Cross-Organizational Service Evolution Management

Odorico von Susani
Department of Information Technology (CUI), University of Geneva, Switzerland
odo@worldcom.ch

Philippe Dugerdil
Department of Information Systems, Haute Ecole de Gestion, Geneva, Switzerland
philippe.dugerdil@hesge.ch

Abstract

Many organizations have started to outsource non-core information processing tasks to service providers. By doing this, organizations abandon their authority on the life cycle of their processes and depend instead on external services. A change in a service's semantics or interface implies that the organizations consuming that service must adapt their dependent applications. To mitigate the risks related to the availability of a service version, organizations establish formal contracts with their service providers. Since service providers are un-coordinated, there is a risk that many services consumed by an organization evolve simultaneously. The adaptation work that is generated by these simultaneous evolutions can be greater than what an organization can handle within the contractual time frames of the evolving services. In this paper, we detail a model and a range of indicators that permit to quantify the capacity of an organization to manage the evolution of the external services it depends on.

Key Words : Maintenance, outsourcing, versioning, e-business

1. Introduction

The emergence of the Internet has been the starting point of the electronic services outsourcing trend. With the Internet as an open digital communication space, more and more companies delegate information processing related tasks to external contractors. This situation is driven by economic and strategic considerations. There is widespread belief that any non-core business process that does not carry a certain intrinsic amount of added value should be outsourced if there is a provider for that process as a service. This is also true for information processes. There has been a noticeable shift towards offshore outsourcing with the emergence of India an China in the IT services industry ([1], [2]). This trend is driven by the idea that a service contractor who specializes in a domain will have particular in-house skills for these domain related tasks and will therefore be able to handle these types of outsourced processes more efficiently than the delegating organization [3]. Additionally, it is also thought that because the contractor will handle this information process for a wide number of delegating organizations, it may be possible for the service provider to decrease the marginal cost of one instance of the outsourced process through the multiple reuse of the infrastructure necessary to handle this process. Thus, the service provider should be able to offer a process as a service at a lower *total cost*[1] than the cost that would be endured by the delegating organization if it were performing the outsourced process in-house.

This context of information process outsourcing has highlighted some technical aspects that were not in the spotlight when all processes were executed within an organization. The most directly noticeable ones include the availability of the outsourced service, the securization of the data transport and the confidentiality of the processed data which are associated to outsourcing risk factors ([4], [5]). Aside these forefront aspects, little attention has been paid to the evolution problems of these outsourced processes. These problems are related to the *authority loss* suffered by an organization when outsourcing its processes.

When a process is outsourced, the delegating organization transfers to the service provider not only the responsibility of correctly executing the process, but also the underlying execution environment, i.e. all the hardware and software that are required for running that process. The service provider is then also responsible for all security aspects surrounding the process execution which include cross-organizational data communication and service access restriction. Furthermore, the service provider receives full control on process availability, implementation and evolution. This means that the service provider has the authority on the outsourced process's lifecyle.

As the owner of this authority, a service provider is entitled to decide arbitrarily many of a process's aspects, including whether a process shall even be available as a service, and if or when some of the process's semantics or technical interfaces shall be modified. As shown in Figure 1, an outsourced process can be viewed as a *sovereign system*[2]

[1] The *total cost* comprises the price of the service execution by the contractor and the internal management costs of the outsourcing organization for the delegated process.

[2] A sovereign system is defined as a system controlled by a single authority, for which decisions like application-level operational policies, platform architecture, object models, authorization policies and

978-0-7695-3596-8/09 $25.00 © 2009 IEEE
DOI 10.1109/ITNG.2009.218

[6] which had its controlling authority changed from the delegating organization to the service provider that is offering the process as a service. For the rest of this paper, an outsourced process is defined as a service offered by a service provider from the delegating organization's point of view.

