fulfilment team has suggested an SLA management BBG containing the SLA related BBs developed, with a corresponding CSS. The Assurance group has defined two BBGs with two corresponding CSSs, one addressing end-to-end service QoS assurance and the other network provisioning with end-to-end QoS management. The Billing group have suggested a BBG with a wider scope than their trial system, i.e. one covering all the BBs needed to perform value-based, QoS dependent charging and billing. To this end they have included the SLA repository BB from the IES Fulfilment group to provide billing related SLA information and the Server Monitor BB from the Assurance group to provide usage data. In addition, they suggest an additional BB, the Subscription Manager, which handles multiple subscriptions to an outsourced billing service from multiple subscribers. The VPN Fulfilment group has suggested a BBG addressing the configuration and provisioning of a VPN service. This includes additional suggested BBs for management the distribution of encryption keys and the provisioning of MPLS links. The VPN Fulfilment group has, however, suggested that a separation of its Contracts into two CCS may be more appropriate, with one addressing IPSEC Management and the other addressing VPN Management. For the latter, two additional contracts are suggested, one for subscription to a VPN service and the other for VPN service reporting. For both of these, more general (i.e. non-VPN specific) Contracts may suffice.

# 3.2.2 Development Methodology

The development methodology questionnaire contained 25 questions which examined the FORM teams opinion and experience in using the guideline. Thirteen completed questionnaires were analysed and the results of this analysis are presented below.

# 3.2.2.1.1 Prior Experience in UML and RUP: Question 1 &2

This question attempted to ascertain the prior experience of the FORM development team's experience of using UML. Over 60% of participants indicated only occasional use of UML, with approximately 30% indicating the never used before, and 30% indicating frequent use. Only one respondent claimed expert usage of UML. This shows that the FORM team were not particularly skilled in UML modelling before the FORM project.

When asked about the prior usage of RUP, 100% respondents indicated that they no prior experience in this development process. Subsequent investigation indicted that several of the organisations used RUP but such use was either project specific or used by a very small set of business units within the organisation. It is important to note that the BP Guideline was not based on RUP, however, as RUP is the most widely used commercially used OO development process, and some of the Guideline workflows are loosely based on RUP, FORMs lack of prior RUP experience indicates the learning curve which was required by the FORM developers.

This project wide lack of experience in RUP and UML was one of the issues which the Guideline had to tackle, both in the presentation of the Guideline and in the examplars & workshops used to instruct partners in the use of the guideline.

#### 3.2.2.1.2 Role in BP System Development: Question 3 & 4

From the questionnaire results, two principal roles were taken by FORM partners were that of

- (i) responsibility for sub system or sub process development,
- (ii) overall contribution to design & development.

The first of these indicates a sizeable, responsibility for the full development cycle of a subsystem or sub process, the second indicates a more general contribution to design/development but not responsibility for that subsystem/sub process.

Question four broke down the principle development roles the partners played in the BP development as: Business Modelling/Requirement analyst designers, Implementator, Development Managers/Reviewers. Once again the design and development roles were very heavily represented with 70% and 61% of respondents indicating they took these roles within the development process. 46% acted in the role of requirements analyst and 30% as development managers/reviewers. As would be expected, FORM partners played multiple roles during the development process, with 53% of respondents playing 2 roles, and 23% playing only one role. Only 7 % (one respondent) claimed to play all four roles.

Subsequent investigation revealed that the development roles were principally that of system analyst, software developer and systems integrator. Few of the respondents in the development process were business analysts in their own organisation, however many had acted in senior system architect roles and had wide experience in the early design of commercial products and research prototypes.

#### 3.2.2.1.3 Usefulness of Business Modelling Workflows in BP Development: Question 5 & 6

One of the key aspects of the BP Methodology was importance and usefulness of the artifacts (e.g. models) which the development process prescribes e.g. Reference Model, Business Use Cases, Business Activity Diagrams. Question 5 asked about the usefulness of the Reference Architecture, where as question 6 asked about the usefulness of the Business Modelling Workflow.

Question 5 asked about the usefulness of the FORM Reference Model. Almost 40% of respondents found the specification of such a reference model useful, with 60% of respondents finding it marginally useful. However none found it highly relevant and none found it not relevant. A frequent comment regarding the reference architecture was that it was useful in setting the overall context of the various business processes under development, in indicating their 'ownership' (i.e. stakeholder within which the business process would execute) and in identifying potential cross stakeholder and inter business process communication. Respondents indicated that the reference model was most useful at this level of abstraction.

Question 6 asked the respondents to rate the usefulness of the Business Modelling Workflow. In particular it asked about the development activities defined in the guideline to specify the FORM Reference Architecture, Business Use Cases and Business Activity Diagrams.

