**TAPAS**

***IST-2001-34069***

***Trusted and QoS-Aware Provision of Application Services***

# TAPAS
# Periodic Management Report
# PM1 – PPR1

**Report Version:** Deliverable DI

**Report Delivery Date:** May 2003

**Classification:** Public Circulation

**Contract Start Date:** 1 April 2002 **Duration:** 36m

**Project Co-ordinator:** Newcastle University

**Partners:** Adesso, Dortmund – Germany; University College London – UK; University of Bologna – Italy; University of Cambridge – UK


**Project funded by the European Community under the "Information Society Technology" Programme (1998-2002)**

## 1. Executive Summary

In the TAPAS project, we are particularly interested in developing solutions to the problem faced Application Service Providers (ASPs) when called upon to host distributed applications that make use of a wide variety of Internet services provided by different organisations. This naturally leads to the ASP acting as an intermediary for interactions for information sharing that cross organisational boundaries. However, despite the requirement to share information and services, autonomy and privacy requirements of organisations must not be compromised. Organisation will therefore require their interactions with other organisations to be strictly controlled and policed. This creates two major challenges. Firstly, contractual relationships between multiple organisations for information access and sharing will need to be governed by *service level agreements* (SLAs), which will need to be defined and agreed between the organisations and then enforced and monitored by the ASP. Secondly, the ASP will have to establish appropriate *trust relationships* with the organisations and implement corresponding security policies before organisations will permit the ASP to act as an intermediary for inter-organisational service invocations.

Unfortunately, ASPs currently lack tools and techniques for offering hosting facilities for such distributed applications: the reason why TAPAS project has been created. First year work has been concerned with developing an overall framework for the development of software tools and techniques for hosting advanced distributed applications. We need to answer a few basic questions: What are the requirements of distributed application hosting? How to specify SLAs? What does "interactions with other organisations to be strictly controlled and policed" actually mean? What is trust and how to 'manage' it? How to incorporate QoS in component middleware? These questions have been answered in the first year deliverable reports: D1 (requirements), D2 (SLA specification), D5 (TAPAS architecture, describing trust management and regulation of inter-organisation interactions), and D7 (TAPAS middleware platform: QoS enabled application servers).

## 2. Project Rationale and Exploitation Plan

### 2.1. Rationale

Organisations, particularly small and medium scale enterprises (SMEs), are increasingly finding it difficult to develop, maintain and manage their IT applications largely due to difficulties in retaining and attracting trained IT staff. *Application Service Providers* (ASPs) hold the promise of providing an attractive solution by making available application hosting facilities on remotely managed servers. However, to work effectively, ASPs must guarantee security, provide resilience and *service level agreements* (SLAs) over commonly available infrastructures. Furthermore, ASPs need to ensure that hosted applications are capable of accessing a wide variety of services irrespective of the platform or the organisation through which they are provided. An ASP typically uses middleware and component technologies for deploying, hosting and managing applications of an organization from a centrally managed facility. However, as organisations become global and distributed, such centrally managed hosting solutions will need to be replaced by multi-site, distributed hosting solutions.

The goal of TAPAS is to develop multi-site, distributed hosting solutions. We argue that many research problems in enterprise distributed computing will need to be solved to achieve the goal of TAPAS. An ASP will increasingly be called upon to host distributed applications that make use of a wide variety of Internet services provided by different organisations. This naturally leads to the ASP acting as an intermediary for interactions for information sharing that cross organisational boundaries. However, despite the requirement to share information and services, autonomy and privacy requirements of organisations must not be compromised. Organisation will therefore require their interactions with other organisations to be strictly controlled and policed. This creates two major challenges. Firstly, contractual relationships between multiple organisations for information access and sharing will need to be governed by SLAs, which will need to be defined and agreed between the organisations and then enforced and monitored by the ASP. Secondly, the ASP will have to establish appropriate *trust relationships* with the organisations and implement corresponding security policies before organisations will permit the ASP to act as an intermediary for inter-organisational service invocations. Unfortunately, middleware services for inter-organisational interactions as outlined above do not yet exist; indeed, development of such services is very much a research problem. Thus ASPs currently lack tools and techniques for offering hosting facilities for advanced Internet based applications.

### 2.2. Exploitation Plan

During the first year of the project, no changes have been made. The plan given here is essentially taken from the description of work. We have many exploitation routes for TAPAS results through our on going research projects and interactions with the advisory panel. We first describe how the academic partners will exploit TAPAS results and then describe the benefits gained by the industrial partner.

**Newcastle University**: The Distributed Systems Group has a strong record of working with industries. In conjunction with Nortel (Harlow research lab), we contributed to the development

of the workflow standard by making a submission to the OMG based on our workflow technology (Nortel and University of Newcastle upon Tyne, “Workflow Management Facility Specification”, Revised submission, OMG document bom/98-03-01). We worked with IBM (Winchester lab) and IONA Technologies towards the development of new transaction standard (contribution to OMG RFP, “Additional structuring mechanisms for the OTS”). Our technologies have been in use in several industrial settings. Results from TAPAS will be used in existing and future research projects on middleware related distributed computing.

The group set up a company in 1998 in Newcastle to productise Arjuna transaction and workflow technologies. The company was bought by Bluestone Software, Inc., a leader in business-to-Web and wireless technologies; in 2001, Hewlett-Packard bought Bluestone and the original Arjuna company operated as HP-Arjuna Labs, a part of HP's middleware division. After merger between HP and Compaq that led to substantial reorganisation of HP, Arjuna became independent again. Now based within the University campus, Arjuna Technologies is a centre of excellence in transaction technologies and is focusing on building products to support reliable Web Services-based applications. Close industry-university collaboration is guaranteed, and research projects on E-commerce platform and services have been initiated. Arjuna Technologies will be represented on the Industry Advisory Board.

**Bologna University:** The research group at the Department of Computer Science of the University of Bologna maintains close cooperation with national and international industries, including Microsoft (Cambridge Research Laboratory) and Sun Microsystems, and national research institutes, the "Fondazione Marconi" and the ENEA (the Italian bureau for new technologies applied to the energy and environment), in the form of joint investigations under contracts and grant programmes.

