

# **Agent based process management : applying intelligent agents to workflow**

P.D.O'Brien and W.E.Wiegand

## **Abstract**

This paper presents an Agent-based Process Management System (APMS) architecture that combines recent developments in autonomous agent technology and distributed computing platforms. It argues that a *service-oriented* view is required in order to meet the requirements of open distributed enterprises, now and in the future. This approach embraces and fully supports the decentralisation of responsibility and local autonomy, within and across enterprises.

The paper begins by summarising existing approaches to workflow. It describes the APMS approach which extends workflow with the abilities to anticipate process requirements, to resource processes dynamically, and to adapt to exceptions. An important feature of APMS is the absence of a centralised business process representation.

Finally, the paper describes an implementation of the APMS architecture, and discusses the engineering issues related to realising such a system within a large organisation.

## **1. Introduction**

Organisations increasingly are becoming distributed. The globalisation of trade, flattening of traditional management structures, and the advent of networked end-user computing all have influenced the shape of the modern organisation. Agent-based process management systems (APMS) combine the latest distributed computing techniques with autonomous software agent technology, to provide a solution to the problems of managing the distributed organisations of the next millennium. APMS represent an extension to the capabilities of existing workflow management systems.

This paper outlines the motivation for using agent technology for business process management and the key benefits of adopting such an approach. It describes how APMS take a service-oriented view of a business process and how they build upon the latest techniques from distributed computing and intelligent systems techniques, and utilise software *agents* to bring on-line aspects of service provision. This approach embraces and fully supports the decentralisation of responsibility and local autonomy, within and across enterprises. Example APMS are given from project ADEPT and project BeAT.

### **1.1. Background**

The 1980's saw a radical change in how organisations were managed and structured. The expansion and liberalisation of international markets resulted in the delayering and downsizing of organisations. Traditional hierarchical, bureaucratic structures [1,2] were replaced with decentralised and semi-autonomous business units. Organisations saw their markets grow internationally and increasingly become geographically distributed.

Structural change was accompanied by new management philosophies such as total quality management (TQM) [3]. This represented an ideological shift from a product driven focus to one which was more customer driven, placing increased emphasis on the quality of service received by customers. Such changes gave rise to the concept of the horizontal organisation [4] where an organisations focus was less on functional units or departments, and more on core business processes. Business processes crossed departmental boundaries and were structured with regards to an overall business objective. Horizontal organisations shifted the emphasis from managing up and down hierarchies to managing across organisations.

A new breed of software, co-ordination software [5] was developed to support the need for cross organisational management. This included computer supported co-operative working, group-ware, and

workflow. Co-ordination software enabled the linking of different applications from various departments in support of a single business objective. This software crossed the traditional departmental boundaries and integrated diverse activities in support of a single business goal. Workflow is the breed of technology developed to specifically manage business processes.

## **1.2. Existing Workflow Systems**

Workflow is the implementation and automation of a particular business process. A workflow management system (WMS) is the software which automates the co-ordination and control of tasks during business process execution, [6].

WMS represent one of the key enabling technologies that supports the management of distributed business processes across an organisation. The workflow approach helps to separate the business logic represented by a business process from the underlying information systems which support that process. This separation allows business processes to be designed without requiring major changes to be made to the underlying computing infrastructure. Workflow automates the enactment of business processes, improving the speed and efficiency of an organisation.

Different WMS exists to suit different types of business process, these have been classified into administrative, ad-hoc, production and collaborative workflow, [5,7,8].

- Administrative workflow systems involve repetitive, predictable processes with simple task co-ordination rules. Examples of administrative workflows would be routing documents within an organisation, which involves standardised tasks, tightly linked, and performed regularly.
- Ad-hoc workflow systems involve more human co-ordination where both process and information are relatively unstructured. An example would be a sales process which can be relatively unstructured and which becomes structured during the execution of the business process. In a sales process decisions can be made during its execution which will determine subsequent tasks. Ad-hoc workflow systems rely heavily on human involvement in controlling and co-ordinating tasks.
- Production workflow systems handle complex business processes which are more critical to an enterprise. These typically involve some form of transaction processing and require accessing multiple information systems. This would include processes such as customer handling and exception handling processes.
- Collaborative workflow supports business critical processes which are less structured and more suited to collaborative working technologies such as Lotus Notes. This is where the group working technologies overlap with the workflow market.

The main differentiating factor between these types of WMS is the structure of the processes and the complexity of the tasks. The process structure ranges from simple linear processes to processes with a high degree of concurrency, decision points, and dependencies between tasks. Task complexity indicates the degree of management functionality required by the WMS, and points to the sophistication of the information systems which support the tasks. However, despite this WMS still cover a relatively small number of typical business processes. Extending the range of WMS so they can be applied to less structured, more complex business processes requires the reassessment of WMS functionality.