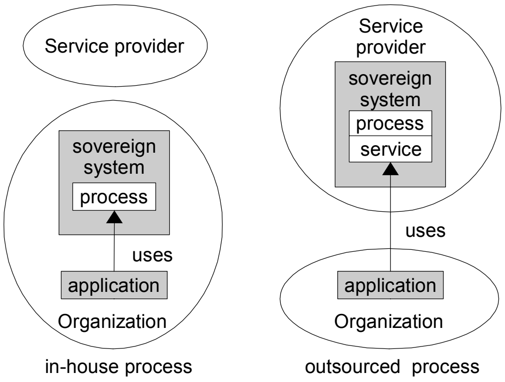

Figure 1. Outsourcing a process as a transfer of authority of a sovereign system

Many organizations need their information processes to be executable on a non-stop basis (24/24, 365/365), no matter if they are implemented in-house or outsourced as services. Since the delegating organizations have no authority on the outsourced processes, they will try to mitigate this availability risk using formal outsourcing contracts ([7], [8], [9]). The availability of a particular service is therefore contractually guaranteed for a defined period of time.

It may be thought that this type of contractual relationship is sufficient to mitigate the risks that are inherent to the outsourcing of an information process. Nevertheless, there are some other aspects that should be considered when outsourcing an information process, i.e. using services from a service provider. Among them, the loss of intrinsic knowledge about a particular process and the adaptation problems due to service evolution appear to be non-negligible. In the case a service provider decides to end the offering of a service, an organization consuming that service must find an equivalent service from another provider. If there is no such other provider for that service, the consuming organization must re-acquire the knowledge related to the process and must also re-implement the process in-house. Depending on the process's complexity, this can be difficult to achieve before the phase-out of the currently consumed service. In this context, the impact of an outsourced service's complete unavailability[3] should always be carefully evaluated before deciding if the service shall really be outsourced.

In the case another provider offers an equivalent service to the one being dismantled, it is not necessary to re-acquire all the process's knowledge or to re-implement it in-house. It is still necessary to adapt the applications of the organization consuming the service such that they can use the new service of the new provider. This is necessary because the syntactic interface and the semantics of the new service will usually be different from the to-be-phased-out one. In that case, as shown in Figure 2, an application must adapt the integration code of its applications consuming the to-be-phased-out service to use the new service from the new service provider.

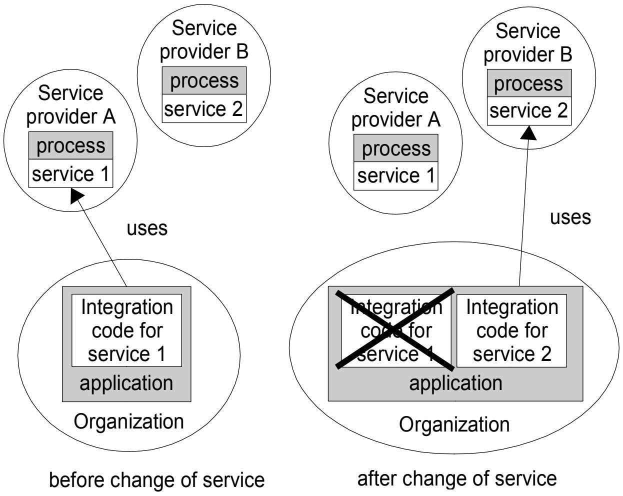

Figure 2. Integration adaptation in case of service changes

If a service provider wants to make some semantic and/or syntactic changes to the service, the situation is almost exactly the same as in the previous case. In this case, the new service provider is the same as current one and the evolution of the service itself is equivalent to the phase-out of the currently offered service combined with the start of the newly offered one.

Since an organization must adapt its applications when switching a consumed service from a service provider to another one, and since a service evolution can be considered as a service switch, it is important to notice that any service evolution done by a provider will have some impact on its consumers. Thus, a service evolution is never limited to the service provider only, but also always affects all the service's consumers. Therefore, any service evolution must be treated by organizations consuming this service as the combination of a service phase-out and the start of a new service.