In addition, this research group is going to be involved in the national project entitled "Infrastructure Support for e-business applications", that will be carried out in collaboration with a number of Italian Universities and companies. This project, which is currently under formal approval by the Italian Ministry of the University and Scientific and Technological Research, will greatly benefit from the results of the TAPAS project. In addition, two further projects, in which the research group in Bologna will be involved, can benefit from the TAPAS project results. Both these projects, entitled "A Distributed Broker for Quality of Service", and "Middleware for advanced services over large-scale, wired-wireless distributed systems", respectively, will deal with issues of QoS at the middleware level. These two projects have been recently submitted for approval to the Italian Ministry of the University and Scientific and Technological Research; both these projects will be carried out in collaboration with other Italian Universities and companies.

The cooperation and collaboration channels mentioned above will be to transfer the results that will emerge from our research activity in the TAPAS project.

**University College London**: UCL relies to a considerable extent on direct industrial funding and consulting. The knowledge required to be able to provide high-quality consulting services is often produced in projects such as TAPAS. UCL therefore hopes to exploit the results of the TAPAS in the following ways:

- Technology transfer initiatives towards the industry.

- Provision of consulting services to external companies (including education and training).
- Dissemination (consisting essentially of publications and courses).

The Software Systems Engineering Group of UCL is well positioned for this exploitation. It has ongoing research collaborations with a number of industrial partners, including British Telecom, Hewlett Packard Labs, IBM Hursley Park, Kodak, Microsoft Research, UK National Air Traffic Services, Philips, Searchspace, Telelogic, Toshiba Corporation, UBS Warburg, Unipower Solutions and the Zuehlke Technology Group. The TAPAS project partners hope to use these good collaborations as a route for exploiting the knowledge produced in TAPAS.

**Cambridge University:** Cambridge University has a plethora of industrial collaborators, and will seek to exploit any and all of its research when appropriate. In this project, links with Microsoft Research and Marconi Research may prove extremely valuable, as both have research laboratories in Cambridge working in collaboration with the University. As well as this, the Compaq systems research lab, and Hewlett Packard may also be potential paths to exploitation given long term relationships with both. Finally, we will be able to use the output of TAPAS within the academic community itself directly to support more performant application services for teaching and research.

**Adesso AG:** Adesso AG is a full service provider for the design, development and operation of e-business applications. The development paradigm applied is that of component-based software development. This paradigm and the application domain of e-business applications perfectly match because most e-business application encompass various COTS components. This does not only pose some extra challenges with respect to system integration, release management and test of e-business applications, but it is also hindering the business model ASP for e-business applications. Due to the heterogeneity of e-business application standard ASP service level agreements usually cannot be applied. Instead it is necessary to relate service level agreements to components of an e-business application individually. This may, for example, mean to define service levels agreements as the following:

- The portal site will be accessible for 98% of the time.
- Access to the e-controlling component is ensured for 90% of the time.
- The minimal recovery time for the access to individual customer data is 20 minutes, the recovery time for profile data is 60 minutes.

This example shows, that different types of functionality ask for detailed agreements. Thus, fine-grained service level agreements help to provide the services needed at affordable costs. Of course, it is possible to offer only more coarse-grained service levels, but this usually leads to cost explosions which are not acceptable for customers. With the possibility to define and implement fine-grained service level agreements developed by TAPAS, Adesso can foster its core business in several ways:

1. It is possible to argue for component-based development of e-business applications, because this is a prerequisite for fine grained service level agreements.

2. The range of software systems which can be integrated into e-business applications which are to ASP-operated is extended. For the time being, systems whose low robustness endanger the

availability of the overall e-business application cannot be integrated. If it was possible to agree for lower services or such a component, it would be possible to integrate despite its robustness.

3. The ASP services of Adesso will be much more attractive, if fine-grained agreements are possible. In contrast to standard offerings, the ASP levels can be precisely adapted to customer requirements.

While the first and second way to foster the Adesso business cannot be calculated in concrete numbers, the third way is supposed to allow an extra 20% growth in ASP business (after being able to define and implement fine-grained service level agreements).

# 3. Project Objectives

## 3.1. Overall Objectives

The overall objective of the TAPAS project is to develop novel methods, tools, algorithms and protocols that support the construction and provisioning of Internet application services. The project will achieve the overall objective by developing QoS enabled middleware services capable of meeting Service Level Agreements (SLAs) between application services and will enhance component based middleware technologies such that components can be deployed and interact across organisational boundaries. The project will develop notations for expressing SLAs to enable specification of QoS, such as the availability as well as trust relationships. SLA trust specifications will be used for deriving service invocation primitives enriched with authentication, non-repudiation mechanisms, with or without the involvement of trusted third parties.

## 3.2. Specific objectives for the reporting period

The work within the TAPAS project has been structured into four technical workpackages:

- WP1 (Application Service Requirements and Specification) will meet the objectives related to SLA specification, Service Composition and Analysis Techniques;
- WP2 (Design of QoS-aware Infrastructure for Application Hosting) and WP3 (Implementation of QoS-aware Core Services) together will meet the objectives related to Trusted and QoS-aware Services for Application Hosting; and
- WP4 (Case Studies and Evaluation) will meet the objectives related to assessment.

First year work has been concerned with developing an overall framework for the development of software tools and techniques for hosting advanced distributed applications. We need to answer a few basic questions: What are the requirements of distributed application hosting? How to specify SLAs? What does "interactions with other organisations to be strictly controlled and policed" actually mean? What is trust and how to 'manage' it? How to incorporate QoS in component middleware? These questions have been answered in the first year deliverable reports: D1 (requirements), D2 (SLA specification), D5 (TAPAS architecture, describing trust management and regulation of inter-organisation interactions), and D7 (TAPAS middleware platform: QoS enabled application servers).