## **1.3. Limitations of Workflow**

WMS have certain limitations that need to be addressed if they are to be applied more successfully and to more business processes. WMS at present are ideal for managing business processes which are fully dimensioned and where all logical paths have been carefully considered in detail and fully described, however, not all business processes are like this. In commercial environments decisions are not always clear cut but involve the balancing of various vested interests and business policies, and resource levels can change. Such business processes highlight a number of shortcomings in existing workflow management systems [9]. They lack:

- Reactivity: Workflow management systems require an *a priori* representation of a business process and all potential deviations from that process.

- **Semantics:** Many workflow management systems lack an appreciation of the content of a business process and do not make decisions based on the nature of the information generated by a business process.
- **Resource management:** Workflow management systems do not control the resourcing of a business process and so rely on a business process being dimensioned beforehand.
- **Heterogeneity:** Workflow management systems tend to take a centralised view with a single workflow management engine that does not operate across multiple-server platforms or multiple client operating systems.

#### 1.4. Agent-Based Process Management Systems

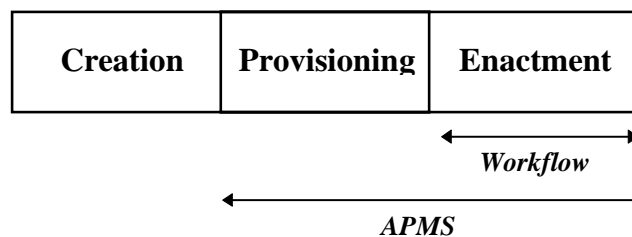


Figure 1 : Business Process Management Lifecycle

Agent based Process Management Systems (APMS) extends the automation of business process management beyond that covered by WMS. The management of a business process can be viewed as consisting of three stages, creation, provisioning and enactment<sup>1</sup>, [13]. The creation stage is predominantly a manual activity which involves the analysis, modelling and definition of the business process. Standardised representations are available for capturing process knowledge such as PIF (Process Interchange Format), [15]. The provisioning stage involves the assignment of resource, including people, equipment, computing time, to support a business process. This requires the negotiating, planning and scheduling to ensure that there is sufficient resource to handle expected throughput of work for a given business process. Lastly, the enactment stage involves the management activities required to ensure that each instance of a business process is effectively executed. This includes routing of work, passing of information, activation of automated activities, and the handling of work lists.

Existing workflow technology automates the enactment phase of the life-cycle, focusing on delivering a workflow enactment service. APMS extend this into the automation of the provisioning stage of the life-cycle model. Subsequently the dimensioning of business processes is brought on-line and integrated with process enactment, resulting in improved re-deployment of resources and increased flexibility during exception handling. Therefore, unlike WMS which are focused solely on the enactment of process tasks, APMS have two objectives, firstly, the timely execution of business tasks and secondly, the efficient use of resources.

APMS combine the latest distributed computing technology with agent-based techniques [10,11,12] providing an intelligent extension and alternative to workflow management systems. APMS allow the interests of different parts of an organisation to be represented by a software agent. The inherent autonomy of software agents enables these parts to retain their autonomy of information and control, and to enable different organisations to automate their interaction in the management of a common business process. Agent autonomy provides a mechanism to handle the inherent heterogeneity and modularity that exists in modern enterprises. APMS provide:

- **Intelligent decision making:** APMS can represent management strategy and policy from a range of perspectives.
- **Anticipation:** APMS plan tasks and schedule available resources in anticipation for their use in a business process.

<sup>1</sup> This lifecycle model is based on that developed in project ADEPT (see [12,13,14])

- Explicit resource management: APMS represent the levels, limitations and value of resources and manage them in support of business activities.
- Reactivity: APMS react to changing circumstances and have the capacity to generate alternative execution paths in response to unique exceptions.
- Heterogeneity: APMS can be distributed over multiple platforms across LANs and WANs using the latest distributed computing technology which is fully open across machines and operating systems.

APMS offer an alternative technology to existing workflow systems. Most importantly, they also offer an alternative vision of how organisations can be structured and managed. APMS take a 'service-oriented' view of business process management, where the resourcing and co-ordination of activities to support an end-to-end business process involves negotiation and collaboration between customer and provider agents.

## 2. APMS: A Service-Oriented Approach

A business process consists of a number of related activities which collectively realise some market need or business objective,[6]. A business process typically comprises sub-processes and tasks. Where a *task* is an activity which performs one logical step in a business process, this could be a manual or automated activity. Dependencies exist between tasks, so they have to be executed in a controlled and ordered way.

