Agents of Change in Business Process Management

Paul O’Brien and Mark Wiegand

0. Abstract

Successful enterprises are built on change. Increasingly, businesses operate in a rapidly evolving environment where the response to changing markets may of necessity be measured in hours and days instead of months and years. Responsiveness and adaptability will be the hallmarks of business success. BT is strategically placed as both a major potential facilitator of this change, as well as benefiting from its technology.

This article describes how agent-based process management systems can provide powerful tools for managing the enterprise of the future. It explores recent work combining distributed computing technology with autonomous software agent techniques for business process management, and argues that these represent a viable supplement to and even an alternative to existing workflow management systems. This is supported by the results of a number of projects, including project ADEPT[1], BeaT[2] and a number of other related projects, which are exploring how leading edge technology can improve the way business processes are managed. This article provides a vision of how agent-based process management systems can support the needs of the “virtual” enterprise of the future and the integration of the information systems of small to medium-sized enterprises (SME).

[1] ADEPT is a three year collaborative project involving BT, ICI Engineering Technology, Loughborough University and Queen Mary & Westfield College. The project is part of the DTI / EPSRC Intelligent Systems Integration Programme (ISIP). This work also draws from a number of other BT projects looking at the development of intelligent business process management systems.

[2] BeaT has investigated the engineering aspects of applying agent-based process management systems to actual BT business processes and the information systems which underpin them.

1.0 Introduction

Existing Workflow Systems

Workflow systems provide powerful automatic mechanisms for managing the execution of business processes. 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. However, workflow management systems have certain limitations which need to be addressed if business process management systems are to be applied more successfully and to more business processes.

Workflow management systems 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 policies, and resource levels can change. Such business processes highlight a number of shortcomings in existing workflow management systems [1]. 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.
- extensibility : many systems are not extensible on-line.
- 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.

Agent-Based Process Management Systems

Agent based process management systems (APMS) combine the latest distributed computing technology with agent-based techniques [2,3] providing an intelligent extension and alternative to workflow management systems. They 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.
- 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 LAN's and WAN's 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.

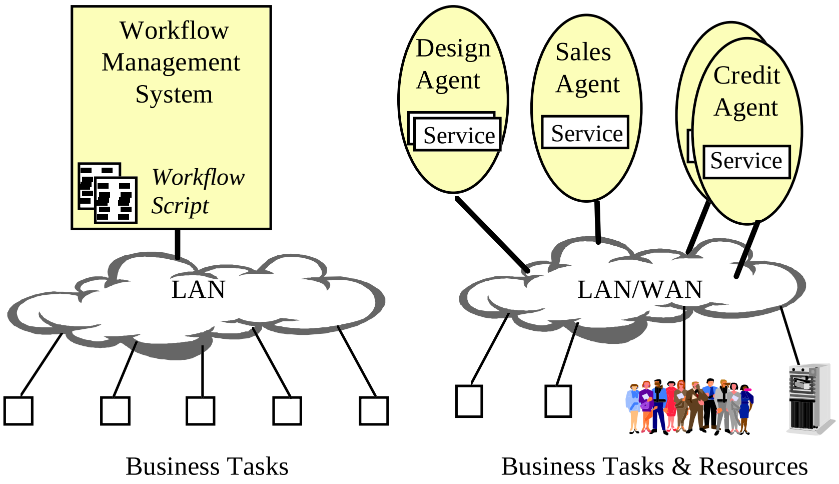

Figure 1 : Workflow Management Systems Compared with APMS

Figure 1 contrasts workflow management systems which have a centralised workflow engine, with APMS which consist of autonomous systems that represent the respective concerns of various business units involved in a business process. 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 [4] for detailed discussions on the empowerment of personnel in large organisations.

An important feature of APMS is that 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 workflow management systems APMS’s have two objectives, firstly, the timely execution of business functions and secondly, the efficient use of resources.

2.0 APMS : A Service-Oriented Approach

A business process is made up of a number of process activities or tasks which are combined to express the functioning of the process. Dependencies exist between tasks, so they have to be executed in a controlled and ordered way. The execution of a task consumes resources which typically are grouped into business units that have semi-autonomous control over the way in which the resources are deployed.

Autonomous software agents communicate with each other over a communications network and negotiate 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 client-server view of a business process has parallels with Winograd and Flores ActionWorkflow methodology [5] although in an APMS this is reflected in the system architecture, with agents performing the roles of provider and consumer of services. There are also parallels with some aspects of the organisation of large enterprises, where service-level agreements may exists between groups and departments.

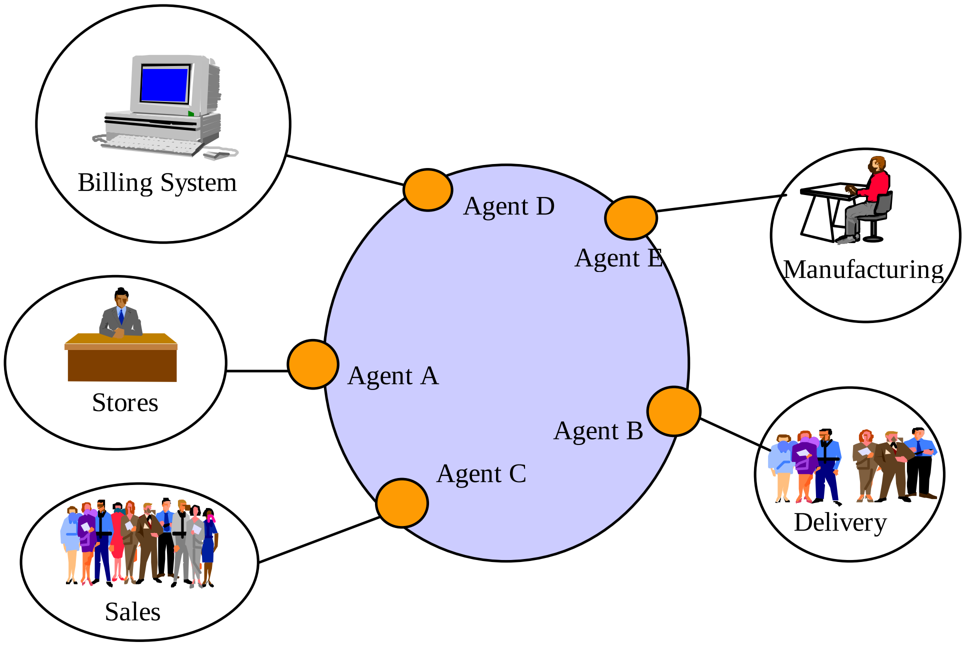

Figure 2 : Agents and the Virtual Enterprise

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. A task represents a primitive functional component of a business process.

In Figure 2, we depict an example of five agents managing a business process. 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 their 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 execution of the service, at agreed and pre-scheduled times.
- on-demand : This allows execution of the service at any time to a specified volume, within an agreed time window.

When a customer requests a service from the sales team, an instance of this business process is enacted. The agents manage the execution of the business process 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.

3.0 The APMS Reference Model

The APMS reference model comprises of 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 3.

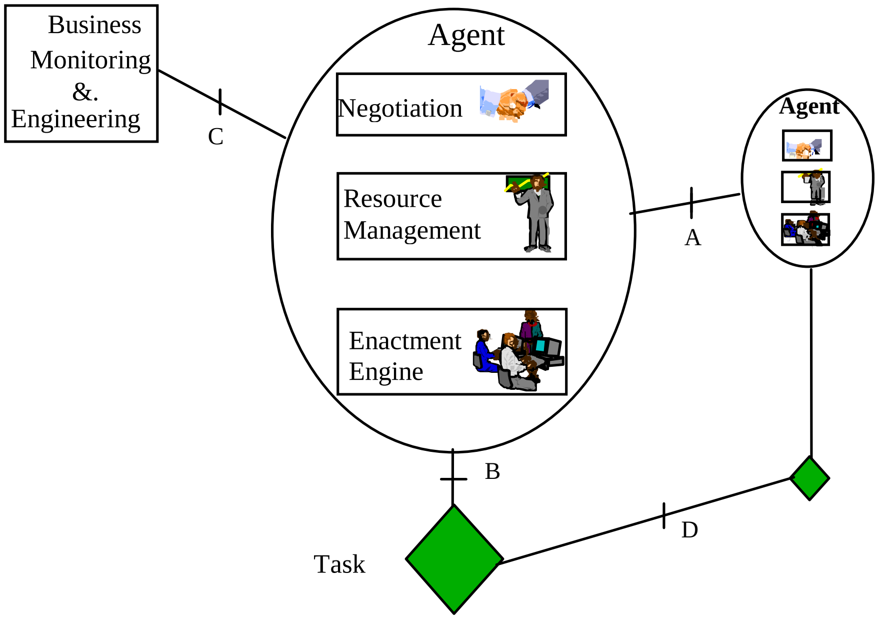

Figure 3 : APMS Reference Model

An APMS agent typically [6] comprises three core modules :

- negotiation module
- resource management module
- enactment module

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 able to communicate effectively about these different parameters a common negotiation information model and language are required. The language consists of agreed primitives and a 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 on sending and receipt of primitives during negotiation between agents.

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.

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. It is the collection of tasks and (sub-)agents that are under the direct control of an agent. An agency reflects the hierarchical grouping of tasks that are under some form of common ownership, and for which there is some reason to attempt to optimise the use of resources in the execution of those tasks. 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.