Any service evolution implies some adaptations[4] or at least an impact study on an organization's internal applications that are directly or indirectly dependent of the evolving service. These adaptations or impact studies consume time

communication protocols can be made independently from other systems.

[3] We define complete unavailability if the service is itself unavailable and if there is no other equivalent service offered anywhere else.

[4] Service modifications like internal code optimizations that don't change the semantics or the interface of a service are not defined as evolutions.

and resources within the organizations using the evolving service. These organizations must have enough time to perform the adaptations between the moment they have been warned about the evolution and the moment when the current service (or current service version) will become unavailable. This time frame should be defined contractually such that the consuming organizations will never be short of time to adapt a service to a new version or to switch to a new service from another provider if the current service they are consuming is being phased-out. When the time frame is set contractually, a consuming organization is able to estimate the risk inherent to the potential evolution of this service, but only for this one. The situation becomes more complicated when an organization consumes many services from several providers.

It is widely accepted that organizations that are outsourcing information processes try to have as few service providers as possible. Nevertheless, since each provider is usually specialized in a certain information processing domain, outsourcing organizations have very often several service providers. Each of these providers processes a subset of a delegating company's outsourced tasks. Because all the contractors (service providers) of a delegating company are independent from each other, they are *un-coordinated*. The set of services offered by each provider can be assimilated to a sovereign system. This means that the services offered by each provider are independently and uniquely controlled[5]. Each provider acts as the supreme authority of these services. Consequently, the provider decides not only who can access or use his services but also when and how these services will evolve.

Because there is no central authority coordinating all service evolutions between the different service providers, organizations that are outsourcing information processes can be subject to un-coordinated service evolution notifications. Therefore, a consumer of services could be notified simultaneously by all the providers of its consumed services that all his services will evolve. Thus, a service consumer can be forced to adapt the local integration code of all his consumed services at the same time. But in this type of situation, the service consumer may not have all the necessary resources to perform the adaptations within the available time frame. Furthermore, since the new versions of the different services will likely start to be available at different times (because they are managed by different un-coordinated service providers), there might be an huge number of local integration code versions necessary to handle the different situations produced by the sequential roll-out of the services[6].

In this paper, we will study the risks that are associated to a multiple outsourcing situation. We then present a method that enables us to quantify these risks and eventually efficiently manage them.

The rest of this paper is structured as follows. Section 2 reviews the previous work that has led to the current research. Section 3 presents the evolution risk and the indicators that permit to assess the evolution exposure of an organization. Section 4 addresses the underlying model of the indicators and discusses the model's setup and assumptions. Section 5 details how the indicators are calculated. Section 6 concludes the paper and presents future research directions.

2. Previous work

To handle the multiple service evolution situation, the idea of automatic upgrades in the context of cross-organizational services has been investigated in [10]. That paper shows why there will never be any automatic upgrade mechanism for distributed processing involving several sovereign systems, despite any technical innovation that could be made in the future[7].

Because of this impossibility, several scenarios depicting multiple simultaneous service evolutions have been investigated [11]. In particular, *temporal coordination*[8] and *multiple concurrent versions* have been identified as two alternatives for performing service evolution as depicted in Figure 3 and Figure 4. However, we have not shown in the latter paper how to provide an optimal solution to the problem.

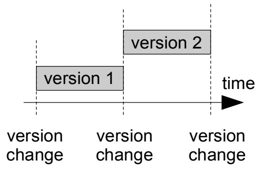

Figure 3. Serialized version release

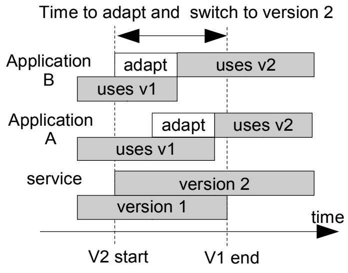

Figure 4. Concurrent services versions

We also highlighted in [11] the requirement for a service consumer to have a contractual relationship with the provider such that the former can manage the time needed for adaptations of the integration code of his local applications. Then, two types of contracts are usually encountered: the contract with a fixed termination time and the contract with a fixed termination delay[9]. As shown in Figure 5, the fixed termination time contract defines the start time and the end time of the service's availability. The

[5] By the provider that offers the services.