# 4. Achievements and project status

## 4.1. Summary

In the TAPAS project, we are particularly interested in developing solutions to the problem faced by Application Service Providers (ASPs) when called upon to host distributed applications that make use of a wide variety of Internet services provided by different organisations. This naturally leads to the ASP acting as an intermediary for interactions for information sharing that cross organisational boundaries. Essentially this means that an ASP should be capable of hosting Virtual Enterprises[1] (VEs): meaning, it should be capable of providing facilities for forming and managing VEs. The main problem in VE management is how enterprises can regulate access to their resources by other enterprises in a way to ensure that their individual policies for information sharing are honoured. Regulating access to resources by other enterprises is made difficult, since each potentially accessible enterprise might not unguardedly trust the others. Enterprises within a VE will therefore require their interactions with one another other to be strictly controlled and policed. And in this context, there will be a clear need among all parties to embark upon their business relationships underpinned by guarded trust management procedures. How can this be achieved?

Further, we note that in the near future ASP will require a distributed execution environment with a number of core services capable of meeting specific non-functional requirements of fault tolerance, availability, security, and timeliness; we will refer to these as *QoS enabled services*. State-of-the-art application services are developed using component-based technologies, such as those provided by the Java 2 Enterprise Edition (J2EE), Microsoft's .NET or the Object Management Group's CORBA Component Model. These technologies support the specification of functional component interfaces. They, however, do not adequately support the definition of the non-functional characteristics of component execution.

The first year work has been concerned with the design of the overall TAPAS architecture. Three main subsystems have been identified:

(i) The QoS management, monitoring and adaptation layer is intended to make the underlying application server QoS enabled. It is responsible for reserving the underlying resources necessary to meet the QoS requirements of applications hosted by that application server, and monitoring the reserved resources, and possibly adapting resource usage (e.g., reserving some more) in case the QoS delivered by these resources deviates from that required by the applications.

(ii) All cross-organisational interactions performed by applications are policed by the Inter-Organisation Interaction regulation subsystem. We have developed techniques that enable relevant aspects of contracts to be converted into electronic contracts (x-contracts) and

---

[1] A Virtual Enterprise comprises *n* independently existing and possibly mutually suspicious enterprises each of which wishes to establish a close business relationship for an agreed period of time without loosing its independence (autonomy).

represented using state machines and role based access control (RBAC) mechanisms for run time monitoring and policing.

(iii) It is necessary to ensure that a hosted application actually meets the QoS requirements (e.g., availability, performance) stated in the hosting contract SLAs. For this reason, we need an application level QoS monitoring service, which must also measure various application level QoS parameters, calculate QoS levels and report any violations. That is the function of the third subsystem shown in the figure.

We can see that QoS monitoring is occurring at two distinct levels: within an application server for controlling use of application server resources and at higher level for controlling application level QoS requirements.

Design of the first two subsystems has been carried out. The third subsystem will be designed as a part of second year work.

### 4.2. Overview of progress made in each workpackages

*D1: Application Hosting and Networking Requirements*

The first part of the deliverable identifies application hosting requirements. Therefore it focuses firstly on the problem domain and inspects the assumption regarding the current situation of the parties involved in ASP business. Furthermore, assumptions about future ASP business in a QoS-enabled environment are motivated, on which the discussion of the requirements is based.

As the next step we discus a model for ASP stakeholders, which is derived from the assumptions as well as from the roles in today's ASP business scenarios. The interests of the stakeholders are usually expressed in Service Level Agreements (SLAs). Content and typical kinds of existing, industrial relevant SLAs are briefly inspected as a preparation for a more detailed discussion of the basic guarantees and commitments defined in SLAs. However, this discussion is preceded by the identification of well-known and upcoming standards that are related to the fields of ASP, software engineering and networking.

The discussion of the guarantees and commitments defined in SLAs highlights the thematic areas of SLAs in a QoS-enabled scenario such as performance, availability, reliability etc.

A dedicated section summarizes the assessment criteria, structured in technical and economic criteria. This discussion is of course based on the observations and results of the previous sections.

The second part of the deliverable discusses QoS networking requirements with respect to internet service models and network SLAs. It includes a discussion of existing technologies such as DiffServ and Multicast as well as inspections on future directions of network SLAs. This includes peering and multihoming, which lead to issues regarding network SLAs between different ISPs. Furthermore, transport level SLAs are discussed, which offer more appropriate QoS facilities than the traditional ISP SLAs and are truly horizontal SLAs. Especially the timescale for execution of transport level SLAs turns out to be much shorter.

Besides these topics statistical models for end-to-end performance, formulas for throughput and expected time are identified, which in turn support the measurement of SLAs and thereby monitoring.

*D2: Specification Language for SLAs*

The term `electronic business', or `e-business', refers to the execution of transactions of commercial significance in a distributed computing context. Such transactions may involve an organisation, its clients and its partners, or they may be internal to an organisation, integrating separate computational assets.

E-business is impeded by technical integration barriers. Recently, standardisation processes have begun to catch up with commercial development and functional integration is being enabled by two forces:

Standardisation of component-based middleware architectures, and standardisation of communication protocols, particularly those based on Internet communication protocols and data formats. These technologies are complementary, and when employed in a client/server situation their use is often termed Application Service Provision (ASP) or web-service provision, if HTTP is used as the underlying transport protocol.

Functional integration may also include the provisioning of infrastructure by one organisation for another, as in the case of Internet Service Provisioning (ISP), Storage Service Provisioning (SSP) and application hosting. This provisioning relies on the use of standardized and established architectures and technologies.

Unfortunately, combining functionality is not the only requirement for e-business integration. Non-functional or quality requirements must also be met. Moreover, businesses must initially meet, negotiate the terms of their collaboration, have some confidence that the services that they purchase will meet their requirements, and that they in turn can meet their client's expectations. Efforts have been made to establish business-to-business marketplaces, in which application services can be traded. Our work stands in the context of these efforts, but addresses the need for description and negotiation of Quality of Service (QoS) properties.