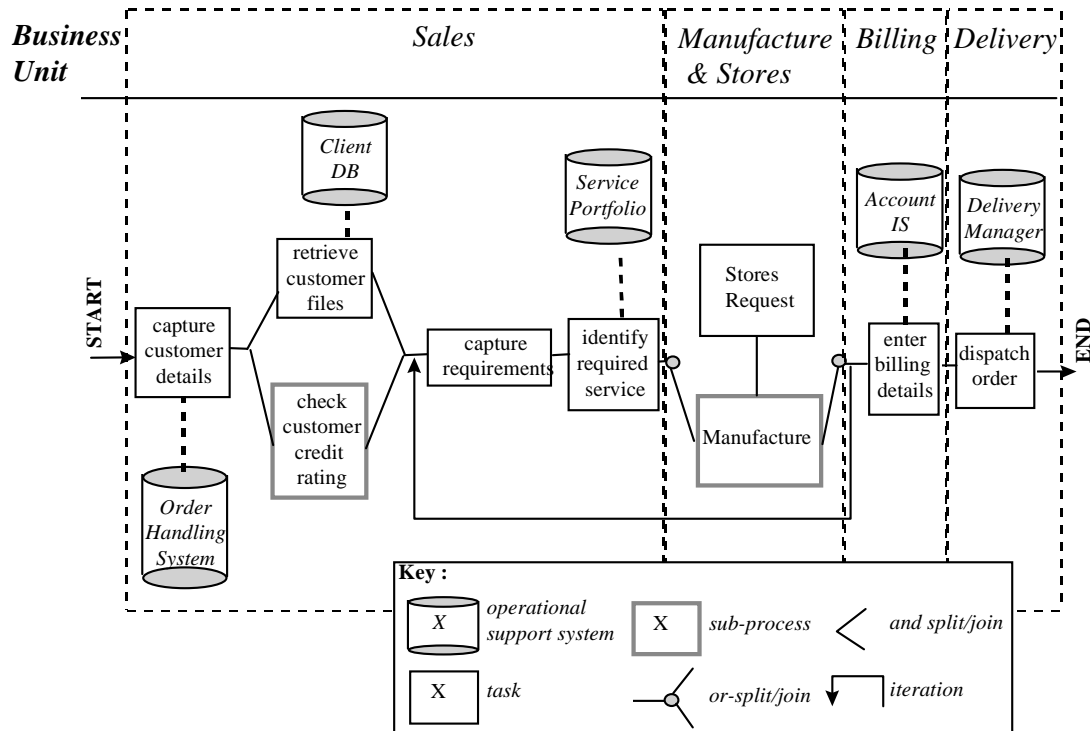


Figure 2 : A Sales Business Process

An example of a business process would be a sales process consisting of those tasks which result in the sale of a product or service. In the example given in Figure 2, the key features of a typical business process are given. There is :

- Hierarchy - as well as process tasks there are sub-processes which can again be reduced into component tasks
- Concurrency - both and-parallel and or-parallel operators are illustrated
- Choice points - at the or-parallel operator the business process can fork off in one of two directions.
- Iteration - if more than one service is being sought part of the process can be repeated.
- Order - throughout there is a notion of order in the execution of tasks and sub-processes.

Underlying this business process are the resources which are consumed during the execution of these tasks. This includes the operational support systems, databases and applications which support tasks and the people who operate these systems. These resources are grouped into business units that have semi-autonomous control over the way in which the resources are deployed.

In the APMS architecture, autonomous software agents represent the interests of these business units and communicate with each other over a communications network, negotiating over how they can collaborate to support an overall business process. Each agent offers services to other agents and can take the role of provider (server) or customer (client) for a service. This provider-customer view of a business process has parallels with Winograd and Flores ActionWorkflow methodology [16,17] for representing business processes although in an APMS this is also reflected in the underlying system architecture, with agents performing the roles of provider and customer of services. There are also parallels with some aspects of the organisation of large decentralised enterprises, where service-level

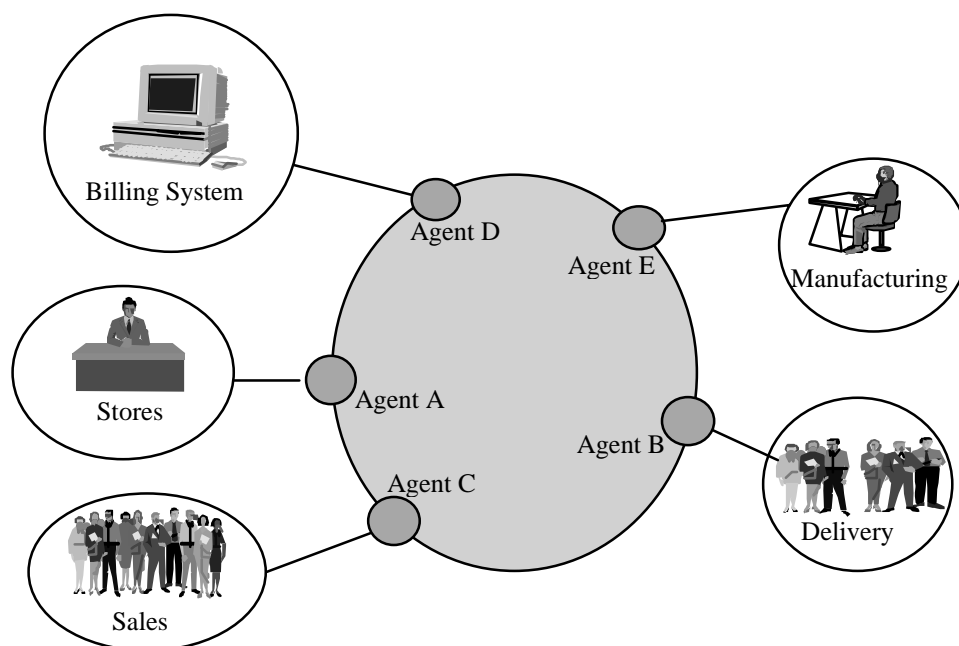


Figure 3: Agents and the Virtual Enterprise

agreements may exist between groups and departments.