[6] Please refer to [10] for a detailed explanation.

[7] This is due to the fact that managers of sovereign systems will never let an outside event automatically trigger some modification in the code of their own applications and not because of any technological reason [10].

[8] Implied by serialized version releases.

[9] Called *termination period*.

fixed termination delay contract defines only the start time but not the end time of the service's availability. Additionally, it defines a termination period which starts at the time the service provider sends a *termination warning* to the service consumers as shown in Figure 6. The service becomes then unavailable at the end of the termination period.

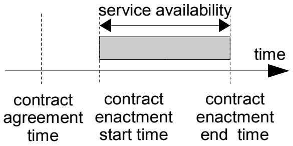

Figure 5. Undefined termination time

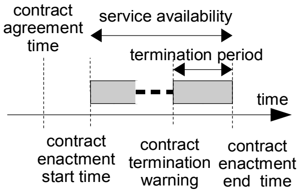
contract agreement time
service availability
termination period
time
contract enactment start time
contract termination warning
contract enactment end time

Figure 6. Fixed termination time

In the general case, a service provider may itself depend on other providers' services to offer its own services [12]. Thus, the fulfillment of the contracts that he has with his customers will depend on the contracts he has with his own service providers as shown in Figure 7. Therefore, the requested/provided services will form a network whose contractual consistency must be analyzed. In particular, one must study the dependency chain between offered and consumed contracts. We have shown in [11] how to compute the temporal conditions[10] of a contract given the others contracts it depends itself on.

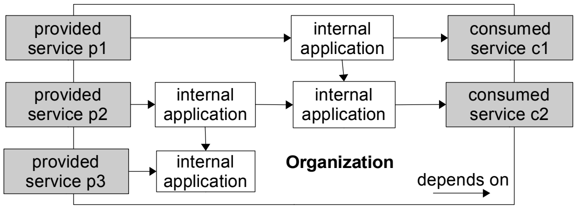

Figure 7. Dependency graph

Thus, it is possible to know at any time t if the services offered by an organization have *adequate*[11] temporal conditions or not.

3. Estimating evolution risk

In case an organization consumes several services from many service providers, it can be exposed to a very volatile adaptation workload within a defined time frame due to unpredictable external evolution events. If the workload is greater than the organization's available resources, it could threaten the organization's internal operations and consequently the availability of the services that it offers to its customers. Thus, it is necessary for an organization to know what evolution situations it will be able to handle successfully and the ones it won't be able to handle, i.e. its risks.

In other words, an organization consuming outsourced services must know what *exposure* it has relatively to external evolution events. We define *evolution exposure* as the risk endured by an organization of not being able to perform all adaptations (due to evolution events) in the required time frame such that the organization's operations can continue to run without any interruption. This *evolution exposure* can be viewed as a way to quantify an organization's ability to adapt to the evolution of the outside world it is connected to.

4. Model

To calculate these indicators, we propose a model that will use the data available within the environment of an organization. We will detail the setup of the model and present some assumptions.

4.1 Model Data

The data elements that are needed in the model are the following:

- The set of consumed services
- The set of provided services
- The application dependencies within the organization (the dependency graph)

Moreover, for each consumed service, the model requires:

- The start time
- The end time or the termination period[12]
- The estimated time to adapt the service from the current version to a new version
- The probability distribution of the service's evolution in the future

4.2 Setup and Assumptions

Let us define P as the set of services provided by an organization, and C the set of external services the organization depends on as shown in Figure 7. In the model, we suppose that any offered service by an organization has *adequate* temporal conditions. This must also be true for all consumed services that are unique to some providers. In case a service is offered by several service providers, it is not necessary for it to have *adequate* temporal conditions. In fact, the consumer of such service can always switch to another similar service offered by another provider if necessary.

