RZ 3144 (# 93190) 06/14/99
Computer Science/Mathematics 7 pages

Research Report

Processes as Customer-Managed Services?

Heiko Ludwig and Andreas Schade

IBM Research Division
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

Research Division
Almaden ● Austin ● Beijing ● Haifa ● T.J. Watson ● Tokyo ● Zurich

Processes as Customer-Managed Services?

Heiko Ludwig and Andreas Schade

IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract - Electronic commerce systems currently focus on and provide sales and payment functionality. Only recently we can see in some domains companies that offer their customers the ability to monitor and control services, which are performed on their behalf, through a network like the Internet. In many cases companies choose Workflow Management Systems (WfMS) to support the service performance. This contribution proposes a framework to relate monitoring and control operations of services to WfMS-enacted processes. This helps organisations to offer customers service management capabilities for their WfMS-enacted services.

Introduction

While there are systems in the area of telecommunications [MFZn97] that allow service customers to monitor and control telecommunication links through a network, we can hardly find any solution that provides means for the customer management of process-oriented services that are enacted using Workflow Management Systems (WfMSs). In contrast to Internet-based advertising and purchasing of services, the management of outsourced services by customers is not well supported by today's systems. One of the reasons is that the relationship between the notions of a service and a process that implements (parts of) this service is not well understood yet.

This paper analyses how processes that are enacted using WfMSs can be managed in the way complex services can be. We proceed in the following way:

- We introduce the notions of managed objects and managed services.
- Based on managed services we introduce a model of a complex, customer-managed service.
- Finally we apply this framework to the customer management of workflow-enacted services.
- We conclude with a discussion of applications of this model.

Model of Managed Objects and Services

The notion of managed objects and managed services emerged in the area of distributed computing and found one of its primary applications in the area of network management, [ITU-T], [Scha99].

Management is a process comprised of two activities: observation and control of a managed system exercised by a managing system. Observation, often also denoted as monitoring, means retrieval and pre-processing of information about the current behaviour of the managed system components. In an event-based monitoring scheme this information is emitted as unsolicited notifications. Based in this information the managing system takes decisions as to how to influence the managed system's activity. Intervention is typically exercised in terms of management operations that modify the behaviour of the corresponding managed components.

A single component of the managed system is called a *managed object*. The precise definition of the term Managed Object depends on the management architecture. In some cases Managed Object are regarded as pure abstractions of controlled resources, in other cases managed objects are management-enabled core objects of the controlled system. In either case, managed objects are equipped with:

- a *management interface* at which the management operations will be invoked and
- special program code, the so-called *instrumentation*, that generates notification messages and that implements the actual management operations.

Aside from its management interface a managed object must still provide the ordinary operational interfaces (also often referred to as primary interfaces) of the encapsulated core object though which the normal object interaction is conducted. A schematic illustration of a managed object is shown in figure 1.

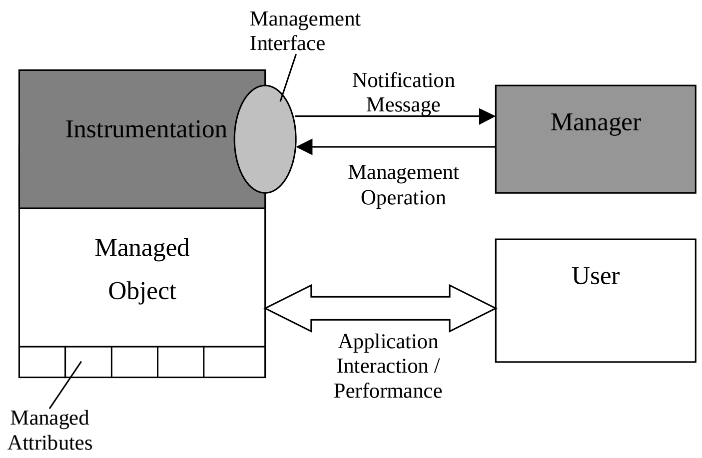

Figure 1: *Managed Object [Scha99]*

Notification messages are sent when an event occurs in the core object, such as a transition of one internal state to another. The content of the event messages depends on the type of the event occurred. Based on these event types event messages can be typed as well by defining their attributes used to convey the notification information.