A service is a packaging of tasks and other (sub-)services that allows an agent to offer or to receive from another agent, some functional operation. A service can be re-used as a component of another service, whereas a task represents a primitive functional component of a business process.

In Figure 3 we depict an example of five agents managing a business process such as that presented in Figure 2. Each agent provides service(s) in support of a typical customer sales business process.

Agent C representing the sales department would negotiate for services from other agents to support its sales function. This could involve negotiating for manufacturing capacity with agent E, and delivery services from agent B. These agents in turn might require other services, for example the agent E might negotiate for services from stores through agent A. As agreements are reached between the agents (as both client and server agents) contracts are established.

Each agent has to ensure that it has sufficient capacity to provide a service before it commits itself to delivering that service. This requires the agent allocating sufficient resources to support those tasks under its direct control, as well as ensuring it has access to sufficient component services offered by other agents.

An agent may need to guarantee the existence and availability of a particular service with some fundamental level of assurance and reliability. There can be three basic levels:-

- One-off: This allows a single execution of the service. The time of activation will be agreed during negotiation.

- Regular (or scheduled): This allows multiple executions of the service, at agreed and pre-scheduled times.
- On-demand: This allows execution of the service at any time to a specified volume, at any time within an agreed time window.

When a customer requests a service from the sales team (Agent C in Figure 3), an instance of the business process is enacted. The agents manage the execution of the business process instance, ensuring that each task and service is performed in a timely and efficient way within the constraints of the agreed contracts. Agents interact to co-ordinate enactment, facilitate the exchange of information and handle exceptions; this might require rescheduling and/or re-negotiation.

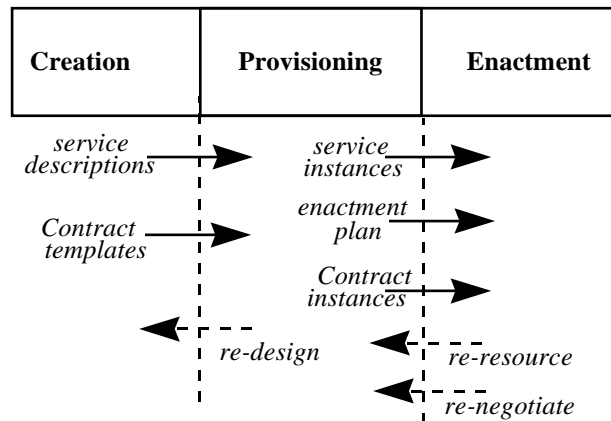


Figure 4: The APMS Lifecycle

Returning to the business process life-cycle model described earlier (see Figure 4), an APMS increases the automation of the management of a business process. The service creation phase involves the definition of service descriptions, contract templates and the selection of negotiating strategies for the agents in the APMS. The service provisioning phase is performed automatically by agents, involving agent negotiation for services, the agreeing of contracts between agents and the assignment of resources. This stage results in specific contracts covering the delivery of services agreed by agents, enactment plans outlining the allocation of resources to tasks, and service instances detailing how services are constituted. The enactment phase is the delivery of those agreed services. If exceptions occur which cannot be resolved by the enactment module, then the service reverts to the provisioning phase for re-resourcing or re-negotiation.

The service-oriented approach of APMS can reflect the inherent distributed nature of large organisations and make the management of an organisation transparent to its logical or physical structuring. Similarly, this approach allows an organisation to adapt and evolve with minimal disruption so that new services or tasks can be defined incrementally, without the need to re-design an entire distributed system. This empowers local semi-autonomous groups to define how they will perform and manage tasks and processes; see [2] for detailed discussions on the empowerment of personnel in large organisations.

### 3. The APMS Reference Model

The APMS reference model comprises three main components: the agent, the business task, the APMS business monitoring and engineering (BME) system, and the interfaces between them. Agents exchange information during negotiation and service enactment via interface A. Agents manage business process tasks, information and resources via interface B. The development and in-service administration and maintenance of agents is performed by the BME system which accesses agents via interface C. During enactment, information can be exchanged between business tasks via interface D. These interfaces are depicted in Figure 5.