The results were:

**Business Modelling Workflows** 

Reference Architecture: 23% highly relevant, 46 % useful, 23% marginally useful, 7% not relevant

Business Use Cases: 15% highly relevant, 23 % useful, 38% marginally useful, 23% not relevant

Business Activity Diagrams: 17% highly relevant, 17% % useful, 38% marginally useful, 30% not relevant

From the results there is a clear drift in the usefulness of the business modelling workflow with 69% of respondents indicating the reference architecture workflow either highly relevant or useful, compared to 34% indicating the business activity modelling workflow highly relevant or useful. Many respondents indicated that the reference architecture was very important in order to understand how specific functional management application areas may interact and co-operate with other functional areas.

Probably the most common difficulty cited by FORM developers was the level of abstractness of the business activities. This would seem understandable, as most of the respondents are not business analysts and therefore not particularly experienced in dealing with this level of abstractness. However, their opinion is important, as such developers typically have to read such business specification and map them down to systems level designs and specifications.

It should also be noted that many developers commented on the usefulness of explicitly modelling the control flow between business activities. It seems that the difficulty was not so much with the business modelling workflow and the resultant activity diagrams, but rather with the level of abstractness of the business activities identified in the different development groups. Where finer grained business activities were identified, the business activity diagrams were considered more useful e.g. assurance, fulfilment/assurance. More specifically developers felt that in such cases where the business activities were more fine grained, they were very helpful in subsequently identifying appropriate Building Block Contracts.

Very few of the interviewed respondents had difficulty following the guideline workflow instructions and there was general positive agreement as to the form of the models generated by the workflow.

# 3.2.2.1.4 Requirement Engineering: Question 7,8, & 9

These questions focused on the usefulness of the guideline's representation of requirements and the engineering of those requirements. From the answers it is clear that use cases and activity diagram specification were found very beneficial (i.e. approx. 85% of participants regarded the use case modelling activity as very useful or useful and 70% of participants regarded activity modelling as very useful or useful). Comments from participants indicated that these were easy to create, read and were useful in discussion of the requirements. The supplementary requirement specification activity was also found to be useful (66%).

When asked to indicate the relevance of the GB909 requirements when performing the modelling work, 18% rated them irrelevant, and 63% rated them only slightly relevant. Typically participants believed they were not that relevant to prototyping work. Also as each partner was focused on a set or building blocks or business process, large parts of GB909 were irrelevant to them. Some other participants indicated that a lot of the requirements just stated general guidelines which are so well accepted as to be superfluous and did not add any extra guidance. Those who rated the requirements as relevant, highlighted the fact such requirements reflected the non-functional requirements of the systems under development.

#### 3.2.2.1.5 Separation into 3 tier (HIT, PAT, EIT) in early analysis stages of design

One of the issues discussed during the design of both the building blocks as well as the BP driven systems, was the use of a three-tier architecture. When questioned as to the usefulness of this separation (during the analysis workflows of the guideline), the project participants indicated a wide divergence of opinion. 30% rated such separation as very useful in the analysis stage, where as 24% rated it as very poor (i.e. useless). This is surprising as such divergent opinion accounts for over 50% of the project participants. The reasons given generally involved the idea that a building block frequently required its own (internal) three layers – or at least two. This indicated the need for Building Blocks to provide persistency as well as business logic. Another opinion indicated that Building Blocks, to be reusable, needed to be less reliant on many database objects (persistency), but could manage the persistency themselves. Allied to this was the need to build business logic processing into these shared building blocks, and thus they could be arguably exist in either the business logic or persistency layers.

Others argued the opposite (more traditional) case, saying that for information objects regularly had to shared across building blocks, and that applying the three-tier architecture to such Building Blocks would assist such information sharing.

#### **3.2.2.1.6** Modelling activities for Control flows and Data Flows

Questions 11 and 12 focused on the modelling of systems processes – in particular the representation of control flows and data flows. 90% of participants rated the system activity modelling work as very good or good for representing control flow logic in the system. No participants rated the activity poorly.

Comments by the participants indicated that the activity was very useful as a prelude to integration of the building blocks (to support the BP driven system). Some participants preferred to use a collaboration diagram rather than an activity diagram, to capture the control flow logic. This is possible where all control logic is modelled activities (or collaborating object) and where no logic is external to such collaborating objects. Another comment stated that the activity diagram was good at capturing basic control flow but is not expressive enough to capture more complex flows e.g. multiple invocation of an activity. In such cases the complex control flow was modelled as an activity.

70% of participants rated the data flow modelling activity as very good or good, again with no participants rating it poorly. Most participants found the data flow modelling straightforward and helpful. One participant did not use activity diagram to represent data flow, but rather used a collaboration diagram, which indicated data objects flowing between the collaborating activity objects. One participant indicated that complex data flow could not be captured using activity models, and that an activity was used to represent the mapping of information from one activity to another.