The management interface provides an abstract representation of the encapsulated managed entity that facilitates the transparent handling of heterogeneous hardware and software components. The definition of the management interface contains:

- management operations modifying the behaviour of the corresponding core object,
- attributes representing core object parameters whose values can be read and changed,
- events message types used to communicate monitoring information (if typed notification messaged are used), and possibly
- management states which can be used to impose constraints on the availability of management operations based on internal states of the core object.

The managed object itself can be complex, consisting of several managed objects that implement the functionality of a *managed service*, as it is called (figure 2).

Externally the managed service provides the same kind of interactions as a managed object, management operations, events and the core performance of the service itself. It extends the concept of a managed object by its internal structure. A service is actually provided by a set of *internal objects*, each of which being a managed object. These internal objects are controlled by the *coordinator*. The coordinator has two roles:

1. *Enactment Control*: It determines how internal objects provide the service's performance by instantiating new objects of particular types and managing them during their lifetime. It acts according to a pre-defined enactment plan.

   If we assume the example of a parcel service the overall service is the transport of a good from sender to receiver. Each leg of the route can be seen as sub-performance. The route is the enactment plan. These internal objects are also instrumented and expose a management interface to the coordinator of the overall service.

2. *External Management Facilitation*: Management operations on the service level have to be interpreted by the coordinator. Therefore the coordinator aggregates the states of the service's internal objects to a (surrogate) overall state of the service. It invokes internal management

operations on the current object instances depending on the state of the internal objects, the overall state and the value of the attributes of the service.

In the case of the parcel transport example, the management operation “return parcel to sender” applied on the service entails different internal management operations, cancel the currently leg and transport the parcel back this route, depending on which legs the parcel was already transported.[1]

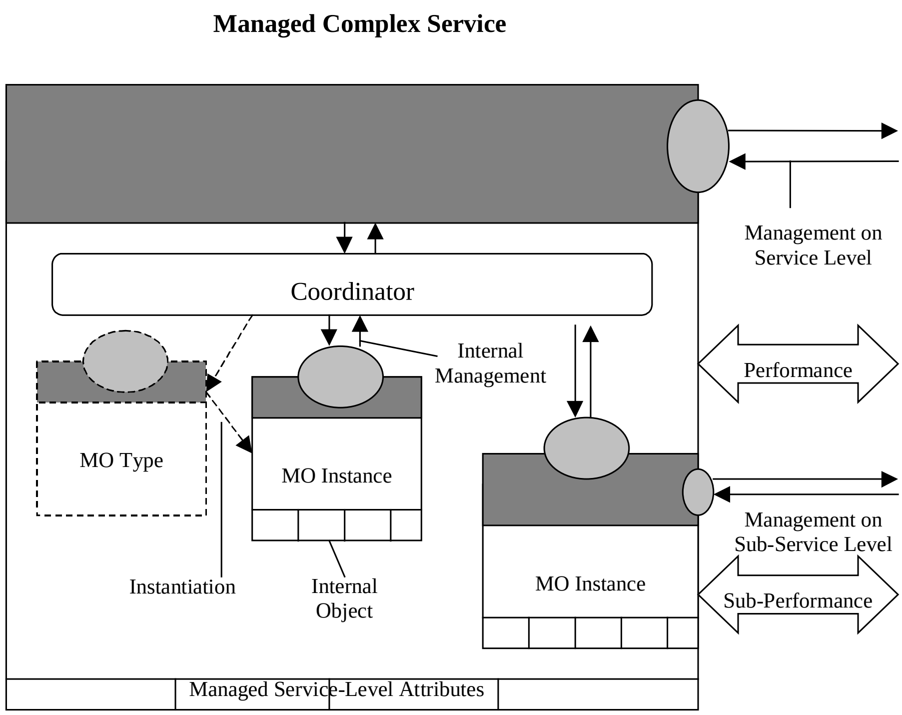

Figure 2: *Managed Service*

In addition to an internal management interface, internal object instances of a service can also expose an external management interface, thereby creating a managed sub-service that is part of the overall managed service for the life-time of this object. The complete management interface of a managed service, the invariant service level interface plus the changing set of sub-service level interfaces, can therefore vary over time. The new management interfaces of the sub-services need to be made known to the service manager as they become available, e.g. using a notification message.