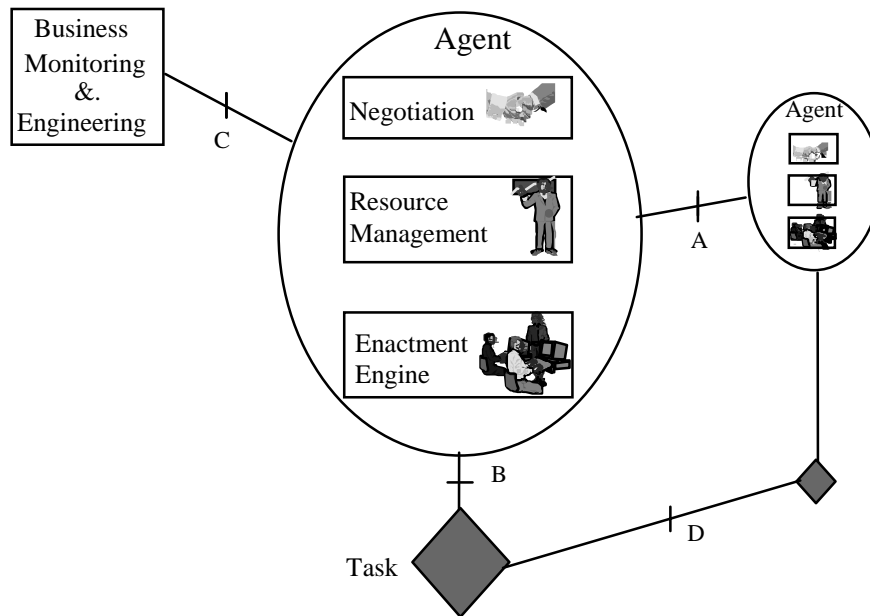


Figure 5: APMS Reference Model

An APMS agent typically [12]3,12,18] comprises three core modules the negotiation module, resource management module, and enactment module. Four interfaces exist, (see figure 5).

Interface A represents the agent interaction interface. This interface supports agent negotiation and information exchange. A standard agent communication language could be used such as FIPA97, [19].

Interface B supports task management. This is an agent-task communication interface that allows an agent to monitor and control tasks that are under its direct management. This could be either a manual or automated business process activity, sometimes known as a workflow enabled task. Typically this would consist of a proprietary interface to a legacy system.

Interface C is the agent management interface, which supports monitoring and on-line engineering of distributed agents. This facility can provide a single view of a business process being managed by multiple agents.

Interface D is the task operational interface, which supports the exchange of information between business process tasks.

### Negotiation Module

Negotiation is the process whereby two agents seek a mutual agreement and commitment on the delivery of a service. Quality, time and cost are three typical parameters which would form the basis for negotiation. As is the case with all negotiations, agreement cannot always be reached.

A contract is the result of an agreement between client and server agents during the negotiation process. The contract contains a list of agreed values for parameters, establishing the terms and conditions for the delivery of the service. For example, the agreed time(s) at which the service will be available and/or activated, the maximum duration of the activation, the minimum quality of the service etc. will be defined.

In order for agents to be able to communicate effectively about these different parameters a common negotiation information model and language are required. The language consists of agreed primitives, content language and protocol to allow agents to suggest modifications to, and consent to, the value of parameters in a contract. The negotiation language protocol defines the valid ordering for sending and receipt of primitives during negotiation between agents. FIPA97 [19] has specified a standard agent communication language and set of protocols to support open agent interoperability.

In a realistic business process an agent will be required to negotiate for multiple contracts simultaneously. Therefore the negotiation management module of an agent must be multi-threaded so that it can support multiple negotiations for different services concurrently.

A number of services can be combined to form a single service. The agent providing the service is designated the controlling agent which negotiates for component sub-services. This grouping of services in support of a single service is termed a 'virtual agency'. A virtual agency is a structure which supports the service management function of an agent. It is a collection of tasks and sub-services involved in the provision of some service by a controlling agent. The controlling agent will have (or require) contracts for the sub-services that it requires from other participating agents. The advantage of virtual agencies is that services can be provisioned automatically in real-time by pooling resources.

During negotiation, an agent correlates and balances multiple criteria both within the negotiation for a single service, and across all the negotiations the agent is involved in. The criteria can be modelled as (partial) ordinal value spaces that represent parameters such as quality, time, cost, etc.

### **Resource Management Module**

An important aspect of an APMS is its ability to perform direct management of resources : the systems, databases, equipment and people that make up an organisation. Resource management is one of the key advantages of the APMS approach.

When a set of resources is under some form of semi-autonomous common ownership, there is usually a requirement to exercise some control over the commitment of those resources, in order to maximise speed, efficiency, etc. and to minimise cost, waste, etc. Delegating a single agent some form of executive responsibility for the set of resources is a way of achieving this. Resource management functionality (in particular scheduling) is either implemented in, or available to, the agent.

One way in which resources may be managed in an APMS is within an 'agency'. An agency represents a grouping of resources under the ownership of a single agent. An agency is the unit for which an agent performs resource management. The agency itself may be physically distributed, but it is logically centralised and managed by a controlling agent. The collection of tasks and (sub-)agents in an agency does not necessarily reflect a functional grouping (though in most organisations this effect may be expected).

During provisioning, an agent can commit to a contract without performing detailed resource allocation. The negotiation management module of an agent can commit that agent (as server) to provide a service. In this way, an agent de-couples negotiation from detailed resource management in order to support real-time performance during the negotiation process. However, full resource scheduling may be performed before the service is required to be enacted.