# 3.2.2.1.7 External Information Modelling

Approximately 60% of participants rated the external information modelling as poor or very poor. Comments from the participants indicated that they believed in some cases the EIM was too abstract, in other cases it was two complex and unusable. Others indicated that the EIM would only be useful to internally help define the contracts of the Building Blocks. Others indicated that transformation activities would be required anyway, so that a consistent, universal EIM for the framework would probably be too difficult to achieve. Others felt that a federated approach, with no universal EIM, would be more appropriate. It is clear, talking to the participants, that a significant influence on the EIM used by participants, was based on the chosen application area, e.g. DMTF CIM for assurance, IEFT and ITU for fulfilment and IPDR for accounting. To attempt to integrate these information models was believed to be beyond the project and something, which may not be worthwhile. Agreement of some basic common information was useful e.g. SLA, etc. However, even for these information objects, transformations were sometimes required.

#### 3.2.2.1.8 Mapping System Activities to Building Block Contracts

59% of participants rated this activity as being performed with little or very little difficulty. 33% of participants rated this mapping as difficult. Participants found that they could easily handle any extra modelling required to accommodate the use of the building blocks. Some participants indicated that as the building blocks were developed within the same organisation as the business processes, the degree of difficulty is perhaps a little flattering. Another indicated that some re-design of the building block originally envisaged was required. This was needed to allow the business logic to be externally invoked rather then be performed completely within the building block.

#### 3.2.2.1.9 Mapping System Information Flows onto BB Contract Boundaries

50% of participants rated this activity as being performed with little or very little difficulty, with 50% indicating it was difficult. No participants rated it as being very difficult. The difficulties experienced tended to involve data transformation (between BB Contracts) and the modelling of extra activities needed to either retrieve or store missing information. However, the participants found they could readily recognise when such design activity was required and were able to deal with it in a systematic way.

#### **3.2.2.1.10** Collaboration Modelling of BB Contract Interactions

Approximately 64% of participants indicate little or very little difficulty in performing this workflow. No participants rated it as very difficult. Comments from participants indicated that this diagram was very useful in communicating the BB Collaborations which provide the basis for the Business Process Implementation.

62% of participants indicated that they were very satisfied that such collaboration diagrams provide enough information regarding the interactions between BB contracts (to support the system process to be implemented).

#### **3.2.2.1.11** Mapping BB Contract collaborations on to BB Collaborations

60% of respondents indicated that this modelling activity was performed with little or very little difficulty with 40% indicating some difficulty. No respondents indicated this task was very difficult.

#### 3.2.2.1.12 Opinion of Methodology/Guideline Description

Most participants (54%) indicated that only between 1 and 5 hours were needed to study the guideline). However, some participant indicated that 30% spent more that 5 hours studying the guideline (this was in addition to several guideline presentations and tutorials given during the project). All indicated that the combination of the workflow descriptions as well as the worked example(s) were very useful in gaining a better understanding of the guidelines prescribed activities.

Participants were asked to rate the clarity of the guideline in describing the various workflows in the development lifecycle. Overall approximately 60% of participants rated the guidelines as very clear or clear, while none indicated that it was very unclear. The only workflow which attracted any difficulty in understanding was the BB Contract Collaboration modelling activity. During the guideline development, this workflow was updated to give greater clarity to readers.

#### 3.2.2.1.13 Summary of the Development Methodology Evaluation

It is clear from the evaluation that the guideline has been successful in assisting all of the FORM development teams in constructing their management systems. The principle advantages of the guideline seems to be its well formed set of development activities which provide tractability of artifacts and designs between the different workflows, and provides well explained development activities. The areas where the guideline could be strengthened is in its treatment of the non-functional requirements which are not explicitly reflected in the system process descriptions. Also some of the limitations on control flow for system processes captured as activity diagrams could be investigated further. However, UML v2.0 is intending to enhanced the control representational ability of activity graphs in its new release. Some issues also remain about the content of the EIM, in particular, regarding the integration of heterogeneous information models from different standardisation bodies.

#### 3.2.3 Technology Selection Guidelines

The J2EE platform represents a single standard for implementing and deploying enterprise applications. The J2EE platform has been designed through an open process, engaging a range of enterprise computing vendors, to ensure that it meets the widest possible range of enterprise application requirements. It is designed to provide server-side and client-side support for developing enterprise, multi-tier applications. Such applications are typically configured with a **client tier** to provide the user interface, one or more **middle-tier** modules that provide **client services and business logic** for an application, and backend **enterprise information** systems providing **data management**.

Within the FORM consortium, partners used as a technology framework to implement the various Building Blocks that are supporting business processes in the different areas tacked by the FORM project (Assurance, Fulfilment, and Billing).