Internal objects can again be complex, being implemented by a coordinator that creates and manages internal objects, and thereby recursively constitute a managed service.

Customer Management of Complex Services

So far we were assuming that manager and managed object are members of the same domain[2], e.g. the same organisation. If an external customer assumes the role of the manager of the service

[1] We ignore the transactional aspect here.

[2] The notion *domain* is discussed for example in [HoCr97]. Many other aspects have to be considered when crossing domain boundaries, such as technical and ontological differences [Ludw99] but are not dealt with in the context of this paper.

further considerations have to be made. As described in [Ludw98a], the involvement of an external customer adds an additional perspective: May the customer perform the management operation at this point in time? Management gateways intercept the interaction between customer and managed service thereby providing access and emission control (figure 3).

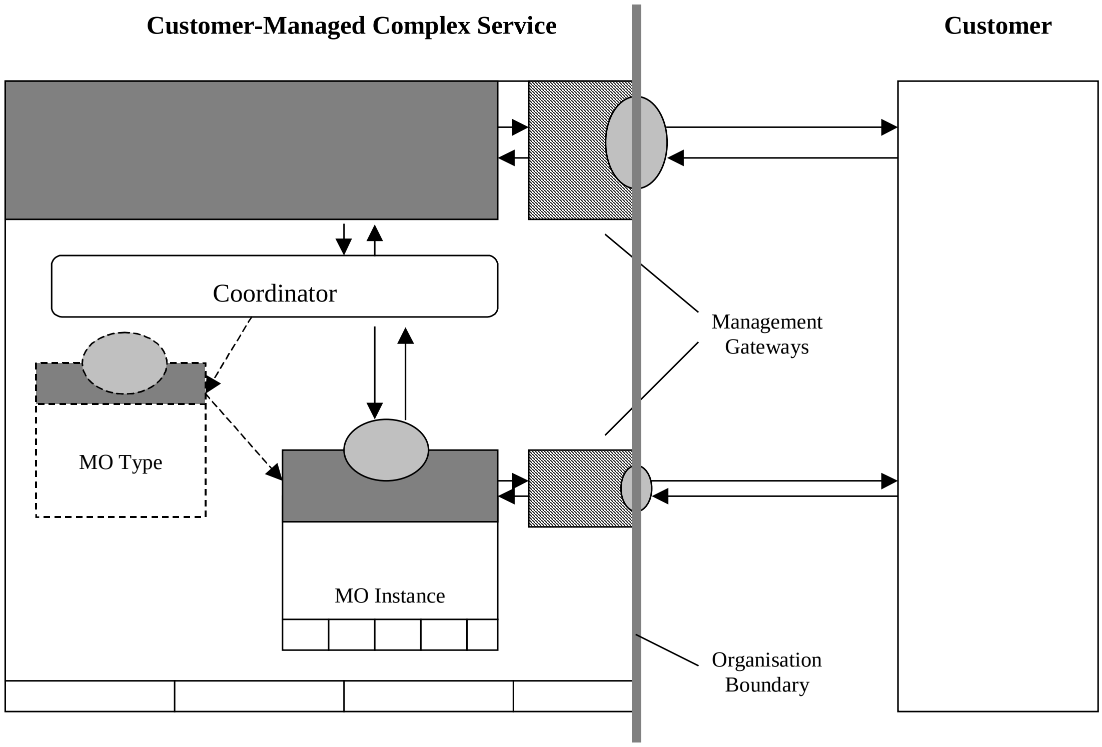

Figure 3: *Customer Service Management (Performance aspect not represented here)*

Access control means deciding whether a management operation that is submitted by the customer is going to be performed at the current state of the service. To take the access control decision, a management gateway can evaluate to the state of the service attributes, the overall service's state of the coordinator and the state of internal objects to conclude its decision.

Emission control decides whether a notification message that is created by the instrumentation of the service is sent to the consumer. The decision also depends on the internal state of the service.

The gateway decides according to a *management interaction policy*. This policy might be agreed on with the customer in some form of contract. The issue how to get the management interaction policy is outside the scope of this text.

Management Gateways have to be provided for all management interactions with the service, on the level of the overall service as well as for each of the sub-services that become available during the enactment time of the service.

Workflow Management Systems as Part of a Managed Service

Which role could WfMS take in the model of a managed service? WfMSs support the enaction of processes on the basis of pre-defined process schemas – taking a production workflow point of view [LeRo97]. Process schemas define the activities that the process consists of (the WHAT aspect of the process), in which order the activities have to be performed (WHEN) and to which performer, employee of application, they are assigned to (WHO). Process instances are created according to process schemas. The WfMS enacts a process instance by instantiating its activities, evaluating when they have to be executed and assigning them to their respective performers. From the point of view of the managed service model, a WfMS enacting a process instance can be seen as the coordinator of the activities of which the process instance consists (figure 4).

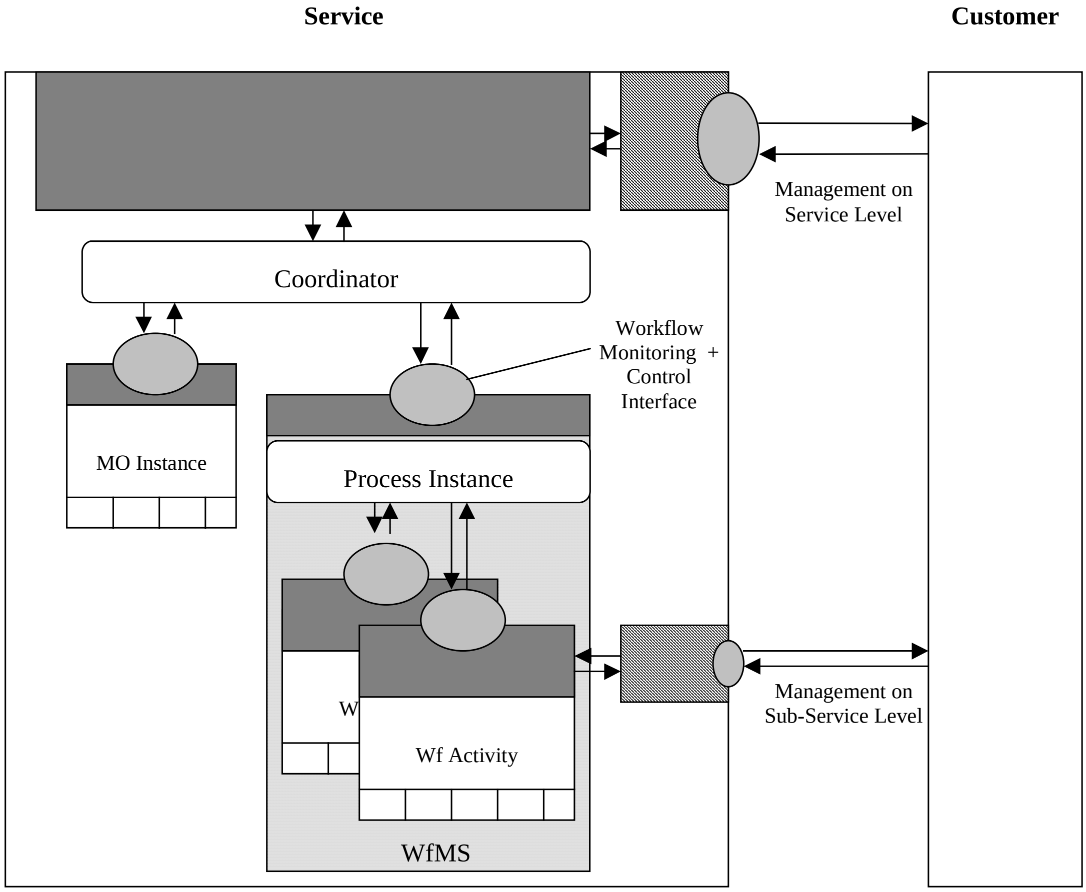

Figure 4: *Customer Service Management of WfMS-enacted services (Performance aspect not represented here)*

Figure 4 shows a scenario in which a service is partially performed by instances of “regular” managed objects, directly managed by a coordinator as described in the previous sections, and also by a process instance being enacted using a WfMS. The enactment of the process instance according to its process schema is represented as a coordinator within the WfMS. It creates and manages the Workflow activities (Wf Activities). Wf activities can be seen as objects that are managed by the process coordinator. The WfMS exposes a management interface (Workflow Monitoring + Control Interface) to its environment that is used by the service’s coordinator to manage the process coordinator. From the point of view of the main service coordinator the process instance in the WfMS is a managed object with a specific interface.

If the customer should be able to manage a Wf activity as sub-service, a management gateway has to be created for it, a management interface has to be exposed through this gateway and the gateway has to be linked to its instrumentation.

Current WfMS like IBM’s MQ Series Workflow provide management interfaces for process instances and Wf activities, e.g. in the form of an API. The sophistication of these interfaces determines which management operations can be offered to customers at the management gateways.

Usually, the management interface of a WfMS comprises operations to retrieve and set data that is related to a process instance or activity and to retrieve and modify a surrogate state according to a state model. (See for example the Workflow Management Coalition's meta-model [WfMC95].) These interfaces can be used to provide general management access for a service's coordinator as well as to link a management gateway for a customer to a particular Wf activity. WfMS that work according to more sophisticated process meta-models, e.g. that provide transactional capabilities, might offer advanced management operations.

Notification messages are not provided in general by all WfMS's management interfaces. If notification messages should be provided the coordinator could periodically retrieve state information (polling) and make them available to the instrumentation, which can send a notification message through the management gateway.

Conclusion

WfMS-enacted processes can be integrated in the model of a customer-managed service and most current WfMS can be embedded into systems that provide customer management. Current WfMSs provide the interfaces that are required to provide management access for the service's coordinator as well as for management gateways to make individual activities of a process instance manageable as sub-processes.

To leverage this approach either WfMSs have to be extended by service management facilities or existing service management platforms, e.g. in the telecom arena, have to be extended to incorporate WfMS-enacted processes. Using this approach, companies can provide customers with Internet-based control of processes that are performed on the customers' behalf.

Acknowledgements

The work presented in this paper was partly conducted in the context of the CrossFlow project that is partially funded by the Commission of the European Union and the Swiss Government as ESPRIT project No. 28635.

References

[HoCr97] Y. Hoffner and B. Crawford: Using Interception to Create Domains in Distributed Systems. In*: Proceedings of the Joint International Conference on Open Distributed Processing (ICODP) and Distributed Platforms (ICDP).* Toronto, 1997.

[ITU-T] ITU-T: *Recommendation M.3010. Principles for a Telecommunications Management Network.* ITU, Geneva, 1996.

[LeRo97] F. Leymann and D. Roller: Workflow-Based Application. In: *IBM Systems Journal.* Vol. 36, No 1, 1997.

[Ludw98a] H. Ludwig: Analysis Framework of Complex Service Performance for Electronic Commerce. In A. M. Tjoa, R. R. Wagner (Eds.): *Proceedings of the Ninth International Workshop on Database and Expert Systems Applications (DEXA '98)*, Vienna, August 1998, pages 638 - 643, IEEE Computer Society, Los Alamitos, 1998.

[Ludw98b] H. Ludwig: *Modelling Customer Service Control using the CoCS Approach.* IBM Research Report RZ 3081, 1998.

[LuWh99] H. Ludwig and Keith Whittingham: Virtual Enterprise Co-ordinator – Agreement-Driven Gateways for Cross-Organisational Workflow Management. In D. Georgakopoulos, W. Prinz, A. L. Wolf (eds.): *Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration (WACC '99).* San Francisco, 1999, ACM Press New York 1999.

[MFZn97] J.-P. Martin-Flatin and S. Znaty*: Annotated Typology of Distributed Network Management Paradigms*. Technical Report SSC/1997/008, École Polytechnique Fédérale de Lausanne, Lausanne 1997.

[Schade99] A. Schade: *Management verteilter Anwendungen auf der Grundlage objektorientierter Interaktionsplattformen*. Shaker Verlag, Aachen, 1999.

[WfMC95] Workflow Management Coalition: *Reference Model*. WFMC-TC-1003, 1995.