### **Enactment Module**

Enactment involves the activation of tasks and agreed (sub)services in a way that meets the obligations established in a contract. A server agent activates its tasks and (sub)services when it is triggered by a client agent. This activation can involve the execution of software, or the sending of a work schedule by fax to an operative, for example. An agent can execute multiple services and tasks simultaneously, and the enactment module must be multi-threaded to allow the agent to activate concurrent operations.

When tasks and services fail, agents can perform corrective actions and try to resolve the failure. The enactment module receives exceptions from the tasks within its agency and from other agents (as servers). Exceptions can be resolved by either:-

- Re-starting the task/service from the enactment module.
- Re-scheduling the task/service in the resource management module.
- Re-resourcing the task/service in the resource management module.
- Re-negotiating the terms of the contract (as the defaulting server agent) in the negotiation module.
- Re-locating the service with another agent (as the aggrieved client agent) in the negotiation module.
- Ignoring the exception and accepting a penalty (if appropriate).



## **Business Monitoring and Engineering (BME)**

The APMS BME system supports developers of organisations to engineer and monitor agent-based business process management systems.

The APMS system languages and interfaces are provided to allow developers to implement and maintain the distributed business processes that have been conceived for the organisation. The design of an agent-oriented business process management system involves the principled transformation of some description of that business process into a number of communicating and co-operating software agents and the services they provide. The structure of an integrated system should reflect the structure of the existing organisation(s) rather than impose structure on them.

Each agent manages business functions that are under its direct control. The realisation of an end-to-end business process could involve contributions from many different agents. The monitoring and administration of a business process based on the APMS approach demands the ability to collate information from all the agents involved, filter this information, and present it in a way that allows a business process owner/manager to understand the contribution from all agents in the enterprise, and to see where problems might occur. The BME has to strike a balance between agent autonomy and the overall business requirement.

## **4. Realising Agent-Based Process Management Systems**

The following two sections describe aspects of the development of an APMS. Section 4.1 describes the ADEPT approach providing an example of an implementation of an APMS, its functionality and how it has been developed. Section 4.2 describes the BeaT study of how APMS can be integrated into actual systems which support an end-to-end business process.

### **4.1. ADEPT: An APMS Prototype**

Project ADEPT has developed a prototype APMS which demonstrates multiple autonomous agents managing nearly one hundred business tasks which make up a typical business process. The ADEPT infrastructure consists of a community of agents that can negotiate concurrently with one another in order to reach agreement on how resources are to be assigned to support a business process. The ADEPT agents are used for resourcing business processes, co-ordinating process tasks and exception handling when business processes break down. Further details of the design of the ADEPT demonstrator system are available in [3,14] ; implementation issues are discussed in [18].

The ADEPT agent consists of three core modules : the Interface Management Module (IMM), Situation Assessment Module (SAM) and Service Execution Module (SEM); these map respectively into the reference model outlined in section 3. The IMM manages the negotiation for services with other agents during the service provisioning stage. The SAM ensures that commitments made by the IMM can be discharged using available resources; the SAM maintains a schedule of how and when available resources are to be used. The SEM manages the execution of tasks and services as the business process is enacted.

Resource management is achieved by encapsulating resources into tasks and managing the assignment of tasks to support a business process. Therefore, in the ADEPT system resources are managed implicitly via the assignment to task interfaces. Task resolution, the composition of process tasks from resources is not supported in the ADEPT system.

The management of information passing between process tasks is performed by the ADEPT agent. The ADEPT agent mediates in the exchange of all information between tasks and where appropriate will perform ontology mappings where different information models exist. Therefore interface D in figure 5 is not supported by the ADEPT architecture. This approach could incur problems where significant amounts of information are passed between process tasks, particularly if that information is in the form of a video or audio stream.

### **4.2. BeaT: Engineering APMS Systems in BT**

Project BeaT has studied the linking of APMS to business tasks within a large organisation like BT and has considered the engineering aspects of an APMS.

In a real application, the tasks that APMS manage will be based on systems that have some direct access to, or effect upon, the real resources in an organisation. These resources may be information systems, databases, applications, even people. There are three main approaches to interfacing APMS with business tasks [18] as depicted in figure 6:

- Direct access to databases: If direct access to databases is available, a wide variety of products are available that can provide object-oriented access to this data. If the data is in an object-oriented database (OODB) then object adapters may be available in a CORBA Object Request Broker (ORB) that permit the database objects to integrate seamlessly with the implementation objects, [20,21].
- Application program interface (API): More usually perhaps, the databases would be protected by an application layer that is designed to help keep them in a consistent state. If an application program interface (API) is provided, it should be possible for a task object 'wrapper' to communicate with the application layer and gain direct access to the application functions.
- Screen-scraping: There are a number of possible reasons why no API may be available. The task may be one that is performed manually; in this case the task object needs a way of messaging the person or people involved. The task may be based upon a system that was never designed to operate in an open, distributed computing environment (so-called legacy systems). In this case, one option may be to access and manipulate directly from the screen buffer that was intended originally for a human user - this is sometimes referred to as 'screen-scraping'. Performance issues and the maintenance of the connection are major considerations with this approach. The back-end system may not have been designed to cope with the loading imposed by multiple automated task clients, and this may even cause the system to fail in its original objective.