In our model, we also make the assumption that any service termination warning is sent *simultaneously* with the new service specification. We also suppose that the new service

[10] In the sense of constraints.

[11] An offered service is defined having *adequate* temporal conditions when these conditions don't violate the temporal constraints of any of the consumed services it depends on.

[12] Depending on the type of contract.

is available at the time its specification is sent.

Moreover, if a service consumer has received one or more termination warnings, we suppose that he immediately starts the adaptation of the applications using these services. This means that the organization never delays the start of the adaptation (no lazy behavior).

We model a service evolution as a random event and consider all evolution events as independent. In other words, we suppose that there are no clusters of evolution events[13].

We also suppose that the very nature of the future evolutions of a service is not known. Therefore, it will not be possible to precisely estimate the time required to perform the adaptation of the local application using a consumed service. We also assume that, in most cases, the adaptation of a service in an application should take less time and resources than the initial integration of the service. Therefore, a safe estimation for the time to adapt the service's integration code in an application is the time that was needed initially to integrate that service (worst case boundary assumption).

4.3 Information Related to Future Evolutions

One of the key elements of the model is the information related to the expected evolutions of the consumed services. To be able to quantify the evolution exposure indicator, it is necessary to collect or derive information about the services' future evolutions. This information is materialized as a service's *evolution probability distribution, i.e.* the set of daily probabilities[14] for a service provider to initiate the termination of a service. If a service is being terminated, the provider sends a *termination warning* to all the service's consumers. From the moment the termination warning has been sent, a service consumer uses the *termination period* specified in its service contract to either adapt to the new service version offered by the same service provider or to find an equivalent service offered by some other service provider and adapt to that one.

The evolution probability distribution of a service can be modeled in different ways. This depends on the possibility to collect information about the expected evolutions of a service from it's provider. If yes, one approach is to literally question the provider about the expected evolutions of the service and translate his answers into a probability distribution. An automatic data collection proposal of this kind is described in Chapter . In case it is not possible to get any information from the provider, it is then necessary to make arbitrary assumptions about the distribution. One possibility is to use a heuristic based on the life cycle of the service's precedent versions or on the life cycles of all services versions known to the consumer. Another possibility is to assume that there doesn't exist any available data about the future evolutions of the service. In that case, it may be appropriate to set a daily probability of termination that is constant for a defined time interval. However, the way to handle this lack of information must be done by the organization on a case-per-case basis.

4.4 Indicators

Along with the evolution exposure, we present several other indicators that permit to quantify the capacity of an organization to manage the evolution the external services it depends on. The complete set of these indicators is the following:

- PERC-SUCCESS: The percentages of situations where an exact set of simultaneous events can be handled successfully by the organization.
- PROB: The probabilities of occurrence of an exact number of simultaneous evolution events.
- PROB-SUCCESS: The probabilities of receiving and handling successfully an exact number of simultaneous evolution events.
- PROB-RELATIVE-SUCCESS: The probabilities of handling successfully an exact number of simultaneous evolution events.
- EVOLUTION-EXPOSURE: The evolution exposure.

4.5 EVOLUTION-EXPOSURE

The EVOLUTION-EXPOSURE indicator consists of the probability for an organization of being able to handle successfully a situation of evolution events that could occur at the time of the calculus. This probability is not specific to a particular number of simultaneous evolution events but is related to any amount[15] of simultaneous evolution events. We denote this probability as the *evolution exposure* which can be seen as the synthetic answer to the question:

"What is the probability that an organization will be able to adapt to the evolutions of its outside world within a pre-defined time frame such that it will not suffer any disruption due to these evolutions ?"

The *evolution exposure* indicator represents the probability of receiving termination warnings and being able to perform the adaptations successfully within the contractual time frame of an organization. The EVOL-EXP indicator is calculated by summing up all probabilities of the PROB-SUCCESS indicator together:

$$\text{EVOL-EXP} = \sum_{i=1}^{N} \text{PROB-SUCCESS}[k]$$

[13] This may not be true in reality, since often a service provider may evolve a set of services at the same time.