The intent of this section is to present the evaluation of the J2EE framework against some of the FORM ODF concepts. This framework takes on board the idea of software Building Blocks. However as stated in section 2.2.4 (Technology Selection Guidelines), this evaluation does not aim at providing a full mapping between BBs as defined in GB909 and the various J2EE features.

It is not the intent of this section to present in details the various features of the J2EE framework, however it is worth pointing out some interesting literatures that present this technology framework in detail. The following white papers are accessible via the Sun java website, and particularly via the "OSS through JavaTM Initiative" web page.

# Simplified Guide to the Java 2 Platform, Enterprise Edition:

http://java.sun.com/j2ee/white/j2ee\_guide.pdf

- Java 2 Platform, Enterprise Edition: Ensuring Consistency, Portability, and Interoperability: http://java.sun.com/j2ee/white/j2eeAnneThomas.pdf
- **Telecom Network Management With Enterprise JavaBeansTM Technology:** This paper explains how builders of the next generation networks can benefit from moving to a standard application server model for their management frameworks. <u>http://java.sun.com/j2ee/white/nmEJBwp.pdf</u>

# **3.2.3.1.1** J2EE Evaluation

# 3.2.3.1.1.1 Procedure

The evaluation of the J2EE technology framework was introduced in the rather early stages of the FORM projects, mainly because it was used as a prime implementation technology for many of the partners. Intermediary results were produced after the first Trial of the FORM systems. These results reflected the evaluation of J2EE against TMForum "Generic Requirements For Telecommunications Management Building Blocks" [gb909], and are summarised later in this document. The following text, also present the result of the evaluation conducted after the second trial of the FORM system. In order to gather the experiences scattered between different workgroups and different partners, a simple questionnaire was produced (see annex B). The idea of this second evaluation was to produce evaluation results against the FORM ODF architecture principles.

# 3.2.3.1.1.2 J2EE Evaluation Main Results

The J2EE evaluation results are divided in two sections, the first one depicts the evaluation against FORM ODF architecture principles. The later summarise the evaluation against GB909 requirements.

#### 3.2.3.1.1.2.1 Evaluating against FORM ODF Architecture Principles

In this section of the questionnaire some principles from the architecture principles expressed in D9 [formD9] where identified as having an impact on the implementation. Answers were provided in a J2EE perspective.

# P2: Building Blocks are pieces of software that are atomic units of deployment (one can be replaced in a running system without requiring other BBs to be replaced or modified).

The answers provided by BB implementers have shown that in general where EJB were used BBs were packaged as Enterprise Application Archives (EAR), while graphical interfaces were packaged as Web Archives (WAR). Therefore although FORM has expressed concern regarding the strict restriction of BB to the three tiers as expressed in GB909, a mapping can be found between the packaging type and the nature of the BBs. The fine granularity of the packaging methods in J2EE allows for a good flexibility in replacing not only a BB independently but also sub-components within a BB (EJB Jar files can be easily replaced within an ear package). Note that where dependencies exist between different Building Blocks, the J2EE ejb-jar deployment file lists all of the dependencies that the Building Block (EJBs) rely on.

The results gathered in the questionnaire show however an important result. Partners, for most, have used application servers that were compatible with the EJB v1 compatibility test. Overall the EJBv1 specification was quite loose on a standard deployment unit. Therefore various application servers are still defining their deployment unit and their descriptor in rather proprietary ways. For instance, when using JBoss-2.2.2 deployment descriptors would have been expressed as "jboss.xml" or "web-jboss.xml" for web archives, those files would not be recognised by other application server like WebLogic, therefore deploying anywhere was still an issue. Overall these compatibility issues are now addressed as part of the version 2 of the J2EE compatibility tests.

P8: A BB implements a Contract in a technology specific form. When mapped from a Contract specification in a technology neutral form, this must be performed using an explicitly described transform.

Within FORM, contracts were not expressed fully in technology neutral form, therefore transform were not necessary. In terms of contract EJB deployment descriptors were seen by many as the technology specific interface of Building Block.

P9: Different Contracts may support different interface definition paradigms, though one Contract specification can only support one such paradigm. Interface definition paradigms include, but are not limited to, model-centric, operation-centric and message-centric. Different paradigms are typically suited to specific ranges of technologies.

Where implemented using the J2EE technology framework, Building Blocks in FORM were mostly based on the RPC based paradigm. This is made possible in J2EE mainly via the use of RMI. Integration with different technologies, such as integrating "EJB BB" and "CORBA BB" was also proved possible in FORM using RMI over IIOP. Late work within the project on Web Services, especially in the Billing group, produced some very interesting results regarding the ease of integration of such services between different platforms (COM, EJB, CORBA, etc...). It is believed that these emerging web service concepts will have an impact on how BB integrates to each other in the future.