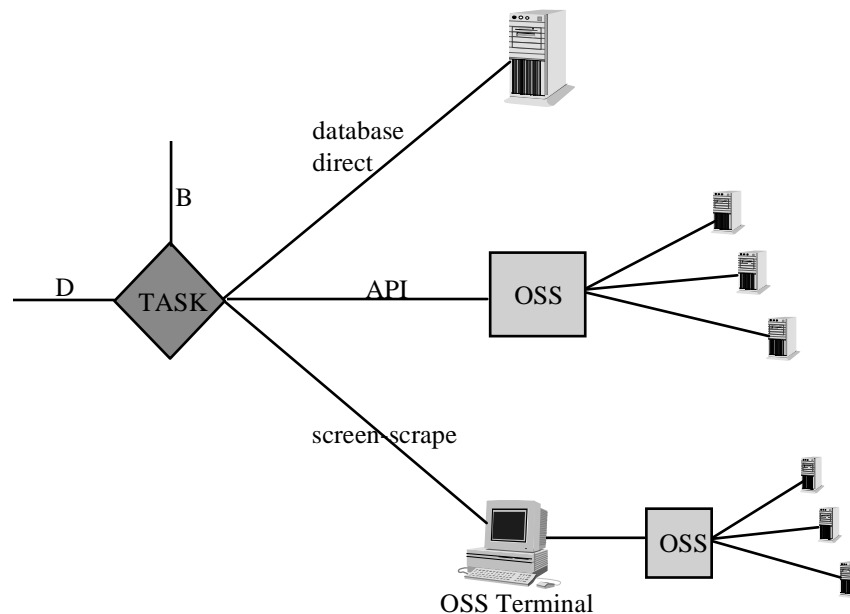


Figure 6: Different Ways to Access Information Systems

A major issue for organisations is that significant parts of a business process are 'locked-up' within large-scale Operations Support Systems (OSS). This can impede improvements to business processes, making the company slow to react to change. The application of workflow management systems to middleware has led the way in separating business logic from the underlying information systems. This approach provides a suitable foundation for the introduction of APMS into an enterprise.

### Distributed Computing Platforms for APMS

APMS rely upon a robust computing platform for accessing distributed functionality throughout an organisation. Such technology that allows agent software to access distributed resources in an organisation is developing rapidly.

- CORBA: A number of workflow management systems as well as project ADEPT have used products based upon the CORBA standard. This standard allows the distribution of software across heterogeneous computing platforms. In APMS this technology can be used to support agent interaction, as well as access to distributed tasks and resources in an organisation, [14,20].
- NextSTEP: Adopts the CORBA approach, but incorporates it into a complete development environment for system designers, and an associated user-interface presentation environment.
- Internet: The internet has already been identified as a low-cost option for distributing workflow management systems, [4,23]. This is expected to make access to distributed resources even easier, though there may be some concerns over performance.

## 5. Future Work

Three areas of research can be identified as key topics for further work.

- Agent Standards : The highly interactive nature of multi-agent systems such as APMS points to the need for the standardisation of agent interfaces in order to support interoperability between agents. FIPA (Foundation for Intelligent and Physical Agents), a recently established agent standards consortium is specifying open agent standards. The FIPA97 specification [19] issued in October 1997 covers :
  - Agent Management : facilities for agent naming, location, communication
  - Agent Communication : a specification of an agent communication language
  - Agent-Software Integration (i.e. facilities for supporting the interoperation between agents and back-end software systems

Similarly, the workflow management coalition (WfMC) are standardising the key WMS interfaces, [16].

The completion and adoption of such standards would be a prerequisite to supporting open interoperability between APMS and would represent an important landmark in the commercialisation of APMS technology.

- APMS Migration Path : Significant costs are incurred in maintaining and updating computing infrastructures in organisations. Existing WMS represent a significant investment and have already addressed issues related to accessing backend systems. Introducing APMS as a replacement is sometimes not a viable option. An alternative is enhancing existing WMS with the value-added aspects of APMS. This allows existing WMS to provide an enactment service and the APMS layer to enhance this with the negotiation and resource management capabilities. Agent Enhanced Workflow is a current area of research at BT laboratories.
- Security : A key factor in making APMS acceptable to the commercial world is ensuring that APMS provide a secure means of managing business processes. APMS require the placing of trust in software agents when performing routine negotiation. Guaranteeing the reliability and security of this technology is key to making it acceptable to industry. Security issues related to agent technology include : authentication (verification of agent identities), confidentiality (restricted access to organisation business functions and data), integrity checking ( ensuring information, data and agent integrity), non-repudiation (security of contract), and secure agent mobility. FIPA have placed security management in its 1998 programme of work.

## 6. Conclusion

Agent-based process management systems have a number of key advantages over existing workflow systems. They automate the provisioning of business processes as well as support business process enactment. The APMS architecture reflects the inherent distribution of responsibility in large organisations, and makes the management of an organisation transparent to the logical or physical structuring of its components. Allowing software agents to represent the interests of autonomous departments or business units allows organisations to adapt and evolve with minimal disruption. New services or tasks can be defined incrementally, without the need to re-design an entire distributed

system. APMS also support the de-centralisation of control in an organisation, empowering local (in a logical or physical sense) autonomous groups to define how they will perform tasks and processes

Research and development of APMS is continuing within the Intelligent Business Systems Research Group at BT Labs. Project IBS (Intelligent Business Systems) is integrating leading-edge agent and scheduling technology in the development of an APMS. Work is continuing also on the application of this technology to the enhancement of existing workflow management systems within BT.

### **Acknowledgement**

ADEPT was a collaborative project under the DTI/EPSRC Intelligent Systems Integration Programme (ISIP). The project partners were BT Laboratories, ICI Engineering Technology, Loughborough University, and Queen Mary and Westfield College (University of London). The opinions expressed in this paper are those of the authors and not necessarily those of BT.

### **References**

- [1] Nelson D., "Frederick W. Taylor and the Rise of Scientific Management", Madison London, 1980.
- [2] Armson R and Paton R, Organisations: Cases, Issues, Concepts (2<sup>nd</sup> edition), Paul Chapman Publishing Ltd, 1994.
- [3] Capezio P., "Total Quality Management : Making it Happen in Your Organisations", National Press Publications, 1995.
- [4] "The Horizontal Organisation", Business Week, 20<sup>th</sup> December 1993.
- [5] Georgakoloulos D., Hornick M., "An Overview of Workflow Management : From Process Modelling to Workflow Automation Infrastructure", Distributed and Parallel Databases, Volume 3, pp119-153, 1995.
- [6] Workflow Management Coalition, "Workflow Management Coalition Terminology and Glossary", Doc. No. WFMC-TC-1011, Issue 2.0, June, 1996.
- [7] Sheth A., "Workflow Automation : Applications, Technology and Research", Tutorial Notes from SIGMOD Conference, May 1995.
- [8] C.Mohan, Tutorial : State of the Art in Workflow Management System Products and Research, NATO Advanced Study Institute on Workflow Systems and Interoperability, August , 1997.
- [9] Trammel K, Workflow Without Fear, Byte, April 1996.
- [10] Nwana H, Software Agents: An Overview, Knowledge Engineering Review, Volume 11, No. 3, 1996.
- [11] Wiegand ME and O'Brien PD, ADEPT: An Application Viewpoint, in Proceedings of Intelligent Systems Integration Programme Symposium, Ambleside, UK, 1996.
- [12] Alty JL, Griffiths DG, Jennings NR, Mamdani E, Struthers A and Wiegand ME, ADEPT-Advanced Decision Environment for Process Tasks: Overview and Architecture, Proceedings of the BCS Expert Systems 94 Conference, Cambridge, UK, pp. 359-371, 1994.
- [13] Jennings NR, Faratin P, Johnson MJ, O'Brien PD and Wiegand ME, Using Intelligent Agents to Manage Business Processes, Proceedings of the 1<sup>st</sup> International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK, pp. 345-360, 1996.
- [14] O'Brien PD and Wiegand ME, Agents of Change in Business Process Management, BT Technical Journal, Intelligent Software Systems Special Issue, October 1996.
- [15] Process Interchange Format (PIF), <http://www.aiai.ed.ac.uk/PIF/References>
- [16] Medina-mora R., Winograd T., Flores R., Flores F., "The ActionWorkflow Approach to Workflow Management Technology", The Information Society, Volume 9, pp.391-404, 1993.
- [17] Action Technologies, "Co-ordination Software : Enabling the Horizontal Corporation", Action Technologies, July, 1994.

- [18] Wiegand ME, Building Practical Agent-Based Systems (Tutorial Notes), presented at The 1<sup>st</sup> International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK, pp. 345-360, 1996.
- [19] FIPA97, Parts 1-3 (Normative Standards), October 1997, (<http://drogo.cselt.stet.it/fipa/>)
- [20] “The Common Object Request Broker : Architecture and Specification”, Object Management Group, Revision 2, 1995, (<http://www.org.omg/corba2>)
- [21] Miller J., Sheth A., Kochut K., Wang X., “CORBA-Based Run-Time Architectures for Workflow Management Systems”, University of Georgia, 1995.
- [22] Kador J, The Ultimate Middleware, Byte, pp.79-83, April 1996.
- [23] Kobielus J, Workflow Standards Developers Need New Rules in an Age of Electronic Commerce, Network World, September 1995.
- [24] Jennings N, Faratin P, Johnson M.J, O’Brien P.D, Wiegand M.E, “Using Intelligent Agents to Manage Business Processes”, Proceedings of the 1<sup>st</sup> International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK, 345-360, 1996.