QoS has considerably been investigated for network protocols and services. We take some of those ideas to the middleware and application level. To do so, we introduce a reference model for inter-organisational service provision at storage, network, middleware and application level. The model provides the basis for the definition of SLAng, a language for Service Level Agreements (SLAs). SLAs capture the mutual responsibilities of the provider of a service and its client with respect to non-functional properties. Our language SLAng meets multiple objectives: It provides a format for the negotiation of QoS properties; the means to capture these properties unambiguously for inclusion in contractual agreements; and a language appropriate as input for automated reasoning systems or QoS-aware adaptive middleware. We have evaluated the expressiveness of SLAng using a case study that supports the QoS-aware implementation of web services for image processing.

Outline of the deliverable: The first chapter looks at the big picture of e-Business automation. It provides a survey of the evolution of e-Business and describes the overall thread between the

new technologies which have been proposed. The solutions chosen by the TAPAS project are, this way, set into the scene. Chapter 2 presents the middleware architecture SLAng is based on, for the provision of application services. In this chapter we first present middleware issues for the execution of distributed processes. We then move on to discuss the concepts of SLAng. Chapter 3 introduces SLAng features. Its basic structure and key concepts are outlined, in order to set the framework for Chapter 4, where we illustrate in details its constructs. There we present the QoS parameters that govern service components deployment, explaining how they are organised in SLAng language. Provided the possibility to specify non-functional aspects while deploying service components, Chapter 5 discuss a case study we have been testing SLAng efficacy on. An overall evaluation of the work is given in Chapter 6 together with a presentation of our research agenda.

*D5: TAPAS Architecture: concepts and protocols*

The aim of report D5 is to describe the interim TAPAS architecture for application hosting. To meet this requirement this document discusses the core concepts and algorithms needed to reason about and to build the TAPAS architecture. It develops the concept of virtual enterprise and describes how interactions between members of such an enterprise can be regulated by means of electronic contracts. Deliverable D5 "TAPAS architecture: concepts and protocols" contains four main contributions:

1. A virtual enterprise (VE) model as a generalisation of hosting distributed applications by an ASP. We explain how business partnerships where members interact through the Internet can be built. To help reasoning about the business interactions between the members of business partnership we introduce a model for building virtual enterprises,

2. Next, we discuss how business interactions between two or more business partners can be regulated. We develop the concept of *executable contracts* (*x-contracts*) using finite state machines (FSMs) that can be used for monitoring and enforcing at run-time the right and obligation of each business partner.

3. We define behavioural trust as *the mutual judgement of and dependence on the expected behaviours of one another's organisations in specifically agreed inter-organisational interactions, such that there is a mutual feeling of relative confidence in these interactions, even though negative consequences are possible*. Such negative consequences will appear as violations of trusted interactions, and thus be interpreted as failures of agreed trust relationships. TAPAS is accordingly investigating techniques for fairness in negotiable multi-party trust agreements, with specific reference to the requirements of role-based access control (RBAC) policies for authentication, integrity and authorisation. Our early work in the area of fair exchange protocols is presented in the Appendix of D5. Moreover, this appendix also includes our initial ideas on formalising trust relations.

4. Any software designed to implement an x-contract will contain two main components: a *contract monitor and enforcer* and a *middleware service* for regulated, non-repudiable information sharing. The contract enforcer is a piece of software that guarantees that the rights and obligations stipulated in the contract are monitored and enforced. A promising approach that seems to meet the requirements of the middleware service that we need is the B2Bobject middleware developed by us. The B2Bobject middleware collects non-repudiable evidence

about information sharing between parties that do not necessarily trust each other. Once deployed, each party holds a local copy of shared information encapsulated in objects. Access to and update of this information is subject to non-repudiable validation by each party. It is assumed that each organization has a local set of policies for information sharing that is consistent with the overall information sharing agreement between the organizations (this agreement will be encoded in the x-contract). The safety property of our system ensures that local policies of an organization are not compromised despite failures and/or misbehaviour by other parties; whilst the liveness property ensures that if all the parties are correct (not misbehaving), then agreed interactions would take place despite a bounded number of temporary network and computer related failures. The Appendix contains details of the B2Bobject middleware.

*D7: TAPAS Architecture: QoS Enabled Application Servers*

The Report D7 describes the architecture of the TAPAS middleware platform that will be developed as part of the TAPAS project. This architecture consists of a collection of middleware services that extend the abstraction of application server, as provided by current middleware technologies, such as the CORBA Component Model and Java 2 Enterprise Edition (J2EE), in order to meet Quality of Service (QoS) application requirements, such as performance, reliability, and security. The motivations for this architecture can be summarized as follows.

Current component-oriented technologies allow designers to construct distributed applications out of reusable and interoperable software components (e.g., commercial operating systems, communication protocols, middleware services). These technologies support the specification of the functional component interfaces; however, they support only partially the definition of non-functional properties (i.e., the QoS) of the component execution.

These technologies promote the use of containers to host application component instances. Specifically, a container provides the run time environment for those instances, and shields them from the complexity of most of the system services, such as the transaction, security, persistence, and notification services. Hence, containers take part in the management of the non-functional properties of the components they host.

Several containers can be hosted by the same application server; thus, QoS negotiation, establishment, and adaptation facilities can be added to the application server and used by component containers to make them QoS-aware.

These facilities are implemented in the TAPAS middleware architecture by two principal middleware services, named Configuration Service and Controller Service, respectively, that can be used to extend an application server. The former service is responsible for discovering, negotiating, and reserving the resources necessary to meet the QoS requirements of a particular application component, hosted by that application server; the latter service is responsible for monitoring the reserved resources, and possibly adapting the component execution in case the QoS delivered by these resources deviates from that required by the component itself.

The TAPAS architecture described in this Report uses Service Level Agreements (SLAs), as discussed in the TAPAS deliverable D1, in order to derive the QoS application component requirements, and monitor the delivered QoS at the component run time. Thus, SLAs, in the TAPAS architecture, are used not only as an inter-organizational contractual feature, but also to govern the component execution.

In order to define the TAPAS architecture, we have carried out an extensive assessment of the current state of the art in the design of architectures developed to meet QoS requirements of distributed applications. From this assessment it has emerged that none of the architectures we have examined fully meet the TAPAS objectives; however, this assessment has allowed us to derive a number of recommendations and design principles we have deployed in the definition of our architecture. These include both such recommendations as the need for incorporating a resource monitoring service in our architecture, in order to assess the resource state at run time, as well as design principles such as those that can be derived from the control theory, in order to deploy adaptation facilities.