P11: Functionally related Building Blocks can be grouped together for the purpose of software release.

Within FORM scope, the purpose was not to achieve software results. This principle, therefore, was not seen as having a big importance on the implementation done. However, as mentioned before, the standard packaging procedures within J2EE allows for such grouping, and therefore combining various J2EE archives (jar, war, ear) files as one software release is possible.

P13: The behaviour of a Contract and interactions with the behaviour of other Contracts on the same BB may be modified at deployment- or run-time. Where this feature is offered it should use explicitly defined business-rules.

This principle would relate largely to the use of policies as a concern while implementing BBs. It was found that policy was not in the scope of all the work groups within FORM.

Different experiments were done however, for example the workflow framework technology, in the assurance group, supports the definition of Processes (business rules), however, it is considered part of the infrastructure and not a Building Block. For the development work done on VPN, the business rules as well as possibility to adapt BB behaviour to specific user context is supported in the VPN-P BB by the use of policy.

#### 3.2.3.1.2 Generic Requirements for Telecommunications Building Blocks

#### 3.2.3.1.2.1 TMF GB909 BB: Core Feature

**Operability**: "A building block must be robust in its ability to maintain the integrity of its function and data at critical times such as during upgrades and network outages. It must be able to survive a wide variety of failure scenarios. In the face of a failure, a building block must employ strategies to resist the failure to the extent that is possible and appropriate". [Integrity, Survivability, Recoverability] According to Sun's definition, a Session bean is an enterprise bean that is created by a client and that usually exists only for the duration of a single client-server session. A session bean performs operations, such as calculations, for the client. While a session bean may be transactional, it is not recoverable should a system crash occur. Session bean objects can be either stateless or they can maintain conversational state across methods and transactions. If a session bean maintains state, then the **EJB container** manages this state if the object must be removed from memory. However, the session bean object itself must manage its own persistent data. Hence in essence, a Session bean does not conform to the operability requirements. However, it was found that a combination of Session Bean for performing operations and entity beans for recording the state of such operation could enable this concept of recoverability. It is also interesting here to point out the current efforts in the Java community to combine technology sets such as J2EE and Jini-based architectures, which are known for their self-healing capabilities [jini\_ent].

Interoperability: "Building blocks must be able to interact in a loosely coupled manner, using mechanisms that support extremely late binding of client and server. Interaction must occur via well-formed, well-documented interfaces (contracts) which are published through a yellow-pages-type trading service, which enables clients to "look up", server contracts by type without knowing the name or location of any specific instance. Subject to security constraints, all clients must have uniformity of access to contracts in the distributed computing network".

In the design of beans a remote interface must be specified through which the functionality of the bean can be accessed. This allows the contract of the bean to be effectively separated from any particular implementation of that bean. In FORM, such an interface is specified as part of the *<interfaceInteractions>* element in the contract specification of a BB [FORMContracts]. Also, most beans make use of a naming service to locate each other and to allow a certain level of late binding. J2EE extends this concept by allowing a mapping to be specified, in an XML file, between the name the code looks up the actual name used in the naming service. Therefore the name of the object to which we wish to bind is not needed until deployment. Experiments in FORM on the Java Messaging Service (JMS) technology, now part of the EJB framework, have showed that JMS allows asynchronous messages to be sent between different building blocks thereby allowing them to be less tightly coupled.

**Commoditisation**: "Common building blocks should become available on an off-the-shelf, shrinkwrapped basis. Industry forums, such as the TeleManagement Forum, should define and test contracts (well-formed interfaces) for common building blocks"

Based on FORM's scope, this "Commoditisation" concept was not evaluated to its full extent. However, FORM consortium has agreed on an XML based Contract specification [FORMContracts].

**Common Management**: "Building blocks must be manageable as applications in a common manner. A company must be able to manage building blocks from multiple suppliers with a single management system."

For this common management, the question still remains "Can BB implemented within the J2EE specification be "plug-and-play" within a complex distributed system?" Experiments have shown that if deployment tools provided by various Application Servers have proved very useful they only provide the basics for management and may not be sufficient for complex distributed systems.

#### 3.2.3.1.2.2 TMF GB909 BB: Requirement and Objective

As mentioned earlier in this document, FORM trial systems were more oriented toward the fulfilment of functional requirements. GB909 BB requirements are for a large part non-functional requirements, related to the quality and reliability of BBs and BBs based systems. Within FORM, earlier in the lifetime of the project, an empirical study was conveyed, presenting possible mappings between GB909 BB requirements and EJB technology features; the overall impact of GB909 requirements was lesser by the importance given to achieving functional goals.