[14] As estimated by each service consumer.

[15] Up to the totality of all consumed services.

5. Conclusions and future work

In this paper we have developed a way to quantify the exposure of an organization, relative to the evolution of the outside world's services it depends on. We have presented a number of indicators that can help an organization to estimate the degree of exposure to these external and uncoordinated service evolutions.

These indicators are dependent on both internal and external information. The internal information consists of the estimated amount of work needed to adapt the local integration code in the case of a consumed service's evolution. Because of the impossibility of knowing the nature of future changes, we have assumed that the time to perform a change is equal to the time spent for its initial integration (worst case assumption). The external information consists of all information that is obtained from the service providers on the probability of evolution of their offered service that is consumed by the organization.

These exposure indicators can be used in different ways. First, they can help management of an organization to assess in a quantified manner the number of simultaneous external evolution events that can be handled by the organization. Then, the organization can express the external evolution risk level that it considers to be acceptable for itself and can then adjust the amount of resources (i.e. the size of the maintenance team) and/or the length of the contracts' termination periods that are needed to comply with that risk level.

In a forthcoming paper, we will include the notion of disruption cost to quantify the financial risk of an organization relative to the evolution of the consumed external services it depends on. Eventually, in the case an organization has identified these financial risks, it will then possible for the organization to set as a target a *desired* financial exposure. The organization will then be able to choose between several strategies to reach this desired exposure. These strategies will include: hiring more people, extending some of the consumed service contracts or reducing the termination periods of its own offered services.

6. REFERENCES

[1] Georg Erber and Aida Sayed-Ahmed. Offshore Outsourcing - A Global Shift in the Present IT Industry. In Intereconomics, Volume 40, Number 2, 2005.

[2] Tomi Dahlberg, Mari Nyrhinen. A New Instrument to Measure the Success of IT Outsourcing. In Proceedings of the 39th Hawaii International Conference on System Sciences, 2006.

[3] Suzanne Berger. How We Compete: What Companies Around the World Are Doing to Make it in Today's Global Economy, Currency, 2005.

[4] Ravi Aron, Eric K. Clemons, Sashi Reddi. Just Right Outsourcing: Understanding and Managing Risk. In Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

[5] Bouchaib Bahli, Suzanne Rivard. A Validation of Measures Associated with the Risk Factors in Information Technology Outsourcing. In Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), 2003.

[6] Lea Kutvonen. Automated management of inter-organisational applications. In Proceedings of the Sixth International International Enterprise Distributed Object Computing Conference (EDOC'02), 2002.

[7] Benoit Aubert, Jean-Francois Houde, Michel Patry, Suzanne Rivard. Characteristics of IT Outsourcing Contracts. In Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), 2003.

[8] Alexander Keller, Gautam Kar, Heiko Ludwig, Asit Dan, Joseph L. Hellerstein. Managing Dynamic Services: A Contract Based Approach to a Conceptual Architecture. In Proceedings of the 8th IEEE/IFIP Network Operations And Management Symposium (NOMS 2002), 2002.

[9] S. Angelov, P. Grefen. Support for B2B e-Contracting – The Process Perspective. In Proceedings of the 5th International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services, 2002.

[10] Odorico von Susani, Philippe Dugerdil. Contract-Based Cross-Organizational Automated Processes. In Proceedings of the 7th IEEE International Conference on E-Commerce Technology (CEC 2005), 2005.

[11] Odorico von Susani, Philippe Dugerdil. Cross-Organizational Service Maintenance Using Temporal Availability Specification and Contracts. In Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM 2005), 2005.

[12] B. Benatallah, M. Dumas, M-C. Fauvet, F.A. Rabhi. Towards Patterns of Web Services Composition, Springer-Verlag, 2002.

[13] Philippe Dugerdil. Optimum maintenance process. Technical Report, Haute Ecole de Gestion, Geneva, Switzerland, 2007.