Further, we have examined two use cases (namely, the hosting of a generic application by an Application Service Provider, and that of specific auction application) in order to expose specific requirements the TAPAS architecture has to meet. As a result of this activity, we have identified a detailed set of functionalities our architecture is to incorporate in order to meet its objectives.

We have then studied how to integrate these functionalities in existing component-based middleware platforms. Specifically, we have examined the integration of these functionalities within the J2EE application server, and their instantiation within two specific implementations of the J2EE platform; namely, Jboss and JOnAS. In addition, in this context, we have examined how the proposed TAPAS architecture would enable the deployment of specific replication technologies.

*D12: First year Evaluation and Assessment Report*

The intention of deliverable D12 is to summarize the project activities up to the end of the first year. We therefore firstly discuss the progress with respect to description of work of the proposal by focusing on the already started workpackages WP1, WP2 and WP3. WP4, referring to assessment and evaluation is in it's very first phases and therefore implicitly addressed by sections about publications and dissemination activities. We discuss the current progress in the workpackages by focusing on the deliverables that are associated to the workpackages. However, we only discuss the already existing or already started deliverables, because it is obviously inappropriate to reason about non-existing documents, even if the basic ideas have been discussed in the workshops. As an observation it can be said that monitoring turns out to be a very important issue with much influence on most activities.

As a summary it can be observed that good progress has been achieved, not only in the dedicated areas of the single deliverables, but as well in synergetic combination of the research areas such as middleware architecture and SLA specification.

Furthermore, D12 reflects the feedback of the Industrial Advisory Board and the TAPAS-related publications. In the area of dissemination activities conferences and cooperations with other research projects are listed. This is completed by an overview of further dissemination activities by adesso, consisting in articles in German business magazines like “Versicherungswirtschaft”.

# 5. Adherence to Workplan

## 5.1. Statement of resource usage

Effort in Person months for reporting period 1/4/02 – 31/3/03

### 5.2. Deliverable schedule update

No changes to the first year schedule were made. The titles of two deliverable reports were changed to better reflect their technical contents. D5 is entitled 'TAPAS Architecture: Concepts and Protocols' rather than ' Architectural Design Document' and D7 is now called 'TAPAS Architecture: QoS Enabled Application Servers' rather than 'QoS Container Interface Specification'.

The work for the second year will be concerned with integrating the results of this year and producing working implementations. The following deliverables will be produced:

D3: Method for Service Composition and Analysis

D8: Container for Group Communication

D9: Container for Trusted Coordination

D10: Container for QoS Monitoring

D13: Second Year Evaluation and Assessment Report

D17: Updated Dissemination and Use Plan

### 5.3. Modifications to Workplan

No modifications have been made.

## 6. Co-operation in project

### WP1: Application Service Requirements and Specification

The aim of Workpackage 1 is to identify the requirements of trusted and qos aware application service provision. This aim has been achieved within the first year of the project to a large extent.

The project has met in a number of plenary meetings in Bologna in April 2002, Cambridge in September 2002 and Dortmund in February 2003 to discuss requirements and an approach to defining quality of service at different levels of abstraction. In addition to these plenary meetings, workshops between partners involved in WP2 were held at UCL (twice) and Bologna.

As a result of this effort, we have two joint deliverables (D1 and D2). D1 defines the requirements for application hosting from an industrial point of view. For D1, the first major task was to develop a document containing the application hosting requirements based on the industrial experience of adesso. This document was combined with a document about networking requirements. The networking part was developed by the Cambridge TAPAS team, Jon Crowcroft and Panos Gevros. The requirements document serves as a basis for the further research and development activities.

These requirements have influenced the definition the Service Level Agreement Language (SLAng) as defined in D2. The language has been evaluated in practice using different application hosting scenarios provided by the Common Picture eXchange environment (CPXe) of the International Imaging Industry Association (I3A) in cooperation with Kodak Ltd, with whom UCL loosely cooperate. Feedback on SLAng from other partners of the consortium is still outstanding (as the language was only completed a month ago) and might lead to an updated version of the language.

### WP2: Design of QoS-aware Infrastructure for Application Hosting

TAPAS architecture will use SLAs not only as an inter-organizational contractual feature but also to govern component execution. To this end, QoS negotiation, establishment, and adaptation facilities will be added to the TAPAS middleware and these will be used by component containers to make them QoS aware. This workpackage has four tasks: middleware architecture, trust management, network control architecture and component execution environment.

We believe that the concept of virtual enterprise is central to the TAPAS project. After studying the requirements document (D1), in particular the market place scenario, we came to the conclusion that business interaction between the APS and the owner of the market place is just a special case of a general problem, namely that of building general purpose business partnerships. It is conceivable that an enterprise might wish to take part in more than one business partnership at the same time. For this to be possible, enterprises need to keep their independence after joining a business partnership. The result of this is that a business partnership can be regarded logically as a new and independent (from its creators) enterprise.

We are interested in implementing business partnerships electronically and call the new enterprise that results from the implementation *a virtual enterprise (VE)*.

We then examined how to regulate interactions between enterprises within a VE. This led us to issues of trust management in VE. We have discussed this at length in D5; the concept of *trust propositions* are of the general form:

A *trusts* B *on matters of* X *at epoch* T

Here, *A* and *B* may be people, computers and their specific resources and services, or even small or large enterprises that admit to *trust relationships*. In the proposition, *A* is placing a trust relationship (*dependence*) on *B* with respect to matters of *X*. Such matters constitute the set of *rights and obligations* of *A* with respect to *B*, such that *B* permits access to specific resources (services) provided by *B* to *A* provided that *A* fulfils specific *obligations* (*conditions*) laid down by *B*. Epoch *T* represents the period during which both *A* and *B* observe the well being of the their trust relationship without incidence of failure.

Our work on executable contracts, we examined how to represent rights and obligations, and how access to specific resources can be controlled. The approach we found most satisfactory was the use of finite state machines (FSMs) for representing contracts. We worked on how rights and obligations can be encoded in FSMs. We have also found that existing work on role base access control (RBAC) fits well with the requirements of VE.

An important requirement of regulating interactions within a VE is the generation of non-repudiable evidence. We need middleware that provides generic services that can be used to support arbitrarily complex interactions between contracting parties. From the viewpoint of each party involved, the overarching requirements are (i) that their own actions meet locally determined policies; and that these actions are acknowledged and accepted by other parties; and (ii) that the actions of other parties comply with agreed rules and are irrefutably attributable to those parties. These requirements imply the collection, and verification, of non-repudiable evidence of the actions of parties who interact with each other. An example of evidence that can be collected is a non-repudiable record that a payment was placed on a certain date. Our work on B2Bobject was influenced by these observations. In D5 we explain with the help of a purchaser-supplier example, how B2Bobject middleware ensures that all operations performed by the purchaser and the supplier are recorded and are non-repudiable. One of the major advantages of B2Bobject is that it ensures this without the need of involving centralised trusted third parties.

Work on WP2 was undertaken in close collaboration with WP1 and WP3. Number of meeting were held to discuss issues of mutual importance. Details of these meetings are given in the description of WP3 (last paragraph).

**WP3: Implementation of QoS-aware Core Services**

The activity in WP3 has consisted principally of: i) assessing issues of design of end-to-end QoS architectures, suitable for deployment within the context of the TAPAS project, ii) investigating issues of trust management, iii) evaluating a number of candidate middleware platforms to be used for the scopes of the TAPAS project, and iv) defining an (interim) TAPAS

middleware architecture. The results of this activity are fully described in the deliverable reports D7 and D5, and can be summarized as follows.

We have carried out, firstly, an analysis of the design issues involved in the provision of end-to-end QoS in geographically distributed environments. To this end, we have examined a number of end-to-end QoS architectures proposed in the literature, including TAO, RT CORBA, Agilos, QuO, and architectures whose design is based on the feedback control theory. As a result of this effort, we have identified a variety of properties and characteristics an end-to-end QoS architecture should possess in order to meet the TAPAS project objectives. In addition, we have developed an initial architectural proposal in which the Java Enterprise abstraction of container is extended to incorporate services for end-to-end QoS provisioning, monitoring, and maintenance. Moreover, issues of QoS-enabled group communications have been addressed, as part of the WP3 activity, as the group communication paradigm is recognized, within the TAPAS project, as the basic abstraction that can support the implementation of the middleware services above mentioned.

Secondly, we have examined a number of candidate middleware platforms that can be used for the purposes of the TAPAS project. Specifically, we have examined the SUN J2EE platform, and its open source implementations JBoss and JOnAS. Both these J2EE implementations appear to be suitable for integrating the TAPAS middleware services mentioned above; we are currently evaluating the possibility of developing a solution that can be deployed in either of these J2EE implementations

Finally, as part of the WP3 activity, we have examined a number of trust management systems, including the Grid CAS and the Globus Toolkit GSI, Pics, Policymaker, Keynote, RBAC, RT, and OASIS, and proposed to incorporate, within the TAPAS middleware, appropriate mechanisms that implement role based access control policies, and manage effectively dynamic trust relationships. An experimental evaluation of these mechanisms is currently being developed.

The activity in WP3 has greatly benefited from meetings and discussions with the partners involved in the development of WP1 and WP2; in particular, the work on application service requirements and specification, developed in WP1, and that on the design of a QoS-aware infrastructure for application hosting, developed in WP2, has provided us with crucial input for the work developed in WP3. Moreover, it is worth mentioning that, the activity in WP3 has been carried out in close collaboration with our colleagues at the University of Newcastle upon Tyne, principally, and has benefited from a number of technical meetings we had with our colleagues of University College London (UCL), University of Cambridge, and Adesso AG. Specifically, in addition to the scheduled project meetings, technical meetings between project members of the University of Bologna and project members of UCL were held in Bologna (7-8 October, 2002) and London (18-20 January, 2003); technical meetings between project members of the University of Bologna and project members of the University of Newcastle upon Tyne were held in Newcastle upon Tyne (6 - 19 December, 2002), and Bologna (24 - 28 February, 2003).

**WP4: Evaluation and Assessment**

The aim of WP 4 is to evaluate and assess the TAPAS results. Due to the fact that the project still is in a quite early phase, the actual activity in WP 4 up to the time of writing was to compile

the deliverable D12, the first year assessment report. However, the assessment and evaluation has started not only with developing the report. Early feedback between industrial partner and research partners has been achieved by intense communication and fruitful team workshops.

Especially the strong knowledge of state-of-the-art component technology such as Enterprise JavaBeans has enabled the adesso team to participate in discussions with valuable contributions.

Current assessment activities include approvals of applicability of the developed methods and technologies in discussions with experienced adesso experts. This will not only help to identify technical issues, but will as well guide the activities into business-relevant directions.

## 7. Co-ordination with other projects/programmes

Santosh Shrivastava, Fabio Panzieri, Jon Crowcroft: participated in a Cluster meeting on engineering of service functionality that was held during the IST Broadband Networking Conference, Bucharest, October 02. Since then, there have been plans to create A Service Engineering Co-ordination Action under FP6.

Santosh Shrivastava is coordinator of MIDAS accompanying measure. The objectives of the MIDAS are (i) to create a research roadmap for the development of the next generation middleware capable of supporting services that are composable and adaptable within the context of large scale systems where quality of service issues ranging from fault-tolerance, timeliness to security and survivability are of paramount importance; and (ii) to build Europe wide partnerships for creating an integrated project (IP) in the Framework Six Programme (FP6).

Wolfgang Emmerich is involved in SEGRAVIS (Syntactic and Semantic Integration of Visual Modelling Techniques) project

Santosh Shrivastava, Graham Morgan attended the ADAPT project (IST-2001-37126) Kickoff meeting in Madrid, Sept 02; TAPAS and ADAPT have specific collaboration on trust management.

# 8. Promotion / Information Dissemination

## 8.1. Conferences and Workshops

*TAPAS project members attended the following conferences during the first year of the project*:

Jon Crowcroft attends the Global Grid Forum meeting, where he co-chairs the working group on High Performance Network requirements.

3rd, International Workshop on Software Engineering and Middleware, Orlando, Florida attended by W. Emmerich

23rd International Conference on Software Engineering, Orlando, Florida, May 2002 attended by W. Emmerich

3rd. International Workshop on Software Performance (WOSP), Rome, July, 2002 attended by D. Lamanna and J. Skene

ACM SIGCOMM 2002, Pittsburgh, USA, 23-25 August, attended by Jon Crowcroft

W. Emmerich: 17th IEEE Int. Conference on Automated Software Engineering, Edinburgh, Sept 2002.

2nd IEEE International Conference on Peer-to-Peer Computing, Linköping, Sweden, 5-7 Sept. 2002, attended by E Turrini.

4th European Dependable Computing Conference, Toulouse (F), 22-25 Oct., 2002 (attended by N. Mezzetti)

Santosh Shrivastava and Paul Ezhilchelvan: IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-2002), June 2002, Washington DC

Santosh Shrivastava: Workshop on Future Directions in Distributed Computing (FuDiCo), Bertinoro, Italy, June 02.

Santosh Shrivastava, Fabio Panzieri, Jon Crowcroft: IST Broadband Networking Conference, Bucharest, October 02.

Giovanna Ferrari attended 7th Cabernet Radical Workshop, Bertinoro (FC), Italy, 13-16 Oct. 2002.

Several members of the Project attended the MIDAS Workshop on "Middleware for Composable and Adaptable Services", Montreux, 14-15 November 2002.

2nd IEE Vacation School on Distributed Software Architecture (September 2002), with talks from Wolfgang Emmerich and John Crowcroft to industrial participants from throughout the UK. For details see http://mcs.open.ac.uk/computing/resg2/documents/BDSA.pdf

7th Cabernet Radical Workshop, Bertinoro (FC), Italy, 13-16 Oct. 2002 (attended by G. Lodi, E. Turrini).

## 8.2. Publications

G. Morgan, A. I. Kistijantoro, S. K. Shrivastava and M. C. Little, “Component Replication in Distributed Systems: a Case study using Enterprise Java Beans” Technical Report, Dec. 2002. (submitted for publication)

J. Skene and W. Emmerich. Model Driven Performance Analysis of Enterprise Information Systems. In Proc. of International Workshop on Test and Analysis of Component Based Systems, Warsaw, April 13th, 2003 in conjunction with European Joint Conferences on Theory and Practice of Software (ETAPS) 2003, Electronic Notes in Theoretical Computer Science, Elsevier Science B. V. To appear.

D. Lamanna, J. Skene and W. Emmerich. SLAng: A Language for Defining Service Level Agreements. Accepted for Poster presentation, Middleware 2003, Rio de Janeiro, Brazil

D. Lamanna, J. Skene and W. Emmerich. SLAng: A Language for Defining Service Level Agreements. In Proc. of The International Workshop on Future Trends of Distributed Computing Systems (FTDCS'2003), San Juan, Puerto Rico. IEEE Computer Society Press. To appear.

W. Emmerich. Distributed Component Technologies and their Software Engineering Implications. Proc. of the 24th Int. Conference on Software Engineering, Orlando, Florida. pp. 537-546. ACM Press. 2002.

G. Piccinilli, W. Emmerich and C. Zirpins and Kevin Schuett. Web Services Interfaces for Inter-organizational Business Processes: An Infrastructure for Automated Reconciliation. In Proc. of the 6th IEEE Int. Conference on Enterprise Distributed Object Computing, Lausanne, IEEE Computer Society Press. pp. 285-292. 2002.

W. Emmerich and N. Kaveh. Component Technologies: Java Beans, COM, CORBA, RMI, EJB and the CORBA Component Model. Proc. of the 24th Int. Conference on Software Engineering, Orlando, Florida. pp. 691-692. ACM Press. 2002.

A. Aldini, M. Bernardo, R. Gorrieri & M. Roccetti, “QoS Evaluation of IP Telephony Services: A Specification Language Based Simulation Software Tool", Systems Analysis Modelling Simulation, Taylor and Francis Group Pub., accepted for publication, December 2002.

N. Mezzetti, F. Panzieri, "The Data Grid: Security and Privacy Issues", Proc. 4th European Dependable Computing Conference, Toulouse (F), 22-25 Oct. 2002.

G. Lodi, " End-to-end QoS-aware Middleware Services", 7th Cabernet Radical Workshop, Bertinoro (FC), Italy, 13-16 Oct. 2002.

E. Turrini, "A Platform for Request Routing in Content Distribution Inter-networks", 7th Cabernet Radical Workshop, Bertinoro (FC), Italy, 13-16 Oct. 2002.

E. Turrini, F. Panzieri, "Using P2P Techniques for Content Distribution Internetworking: A Research Proposal", in proceedings of the 2nd IEEE International Conference on Peer-to-Peer Computing, Linköping, Sweden, 5-7 Sept. 2002.

N. Cook, S.K. Shrivastava and S.M. Wheater, "Distributed Object Middleware to Support Dependable Information Sharing between Organisations", IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-2002), June 2002, Washington DC.

S.K. Shrivastava: Middleware for supporting inter-organisational interactions, Proceedings of Workshop on Future Directions in Distributed Computing (FuDiCo), Bertinoro, Italy, June 02.

### 8.3. Press Articles

Among the public relations activities addeso started in the first year are articles in business magazines in order to raise interest not only in the technological oriented or as well scientific community, but to influence decision makers. Hence adesso started public relations activities to make the TAPAS project known in the industry. This is achieved by placing articles in German business magazines.