#### 3.2.3.1.3 J2EE evaluation Summary

Within FORM, the J2EE technology, once learned by partners, has enabled rapid prototype development of the initial set of building blocks. It has allowed experimentation with the design and architecture of the system with little effort in achieving the core middleware services. Some BBs have been ported from one J2EE server implementation to another, showing that the concept of commoditisation is achievable. Although it is believed that the current effort in ensuring Application server better compatibility with J2EE specifications, will greatly improve the applicability of this concept.

Although some enhancements to this technology will be necessary for a widespread adoption in the industrial environment, J2EE has emerged as a major player in the technology for Business-to-Business environments. Main enhancements to such platform should be to provide better management possibilities as well as services such as yellow pages and notification. Interoperability between platforms is also an important issue.

To conclude, within the scope of the project, J2EE was found as a suitable technology for achieving Building Blocks functional requirements. If non-functional requirements were not all addressed when implementing BBs, some initial results proved that J2EE is also suitable for fulfilling most of GB909 Building Block requirements and objectives.

# 3.2.3.2 XML for Contracts

In this section we look at the possibilities of using the eXtended Mark-up Language [XML] to realise some of the benefits listed in the architectural principles P7, P8 and P10 in [formD9].

#### P7: Contracts may be defined in a technology neutral form or a technology specific form.

One of the benefit we aimed for was that standards bodies can mediate industry agreements on Contracts in a technology neutral form without necessarily excluding specific target technologies, thus potentially making the specification more long-live in the face of rapid technology churn. A major requirement for such a technology neutral specification is that it must be extensible so new data types required to support new management functions in Building Blocks can be added to existing Contract specifications. XML is a meta-language and as such does not restrict any form of extensions, as a consequence of this list of different specification languages that are now based on XML is long and continues to grow. System Integrators should benefit from comparing Contract specification presented in the technology neutral form without being forced into an early, potentially excluding, technology decision and with the knowledge that any implementations obtained may offer stable functionality over technology changes. XML support this by enabling Contracts to be human readable and there by less complicated to compare than specifications which are not human readable.

*P8:* A BB implements a Contract in a technology specific form. When mapped from a Contract specification in a technology neutral form, this must be performed using an explicitly described transform.

One of the benefits of this architectural principle is that ISVs may use the explicit transform between the technology neutral and technology specific forms to allow different technologies (or even different profiles of the same technology) to be used to implement the same Contract to match market requirements. Similarly, System Integrators may use such explicit transforms to implement the same Contract to match the different integration technology requirements of different systems. Extensible Stylesheet Language XSL [XSL] and XSL Transformations XSLT [XSLT] are recommendations from WWW Consortium. XSL is a language for transforming XML documents into any other text based representation incl. new XML documents, where the transformation details are specified in XSL Stylesheets. XSLT enables integration of information from multiple XML documents in the text-based output. XSL Transformations could be used by ISVs and system integrators to translate XML documents from a common technology neutral specification language, to more detailed technology specific specification languages which does not have to be XML based. The XSL Translation does not support use of dynamic variables e.g. counters, so such additional information must be written in temporary XML documents referenced in the XSL document or added to the XML document which is being translated.

# *P10:* The definition of the information that is passed via a Contract should be published separately to the Contract specification.

By utilising XML, standard organisations are specifying and standardising collections of data type definitions also known as XML vocabularies. As a benefit, standards bodies are able to encourage commonality in Contract specifications by both using and publishing industry agreements on information that may be passed via a Contract. By using XML vocabularies ISVs and System Integrators are both able to encourage better interworking between the BBs it produces by separately publishing and reusing the information passed over Contracts it designs.

XML also provide a method to make vocabularies globally unique by utilising the Namespace facility of the XML Schema recommendation [XSD]. The use of namespaces enable XML documents to reference XML vocabularies through Uniform Resource Identifiers URIs known from HTTP. Different Namespaces may be used to differentiate between basic Vocabularies and Vocabularies with extended data types or data types specialised for a specific target technology.

The tML Framework provides a standard definition for the development of interoperable interfaces based on the use of the eXtensible Markup Language (XML), within the TMN context. The goal for use of the Framework is to guide the development of tML schemas and vocabularies and to provide a common method for the definition of tML data to be exchanged and to provide a mapping to existing standards to promote re-use whenever possible. The tML Framework consists of: a set of rules and objectives to be applied in developing standardised schemas based on existing models in standards; specification of common tML tags, namespaces, and URIs; transformation of existing definitions in external libraries for use with tML; and requirements for repositories and registries.

# 3.2.3.3 XML for mediation

In this section we look at how XML have been used in the trials to support mediation of data exchanged between system components or to support mediation between system technologies. Areas where benefits have been found will be identified in the following.