Enactment Module

Enactment involves the activation of tasks and agreed (sub)services in order to meet the obligations established in a contract. Server agents activate tasks and services when triggered by client agents. This can involve the execution of software or the sending of a work schedule by fax to an operative. An Agent executes multiple services and tasks simultaneously. The agent must be multi-threaded to allow the agent to activate concurrent operations and handle multiple negotiations.

When tasks and services fail, agents can perform corrective actions and try to resolve the failure. This can be achieved by either re-resourcing a task, or re-negotiating for a service. The service/task management module receives exceptions from the tasks within its agency and from other agents (as servers). Exceptions can be resolved by either:-

- re-resourcing the task/service by the resource management module;
- re-negotiating the terms of the contract (as the defaulting server agent);
- re-locate the service with another agent (as the aggrieved client agent);
- ignore the exception and accept 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.

Within APMS, agents offer services to each other based upon process activities (tasks). These services are combined in order to realise business processes. The techniques for defining services that may be useful in such a scenario correspond to business process definition tools, though it is important to note that these services are not expressing what the complete process is. The service definition phase must be linked to a *methodology* that ensures that the defined services are useful to the enterprise (and will therefore be used) and that they will together realise an end-to-end business process. We refer to these conditions as *necessary* and *sufficient* service definitions.

The APMS system languages and interfaces are provided that 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 that 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 autonomous agents and the overall business requirement.

4.0 Realising Agent-Based Process Management Systems

The following two sections describe aspects of the development of 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 [6] has developed a prototype APMS which demonstrates multiple autonomous agents managing nearly one hundred business tasks which make up a typical BT 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 breakdown. Further details of the ADEPT demonstrator system are available in [7,8].

The ADEPT Architecture

The ADEPT agent consists of three core modules : the Interface Management Module (IMM), Situation Assessment Module (SAM) and Service Execution Module (SEM), these respectively map into the reference model outlined in the previous section. 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. In the ADEPT system resources are managed implicitly via the assignment to task interfaces.

The ADEPT Demonstrator

The ADEPT demonstrator prototype (see Figure 4) is a full integration over the Common Object Request Broker Architecture (CORBA)-compliant platform, DAIS from ICL. CORBA supports the distribution of computing objects across different machine and operating-system architectures. The ADEPT demonstrator consists of the following main components :

- a community of nine agents using the ADEPT agent architecture.
- a BT Task Simulator that emulates the relevant behaviour of nearly 100 distinct business tasks.
- an Agent Visualiser, serving part of the monitoring requirements for the APMS BME.

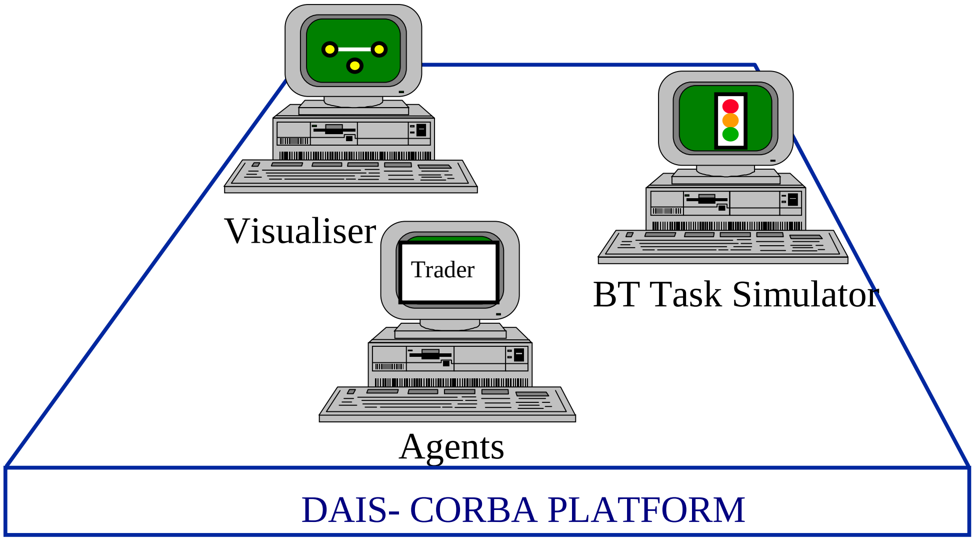

Figure 4 : The ADEPT Prototype

The ADEPT agents are multi-threaded, supporting the concurrent execution of several business processes. Nine agents are used in the prototype to manage nearly 100 business tasks. The CLIPS rule based language (freeware developed by NASA) has been used in the implementation of the agents. Each agent contains a separate CLIPS environment and is registered with the DAIS trader.

The BT Task Simulator emulates the relevant behaviour of distinct business tasks providing a simulation environment for the agents which is used for both testing and demonstration. Tasks may have varying execution times and different probabilities of success or failure. Quintus and Quintus Objects has been used to implement the Task Simulator. Each task object is registered separately on the DAIS trader.

Agent visualisation tools provide the means of displaying agent communication and interaction. The visualiser does not register itself as a server with the trader, rather it contacts agents as a client and passes them its own object reference to allow them to send information back for display. The visualiser can be seen to include functionality that would be included in an APMS BME.

4.2 BeaT : Engineering APMS Systems in BT

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

Accessing Tasks and Back-End Systems

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 [9] :

- 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.

- 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 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.

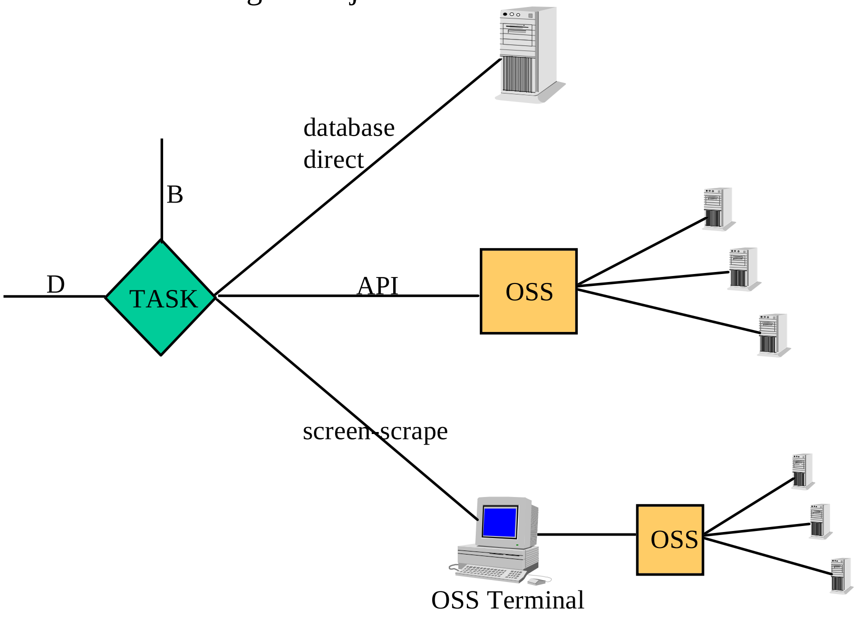

Figure 4 : Different Ways to Access Information Systems

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.
- NextSTEP : Adopts the CORBA approach, but incorporates it into a complete development environment for system designers, and an associated user-interface presentation environment.
- Internet : including Java, which represents a further advance in technology which is expected to make access to distributed resources easier.

5.0 Conclusion

Agent-based process management systems have a number of key advantages over existing workflow systems :

- They reflect the inherent distribution of large organisations.
- They make the management of an organisation transparent to the logical or physical structuring of its components.
- They allow 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.
- They support the de-centralisation of control in an organisation. This empowers local (in a logical or physical sense) autonomous groups to define how they will perform tasks and processes.

6.0 Future Work

Research and development of APMS is continuing within the Intelligent 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.

Acknowledgements

The authors would like to acknowledge the support of the ADEPT consortium, and their colleagues Nader Azarmi, Robin Smith and Hyacinth Nwana. The opinions expressed in this paper are those of the authors and not necessarily those of BT.

References

[1] Trammel K, “Workflow Without Fear”, Byte, April 1996.

[2] Smith R., “Software Agent Technology”, Proceedings of the 1st International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK, 557-551, 1996.

[3] Nwana, H., “Software Agents : An Overview” Knowledge Engineering Review, Vol. 11, No 3, 1996.

[4] Armson R, Paton R, “Oganizations : Cases, Issues, Concepts”, (2nd edition), Paul Chapman Publishing Ltd, 1994.

[5] Medina-Mora R, Winograd T, Flores R, Flores F, “The ActionWorkflow Management Technology”, The Information Society, Volume 9, 391-404, 1992.

[6] Alty J.L., Griffiths D, Jennings N, Mamdani E, Struthers A, Wiegand M, “ADEPT-Advanced Decision Environment for Process Tasks : Overview and Architecture”, Proceedings of the BCS Expert Systems 94 Conference, Cambridge, UK, 359-371, 1994.

[7] Jennings N, Faratin P, Johnson M.J, O’Brien P.D, Wiegand M.E, “Using Intelligent Agents to Manage Business Processes”, Proceedings of the 1st International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK, 345-360, 1996.

[8] Wiegand M.E, O’Brien P.D, “ADEPT : An Application Viewpoint”, in Proceedings of Intelligent Systems Integration Programme Symposium, Ambleside, UK, 1996.

[9] Wiegand M.E, “Building Practical Agent-based Systems” (Tutorial Notes), given at The 1st International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK, 345-360, 1996.