Up to now the TAPAS project has been mentioned in the following articles:

- Article „Marktbelebung durch mehr Sicherheit und Qualität" („Market upturn by more security and quality") in „Versicherungswirtschaft" ( „Insurance economy") pp 67, Verlag Versicherungswirtschaft GmbH, 1/2003, Karlsruhe, Germany
- Article „EU-Forschungsprojekt fördert ASP-Markt" (EU research project encourages ASP market) in „Industriemanagement" („Industrial management"), pp 77, GITO mbH Verlag,1/2003, Berlin, Germany

Currently we plan to expand the public relations activities by placing articles in further magazines like "Computerwoche" ("Computer week").

### 8.4. Activities related to standardisation

UCL joined OMG, with a view to reinforce the relationship established with the OMG through Andrew Watson's participation in the IAB and to contribute findings of TAPAS to OMG's standardization efforts on MDA, UML profiles for EJB, real-time and scheduling.

### 8.5. Industrial Advisory Board Meetings

The members of the IAB were invited to meet with the TAPAS Executive Board at the TAPAS EB/IAB Meeting, held in Cambridge on 8th July 2002. Five members of the IAB attended: Dr. Tobias C. Kiefer (Commerz NetBusiness AG), Andrew Watson (Technical Director of the OMG), Prof. Dr. Rudolf K. Keller (Zühlke Engeneering), Dr. Stuart Wheater (Arjuna Technologies), Paul McKee (BT exact Technologies).

This section summarises the remarks made by these IAB members with regard to the progress of TAPAS by July 2002, as evidenced by the presentations.

During the meeting the IAB members were interested to learn more about the project's aims and it's context regarding different participants and their main tasks.

The project partners presented their approaches to the different TAPAS issues, like the ASP requirements, the SLA modelling language, TAPAS middleware architecture and multicast

protocols. At the time of the meeting the application hosting and network requirements had roughly been identified and classified in a draft version of deliverable D1.

The IAB members expressed the general impression from the presentations that all research activities are heading in the same direction. In turn the feedback of the project team was that all project partners profited from the comments of IAB, which expressed real-life concerns on TAPAS topics.

Stuart Wheater asked whether the TAPAS project intended to formalise SLAs. In fact, the activities for definition of SLAng include the formalization.

Rudolf K. Keller was interested in the different levels of SLAs, namely application, system and network levels and how it will be differentiated in the modelling language. He also asked about the project research management and how all work packages fit together. He remained positive about the progress of the project.

Paul McKee initiated a lively discussion about the business requirements at the application level.

There was an interesting exchange of views about authentication mechanisms in different EU countries, especially about digital signatures. Tobias C. Kiefer expressed serious concerns regarding PKI models. Regarding his banking background PKI is hardly used in the practice, because there are no economical models at the technical level, e.g. there are still different standards in different countries.

The other comments of the IAB members regarding their perception of the progress of TAPAS all referred to the security and trust issues. On the one hand it is necessary to deal with certificates to standardise work across international systems. On the other hand Paul McKee commented that companies probably would not want to pay the overhead for just one transaction.

Summarised by Jon Crowcroft saying that the economic approach as Tobias C. Kiefer mentioned is a very viable one to investigate.

The Second IAB meeting was held on 3 April, London, attended by Dr. Tobias C. Kiefer, Dr. Stuart Wheater (Arjuna Technologies), Paul McKee (BT exact Technologies). We presented the results of the first year work, along the lines of the Review. The discussion, in terms of Questions and Answers is summarised below.

Q (Wheater) Can clustering influence the architecture designed in D7? How? Where to use it?

A (Panzieri) Probably it will be useful in the Configuration Service, Bologna is investigating this possibility

A (Shrivastava) It will be interesting to know which parameters to control in clustering and how to do it.

Q (Mckee) How to manage different instances of the same application (maybe on physical different machines, maybe not) with different SLA? (example: the ASP is hosting an application of a client with some SLA, but the client would like to invest more money on that specific

application, receiving a better service from the ASP, that implies to change the SLA for the given application. How does it work in this case? It is better to use a different SLA than add clauses to the old one, it could add complexity in hosting and running the application)

A (Emmerich) We have to understand how to implement the monitoring of different SLA at different levels. Do we need different components, in accordance with different levels of vertical agreements (routers/databases/containers)? We could address just one solution, focusing on one type of monitoring, without trying to provide the overall solution.

A (Crowcroft) or we could use different techniques and tools for monitoring different levels (ex: there is plenty of tools for network monitoring)… maybe linking together all the results in order to provide a detailed report.

A (Shrivastava) The given view monitors just the middleware level, and the application server, we need to investigate if it is possible provide monitor at higher levels (example: there are companies that provide to the customers monitoring of services or hosted applications provided by other companies, working as TTP as well)

A (Beckmann) Let's focus just on the middleware, there is no academic research on the application servers monitoring, instead there are many tools already used for DB and network monitoring.

A (Emmerich) It could be possible to use different approaches for different modelling of performance monitoring (statistical modelling, object oriented, queuing), but carefully: it would be confusing.

Q (McKee) asked about how exception reports were being routed.

A. During the discussion it was noted that some exception at a higher-level of abstraction might be caused by an exception at a lower abstraction level. Moreover, it was noted that the party interested in learning about an exception might not necessarily be the one that experience a lower service level.

Q (Wheater) asked about how we could check semantic consistency between the different levels of abstraction.

A. It was explained that this was being done by the work on QoS analysis and design that is being done at UCL and will be explained in a deliverable due at 18months. The approach taken is that UML profiles are being used to express QoS parameters in a design and then these profiles are mapped into some analysis domain (e.g. Queuing networks, stochastic process algebras or Bayesian belief networks).

## 9. General Comments

The first year activities already have utilized recent developments and research topics such as Model Driven Architecture (MDA), which has been incorporated into the Model Driven Performance Analysis (MDPA) research.

Besides technical aspects we see, especially in Germany, an economic downturn, that leads to reinforced outsourcing activities. This is reflected in various articles on ASP in the German press.

Since TAPAS started, a Web Services have arrived on the scene and are rapidly being adopted in industry. In order to ensure validity of the TAPAS results, Web Services have been incorporated into the Service Level Agreement Formalism defined at UCL.

## 10. Conclusion

The project has produced all the deliverables on time and is running on schedule.