An important area identified was validation of arguments and return values which where exchanged as XML. Normally mismatches in the structure of the information sent between client and server are not revealed until run-time. During development and testing the XML Schema enabled developers to ensure the validity of the arguments and return values. This benefit may have been enhanced by the fact that developers used the same XML parser software i.e. Apache [Xerces].

One major difference between XML and other specification languages is that XML enables human readable specifications. As the Bandwidth Broker XML Schema was easily readable – it could be used at design phase to show designers of other building blocks what kind of information the GQIPS required. It was found to be a straightforward means of information mediation and system implementers found it to be an easy mean of receiving information to be processed in the GQIPS.

XML and the tools available made it easier for partners to co-operate on defining the types used in a common information model. Another notation than XML could have been used, but the use of XML has the added benefit that it is not necessary to translate the model into something (e.g. text or objects) that can be used during development. XML is directly usable in the development. Using XML enables developers to de-couple the internal information model from the information model exposed in the interface thus ensuring that the component is truly a black-box component. Use of XML for definition of interfaces; bring a degree of neutrality as it allows an implementation based on different interface paradigm. Another important area is the ability to upgrade and versioning XML based information to adapt to upgrades in system components and applications.

Where workflow framework technology was used to support execution of management processes, the XML Metadata Interchange [XMI] provided a standard format for parsing the UML Activity Diagram Process Models into Process Definitions. An additional benefit of this is that developers may use any UML case tool that can export XMI for process modelling.

Where DMTF's Common Information Model (CIM) was used by Building Blocks the main advantages of using XML were the pre-existing mapping of the desired information to XML schemas and the ability to use the large number of XML tools to access and manipulate the information. XML based WBEM messages were used for Remote Method Invocation between the CIM system and the Building Blocks requiring access to the CIM. A benefit of this was that a common interface could be defined for each software component e.g. Java Bean which required access to the CIM.

### **3.2.3.3.1** Assessment of use of Policy-based techniques

Policies were used in Trial 2 with related to five BBs, namely:

- The SLA Negotiation Engine BB in the IES Fulfilment trial system
- The VPN Provisioning BB in the VPN Fulfilment trial system
- The IPSEC-P BB in the VPN Fulfilment trial system
- The Server Monitor and Performance Monitor in the Assurance trial system

No single approach was taken to applying policies, and different policy languages were adopted. The SLA Negotiation Engine BB used a custom XML-based policy language, POLML, which is a superset of the IETF/DMTF languages, designed to allow for better design time grouping of policy rules. The VPN Provisioning BB used the DROOLS policy language (see <a href="http://drools.org/">http://drools.org/</a>). The IPSEC-P BB used the policy elements from the IETF Drafts "IPSec Configuration Model" and "IPSec Policy Information Base", for the Policy Objects to be exchanged between IPSec-P and VPN-P (i.e. the BB on the layer above). The Server Monitor and Performance Monitor BBs made use of the DMTF Policy Model. Its policies were generated at run-time based on terms obtained from an SLA, so they made little use of existing rules, policies or conditions, expect for the DMTF representation for a time period condition.

On the whole, policies were used only to control the behaviour of the BB, or for policies passed to another BB. With the SLA Negotiation Engine, however, the POLML language allowed references to be made to other entities, such as other BBs, which could be called to assess a condition or perform an action. This potentially allowed a policy to control the interactions with other BBs.

Overall the use of policy was found to be advantageous in providing flexibility in the functionality of the BB, but using policy languages to obtain this flexibility proved to be complex.

# 4 Conclusions

This document presented the Inter-Enterprise Management Framework assessment structure and then the results of the evaluation. The evaluation was carried out under two main headings, i.e., the operational requirements and the development aspects. The four test team trials of the system models outlined in [formD11] are presented and compared against the operational requirements as perceived by the end customer and Inter-Enterprise System Provider. The development aspects are outlined under the headings of Logical architecture, Development Methodology and Technology Selection Guidelines.

Annex A contains the test case plans and results. Annex B contains the questionnaires used for the development aspects evaluation.

# **5** References

[FORMContracts] Index to the Contract catalogue:

http://www.cs.ucl.ac.uk/research/form/models/ContractCatalogue/

- [formD9] D9 Final FORM Framework, Eric Leray, IST-1999-10357/WIT/WP3/1019.
- [formD11] D11 Final Inter-Enterprise Management System Model, Birgitte Lønvig, IST-1999-10357/LMD/WP4/0522.
- [formD12] Wade, Vincent, "D12: Guidelines for Co-operative Inter-Enterprise Management", IST-1999-1057/TCD/WP3/012, February 2002
- [gb909] Generic Requirements for Telecommunications Management Building Blocks, GB909, Member Evaluation Version 2.0, TeleManagement Forum, September 1999
- [Internet2 QBone] Various standards for QoS according to QBone are available at <u>http://qbone.internet2.edu/</u>
- [IPSec Configuration] IETF IP Security Policy WG: "IPsec Configuration Policy Model", Draft v/2, draft-ietf-ipsp-config-policy-model-02.txt
- [IPSec Policy] IETF IP Security Policy WG: "IPSec Policy Information Base", Draft v/2, draft-ietfipsp-ipsecpib-02.txt
- [jini\_ent] "Jini.org Enterprise Project", Jini for the Enterprise and for Web Services, http://developer.jini.org/exchange/projects/enterpriseweb/index.html
- [lewis] A Logical Architecture for an Open Development Framework for Component-based Management Systems, Dave Lewis, IST-1999-10357/UCL/WP6/0921-v1
- [M.3108.1] ITU-T Recommendation M.3108.1: "Information Model for Management of Leased Circuit and Reconfigurable Services"
- [M.3108.3] ITU-T Draft M.3108.3: "Information Model for Management of Virtual Private Network Service"
- [M.3208.1] ITU-T Recommendation M.3208.1: "TMN Management Services for Dedicated and Reconfigurable Circuits Network: Leased Circuit Services"
- [M.3208.3] ITU-T Draft M.3208.3: "TMN Management Services for Dedicated and Reconfigurable Circuits Network: Virtual Private Network Service"

[Xerces] XML Parser software

http://xml.apache.org

[XMI] XML Metadata Interchange, Object Management Group OMG

http://www.omg.org/technology/documents/formal/xmi.htm

[XML] XML 1.0 Second Edition

http://www.w3.org/TR/2000/REC-xml-20001006

[XSLT]XSL Transformations (XSLT)

http://www.w3.org/TR/1999/REC-xslt-19991116

- [XSL] Extensible Stylesheet Language (XSL) Version 1.0 http://www.w3.org/TR/2000/CR-xsl-20001121/
- [XSD] XML Schema Part 0: Primer

http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/

# 6 Acronyms

Acronym	Definition
ACS	Accounting Data Store
AS	Application Service
ASP	Application Service Providers
AS-CP	Application Service – Customer to Provider
AS-PP	Application Service – Provider to Provider
B2B	Business to Business
BB	Building Block
BBG	Building Block Group
BIM	Boundary Information Model
CIM	Common Information Model
СІМОМ	Common Information Model Object Manager
СОМ	Component Object Model
CORBA	Common Object Request Broker Architecture
CSS	Contract Set Specification
DLL	Dynamic Link Library
DMTF	Distributed Management Taskforce
DOM	Document Object Model
DS-CP	DiffServ – Customer to Provider
DS-PP	DiffServ – Provider to Provider
DTD	Document Type Definition
EAR	Enterprise Application Archives
EC	End Customer
E-IPDR	Extended Internet Protocol Detail Record
EIM	External Information Model
EJB	Enterprise Java Bean
FMA	Federated Mediation Adapter
GDMO	Guidelines for Definition of Managed Objects
GQIP	Guaranteed QoS IP
GQIPS	Guaranteed QoS IP Service
GQIPS-PM	Guaranteed QoS IP Service – Provider Management
GQIPS-PP	Guaranteed QoS IP Service – Provider to Provider
GUI	Graphical User Interface

HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IES	Inter-Enterprise Services
IESC	Inter-Enterprise Service Customer
IESP	Inter-Enterprise Service Provider
IES-CM	Inter-Enterprise Service - Customer Management
IIOP	Internet Inter-ORB Protocol
IP	Internet Protocol
IPDR	Internet Protocol Detail Record
IPSec	IP Security
ΙΟ	Information Object
ISP	Internet Service Provider
ISV	Independent Software Vendor
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
J2EE	Java 2 Platform, Enterprise Edition
JAR	Java Archive
JDK	Java Development Kit
JMS	Java Message Service
NGOSS	New Generation Operating System Support
ODF	Open Development Framework
ODP	Open Distributed Processing
OMG	Object Management Group
00	Object Oriented
QoS	Quality of Service
RBS	Rating Bureau Service
RMI	Remote Method Invocation
ROI	Return on Investment
RPC	Remote Procedure Call
RUP	Rational Unified Process
SLA	Service Level Agreement
SVG	Scaleable Vector Graphics
ТСР	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TM Forum	Tele Management Forum
TMN	Telecommunications Management Network

ТОМ	Telecoms Operation Map
ТТ	Test Team
UML	Unified Modelling Language
UML	Unified Modeling Language
URI	Uniform Resource Identifier
VPN	Virtual Private Network
VPNS-CM	Virtual Private Network Service - Customer Management
VPNS-PM	Virtual Private Network Service - Provider Management
WAR	Web Archives
WAP	Wireless Application Protocol
W3C	World Wide Web Consortium
XHTML	eXtensible HyperText Markup Language
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformation