Managing Faults in the Service Delivery Process of Service Provider Coalitions

Patricia Marcu
Munich Network Management Team
Leibniz Supercomputing Centre
Boltzmannstr. 1, 85748 Garching, Germany
marcu@mnm-team.org

Larisa Shwartz, Genady Grabarnik
David Loewenstern
IBM T. J. Watson Research Center
19 Skyline Drive, Hawthorne, NY, 10532, USA
{*lshwart, genady, davidloe*}*@us.ibm.com*

Abstract

In recent years, IT Service Management (ITSM) has become one of the most researched areas of IT. Incident Management and Problem Management form the basis of the tooling provided by an Incident Ticket System (ITS). As more compound or interdependent services are collaboratively offered by providers, the delivery of a service therefore becomes a responsibility of more than one provider's organization. In the ITS systems of various providers seemingly unrelated tickets are created and the connection between them is not realized automatically. The introduction of automation will reduce human involvement and time required for incident resolution.

In this paper we consider a collaborative service delivery model that supports both per-request services and continuous high-availability services. In the case of high availability service the information stored in the ITS of the provider often includes information on the outage of a particular service rather than on the failure of a particular request. In this paper we offer an information model that consolidates and supports inter-organizational incident management and probabilistic model for fault discovery.

1. Introduction

ITSM focuses on the development of methodologies and tools that facilitate providing high quality IT services with maximum efficiency and dependability. Incident and Problem Management processes of ITSM are critical for successful delivery as stressed by the IT Infrastructure Library (ITIL) as best practice in the management of the IT infrastructure, development and operations [1].

The Incident Management Process is supported by various tools including Incident Ticket Systems (ITS). These are software systems used in an organization to record information about IT service failures or malfunctions as well as the degeneration of the functionality of the IT infrastructure.

The delivery of a service within an organization is usually a well-understood and controlled process. IT Service Delivery processes that span multiple organizations are not investigated enough. Since many providers participate in the delivery of a composed service, the root cause of an incident issued by the customer is not easy to identify. To facilitate localization and resolution of an incident, an inter-organizational ITSM (ioITSM), with an inter-organizational (**io**) CMDB as its most important part, aims to support the management of the processes that engage various organizations [2].

Inter-organizational IT service delivery is defined by the fact that the IT service is delivered collaboratively by a number of providers or suppliers. In this paper we discuss services in a broad sense as a co-production of consumer and provider. Examples of services are hosting services, internet/network providing services as well as web service. We consider a IT service delivery model that supports per-request services and high-availability services. In case of high-availability services fault localization is particularly time-consuming due to a large redundancy that is usually built into the fulfillment process for this type of service.

In our approach we propose a generic data model that consolidates and supports **io** Incident Management. Moreover this is used to correlate faults from different IT service provider organizations. While request ID is useful information for ioITSM, it is unrealistic to expect that it will be always preserved in the ioITSM of the providers of high-availability services. In this paper, we built a probabilistic model for fault discovery and outline major principals for minimizing the length of a search path and the time necessary to identify the service provider responsible for the fault.

The paper is structured as follows: a motivating example is described in Section 2. Section 3 provides a brief overview on related research. Inter-organizational service delivery models, especially heterarchical and high availability services are described in Section 4. Section 5 introduces new concepts and describes the **io** fault correlation method. Section 6 describes the formalization of the algorithm for a special case of Service Provider Coalition (SPC). Section 7 concludes the paper with open issues and further work in this research area.

2. Motivating Example

The GÉANT2 network is the first international production hybrid network, combining the operation of a shared IP infrastructure (basic service) with the ability to

978-0-7695-3811-2/09 $26.00 © 2009 IEEE
DOI 10.1109/SCC.2009.41

provide additional dedicated point-to-point links (advanced services). The point-to-point services *GÉANT Plus* and *GÉANT Lambda* represent a new era in networking and telecommunications technology, and aims to reach new levels of service to the research and education community [3].

The basic service, now known as *GÉANT IP*, provides access via the GÉANT2 network to the shared European Internet Protocol (IP) network. It offers a robust, high-bandwidth solution to the international connectivity requirements of the majority of academic users, allowing transit for IP traffic between European National Research and Education Networks (NRENs), and between European NRENs and associated networks globally. Part of the seventh generation of the European research and education backbone, the *GÉANT2 IP* network is over-provisioned by design, to allow small-to-medium-sized traffic flows (*i.e.*, up to 1 Gb/s) an uncongested path.

GÉANT2 offers two distinct classes of point-to-point services to NRENs who require dedicated international circuits for their users: *GÉANT Plus* and *GÉANT Lambda*. The *GÉANT Plus* service allows NRENs to request point-to-point circuits of between 155 Mb/s and 10 Gb/s across an existing network of pre-provisioned links. *GÉANT Plus* is built on a shared infrastructure. The *GÉANT Lambda* service provides private, transparent 10 Gb/s bandwidth between GÉANT2 NRENs. It is only available to NRENs subscribing to the GÉANT Plus service.

Key to the successful delivery of the point-to-point services is the *End-to-End Coordination Unit (E2ECU)*, which is responsible for the overall monitoring of E2E circuits and for coordinating the information flow and communications between the actors in the different domains involved in each E2E circuit. The E2ECU may be notified of an outage either by E2EMon (E2E Monitoring System) or by someone in the domain. On being notified, the E2ECU raises a Trouble Ticket (TT) containing information such as the names of the domain link or interdomain links affected, the name(s) of the domain(s), the name of the project affected, and the time of the outage. The E2ECU then contacts the relevant domains to request information regarding the outage and to assist them in interpreting the errors; in the case of an inter-domain link, the E2ECU will contact both domains involved. The E2ECU distributes any updates regarding the outage to all partners in the project affected.

3. Related Work

This section reviews prior research related to the correlation of trouble tickets for Incident and Problem Management and fault diagnosis and to service provider inter-organizational (**io**) service delivery.

Van der Aalst focuses in [4] on **io** workflows, *i.e.*, workflows crossing organizational boundaries inside the company or between companies. Two architectures for **io** workflow are described here: extended case transfer and loosely coupled. In the case of extended case transfer a vertical decomposition is used so workflow instances are partitioned over the business partners involved. By contrast the loosely coupled **io** workflow uses a horizontal decomposition where the process itself is partitioned. This means that each partner has its own private workflow that somehow connects to the workflow processes of the other partners. In our work we will take into account this approach in the specification of the service interoperability type.

In his work [5], Hedlund points out the restrictions that exist in the hierarchical organizational structure for service delivery. Therefore he proposes another organization of service delivery: the heterarchy. Another organizational structure for service delivery that we consider in our work is heterarchy. The concept of heterarchy in IT Services as horizontally chained services is further researched in [6].

In previous work [7] we introduce the concept of Service Provider Coalition (SPC). A SPC is a a group of IT service providers that together supply a composed service and have group authority and responsibility to consumers of their services. Within an SPC, each of the providers provides a part of the service to a customer. A brief idea about correlation of incident tickets is stated but this can be only applied on pre-request services. In the current paper we extend this idea especially on the IT service delivery of the SPC, we propose an fault manifest correlation model and we formalize this.

In [8] we proposed a model to correlate customer incident tickets and provider' resource tickets based on three criteria. According to the algorithm described there the correlation happens in three stages. First a category-based correlation that relies on matching service identifiers with associated resource identifiers is performed using similarity rules. The correlation of configuration items that are critical to the failed service with previously identified resource tickets in order to optimize the topological comparison follows. Constraint adaptive probing, that extends work described in [9], is finally done in order to minimize the correlation interval for temporally correlated tickets. Mentioned above paper together with this paper describe fault localization in inter-organizational service delivery process from the incident identified by customer to the faulty resource identified by the provider.

Lange and Nerb propose in [10] a trouble report format, which extends Customer Service Management (a management entity that addresses the relationship between customer and provider) towards bidirectional inter-domain Problem Management. A generic interface and a generic set of information are defined. These can be applied independently of the service and the position in the hierarchy. While it is a valid approach in **io** problem management, it doesn't take into account other organizational structures besides hierarchy.

An **io** Configuration Management Database (ioCMDB) is depicted in [2] as an enabler for **io** IT Service Man-

agement (ioITSM) Processes. This work describes processes for ioCMDB usage and an ioCMDB information model. An ioCMDB stores references to information in the CMDBs of different organizations participating in **io** service delivery.

We would like to thank our reviewers for bringing to our attention two additional references. Tai *et al.* [11] introduces service clubs as a collaboration place for a specific service community, the environment for exchange of complimentary or interchangeable services. By contrast, in this paper we describe the results of our research in delivery aspects of service management. Udupi *et al.* [12] considers policy-based governance of cross-organizational service agreements. By contrast, our paper deals with cross-organization service management for hierarchical and heterarchical service delivery models.

4. Inter-organizational Service Delivery

As described in Section 2 using the concrete example of the GÉANT2 network, in large, inter-organizational service provider environments more kinds of services exist. In our example there were basic services and advanced services. Hence in this section we will distinguish between the delivery of two types of services: per-request services and high availability service for continuous operations. In Figure 1, customers A, B and C contract with a service provider providing a composite **Service**. The service provider fulfills the service by subscribing in turn to per-request services 1 and 2, provided by providers 1 and 2; and high-availability service 3, itself a composite service provided by a Service Provider Coalition.

4.1. Per-request services

Service 1 and **Service 2** deliver partial tasks needed by the **Service**. They are delivered by a service provider (SP) that orchestrates overall delivery of composite services through decomposition and allocation of providers for atomic services (from its point of view), and requests and receives responses back to its queries. This kind of service delivery is offered for long running processes.

The SP has sole responsibility for a service that is being consumed by the customer (and its users) and is a single point of contact for customers. The SP could subscribe to other services from other providers in order to fulfill this service. Each of the providers communicates with the SP, which is the single point of contact with the customer. A very simple example is the e-mail service that is provided to a consumer organization by the email SP organization. The SP in turn subscribes to the services of the provider of DNS (for instance Service 1 or Service 2).

In this type of organization of service delivery, the SP has a responsibility to customers for the overall services and therefore the SP requires knowledge of service delivery and execution at all stages of the fulfillment process. The Fault Manifest Management system enabled by ioCMDB and CMDBf proposed here aims to help the SP to facilitate service incident localization.

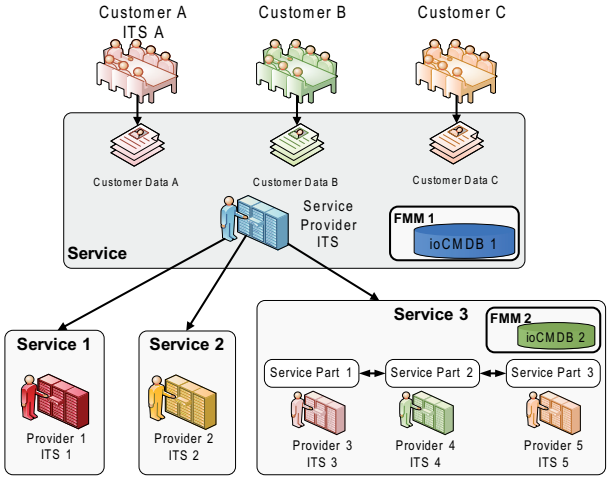

Figure 1. Complex inter-organizational service delivery model

4.2. High availability services

Service 3 is delivered by a Service Provider Coalition (SPC). This has been described in detail in [7]. The probabilistic model that we discuss in Section 6 of this paper models the delivery process of a SPC as a directed acyclic graph.

The flow of information and provider-consumer relationships can be established at the time of service design and hard-wired into the ticket information system, making it is easy for the SPC to relate ticket information. However, as the goal of the SPC is to achieve high availability of the services for the consumers, the SPC organizes the providers' services into *late-binding* services with a high level of redundancy, which can perform best for the consumers under specific circumstances [13]. In this case the model for the sharing of ticket information has to include information needed to select correct providers and consumers somewhat independently.

Heterarchy as defined in prior art is a horizontally chained organizational structure of providers that cooperatively deliver services to a customer. The SPC service delivery model includes heterarchy as a special case. This kind of service delivery model is used, for example, for providing services in multinational projects and complex distributed environments that provide parallelism, dynamic computations, dynamic data access to large data sets and long running computations and require high availability services.

In Section 2, the End-to-End Link service is an example of a high availability service in large multinational networks where a negotiator entity defined in [6] coordinates interaction with customer for service delivery. One representative project is the provisioning of the infrastructure for the Large Hadron Collider (LHC) at CERN in Switzer-

land. It is expected that its experiments will produce 15 Petabytes yearly. As stated before E2E Links connect organizations located in different countries and cross the networks of different providers (domains). In providing the E2E Link services the provider (member of the SPC) has to collaborate in setup, maintenance and management tasks. One of the major problems in the realization of this service is tool heterogeneity (*e.g.*, different ITS's in different domains).

In our work we introduce a Fault Manifest Manager (FMM) that utilizes an ioCMDB and the CMDBf protocol in order to facilitate the exchange of incident information between providers of a service delivered by a SPC.

5. System and Method for Correlation

In this section we describe the inter-organizational Fault Manifest Management (FMM) system and methods for the correlation of fault information in response to a customer's Incident Manifest.

In the first subsection the new concept of Fault Manifest will be defined. Although many formats of trouble tickets exist, there isn't a 'common' ticket format for **io** service delivery established yet. The semantic obstacle is in this context really big. That's why we picked up the most valuable information which is significant for our approach especially that information that more service provider possibly will share respectively will post.

The next two subsections describe the components involved in the correlation and methods of correlation. The last subsection describes the method of communication that is required to support the correlation.

5.1. Fault Manifest

A fault manifest is information provided by the participants of a service's delivery in order to enable communication with the goal of achieving quality of overall service. For the two different service models we allow two different manifest types: the incident manifest used in per-request services and the outage manifest used in high availability services that support continuous operations.

In Figure 2 the XML-Schema for the fault manifest (**FaultManifType**) is represented. We consider the complex type **FaultManifType** and the extensions (elements) **IncidManifType** (for incident manifest) and **OutageManifType** (for outage manifest).

The type **FaultManifType** has following attributes:

- *faultID*: the unique identifier for the fault manifest.
- *serviceID*: the unique identifier for the service on which the consumer and the service provider (SP) or SP coalition (SPC) have agreed.
- *description*: a human-readable text field with the description of the incident.
- *status*: the reported status of the incident, which can be new, pending, closed or another gradation on which the customer and the SP or SPC have already agreed.

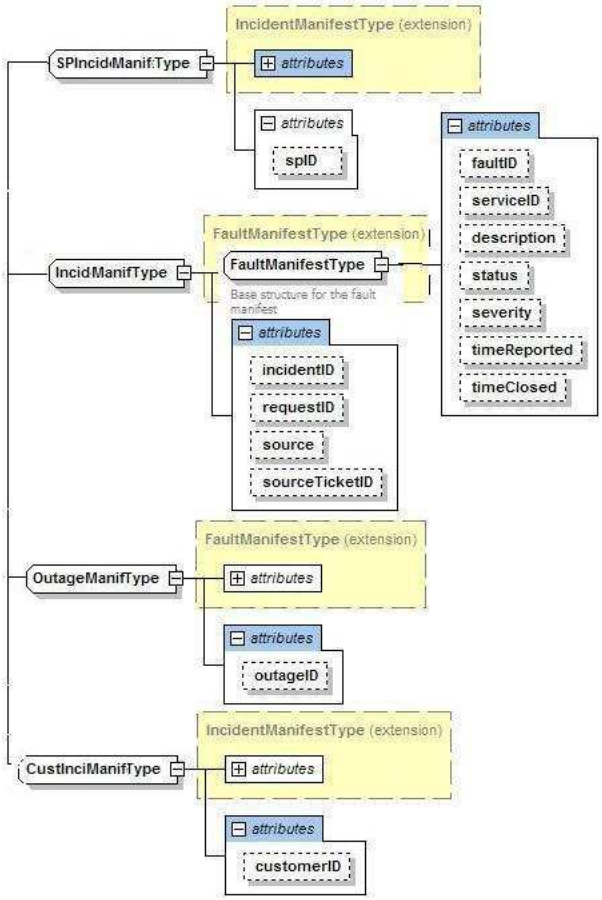

Figure 2. XML-Schema for Fault Manifest

- *severity*: represents the degree to which the service is affected as perceived by customer or by provider respectively to the source. Similar to the *status* attribute, possible values for severity are defined in advance and agreed upon between consumer and the SP or SPC.
- *timeReported*: the time when the incident was reported.
- *timeClosed*: the time when the incident was closed. This and the previous attribute are relevant for statistical information concerning the incident resolution time and for evaluation for the compliance to the SLA.

1) Incident Manifest. The *incident manifest*(IM) as described in [7] is needed in order to restrict the information that will be exchanged between the different ITSs. The smaller the set of information transmitted between the ITSs, the easier the correlation between tickets.

The IM type is here represented in an XML-Schema format by **IncidManifType**, which has two extensions: **CustIncidManif** (customer IM) and **SPIncidManif**, (SP IM). The attributes described in our former work are the same.

	Customer IM (CustIncidManif)	SP IM (SPIncidManif)	Outage Manifest (OutageManif)
Attribute	Value	Value	Value
incidID	320054D	453999	
outageID			18645
serviceID	ipconn0020	ipconn0020	ipconn0020
requestID	session342434		
description	Connection down	Router down	Router Down
source	customer	provider	SPC
sourceTicketID		12345678	
status	pending	new	active
severity	high	medium	high
timeReported	200811112011	200811111922	200811111900
timeClosed			
customerID	A2816AB		
spID		A2824CN	A3341EF

Figure 3. Example of different Fault Manifest

Example: A **CustIncidManif** is given in Figure 3 (first column of the table). The second column exemplifies a **SPIncidManif** (SP IM), created by the FMM as a result of discovering an **OutageManif** (third column) in one of the ITSs of providers in SPC.

Note that providers can act as SPs for their customers as well as consumers of other services. Therefore customerID and spID could have same value in **CustIncidManif** and in **SPIncidManif**.

Example: In Figure 3 the customer **A2816AB** of the ip-connectivity service is also a provider for another customer for the e-mail service **email123** (*serviceID*). In this case the former customer becomes a SP with the *spID* **A2816AB** so he is publishing a SP IM with a *incidID* **320054D** (the same as published as customer) with the same description at the same time. The *source* is now provider, the *status* is **new** and the severity is medium for the this customer IM.

2) Outage Manifest. This type is similar to the IM but is only specific to SPC, particularly to the SP of the SPC. As we described above an IM needs a *requestID* which identifies the task requested by the customer of the SP when the incident occurred. As high availability services are continuous services, they do not depend on a requestID.

The outage manifest is represented by **OutageManif**, which has an additional attribute *outageID* representing the unique identifier for the outage within the provider domain represented as string.

5.2. Fault Manifest Management System

A SP that interfaces directly to an external customer requires a point of contact for incident management, regardless of whether the SP is a stand-alone, manages a delivery of a service by providers, or is a SPC, and also regardless of whether the external customer is an end-user or only another SP. Typically in a hierarchical model there is one ITS that also has responsibility for incident management for the entire hierarchy; in a SPC one ITS may be assigned the role. Since this point of contact works with the fault manifests described in the previous section, we call it the Fault Manifest Manager (FMM).

Example: If the FMM handles incidents pertaining to tickets in more than one organization, then will require an **io** CMDB (ioCMDB), as described in [2] to manage configurations across organizations. In figure 1, the hierarchy has an FMM (FMM1) that manages incidents for three services. Services 1 and 2 do not have a separate FMM because FMM1 is responsible for their incident management, but in this example, Service 3 has its own FMM (FMM2).

The FMM is responsible for establishing the correlation between a customer incident report and the tickets generated as providers address the faults causing the incident. The FMM extends and incorporates all the components involved in the **io** incident correlation as described in [7]. It contains two components to do this work.

1) The Correlation Engine. correlates the information published by different (provider or customer) ITSs.

2) The ioCMDB. stores information about customers, providers, services and parts of services as agreed upon between the consumers and providers, as well as IMs and some transactional information for tracking purposes.

a*) Service Catalog.* The FMM provides customers and providers with an interface for service registration during the **design and on-boarding** process (**initialization process**). The information provided is described in [7].

b*) Customer Incident Manifest Catalog.* Customer IMs are stored in the ioCMDB as they are created from a customer incident report or received from a customer FMM.

c*) SP Fault Manifest Catalog.* Provider IMs as well as Outages manifests are stored in the ioCMDB as they are generated from the data received from a provider ITS or received from the FMM of a SPC.

d*) Transaction Record.* Every SP IM handled by the FMM generates a record of the transaction, including the customer ID, incident ID and service ID. Also, the relationships connecting the transaction to the corresponding entries in the Service Catalog, Customer IM Catalog and SP Fault Manifest Catalog are maintained in the ioCMDB. These entries index request information that is later used to bypass the correlation process for updates.

e*) SPC Delivery Processes (SPCDP).* During design and the on-boarding process the SPC provides SPs with a delivery process (SPCDP). The comprehension of this service delivery process as being composed from atomic processes provided by various providers in the SPC is used by the SPC's ioCMDB FMM for fault discovery.

5.3. Method of correlation

In order to do the correlation we will describe three kinds of processes: initial registration, submission of a customer incident report, and update. We will describe here all of these processes but a detailed activity workflow and the algorithm for fault discovery in SPCDP will be shown only for the initial submission of a customer incident

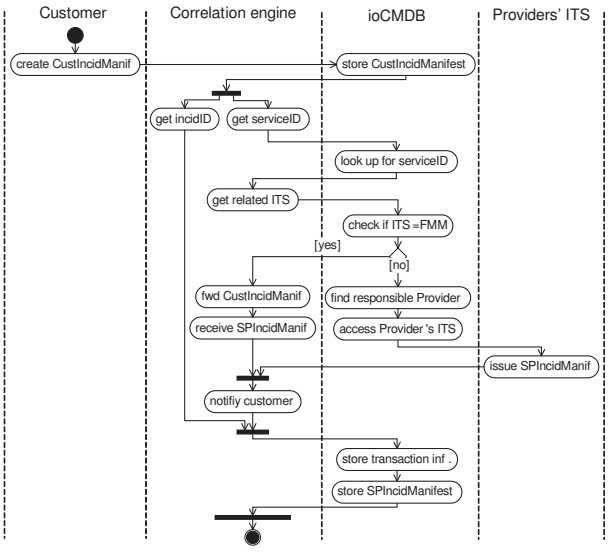

Figure 4. Activity Diagram for the Correlation of Incident Manifests

report. Parallel to the description of the proceses, the use of the manifests in section 5.1.1 and components in section 5.2 will be illustrated using a scenario based upon the model in Figure 1.

1) Initial registration. During initial registration, each ITS registers with its FMM, announcing which services it provides.

Example: ITS 1 and 2 register their services with FMM1, and ITS 3-5 register their services (service 3 parts 1-3) with FMM2. FMM2 also assigns particular atomic services (from its point of view) within SPCDC to be handled by each of ITSs 3-5. Note that the detailed process of this assignment is out of scope for this paper. Finally, FMMs may register with other FMMs that manage combined services: in this case FMM2 would register service 3 with FMM1. The registration data (including whether an ITS or FMM was registered) are stored in the Service Catalogs of the ioCMDBs.

2) Submission of a customer incident manifest. Figure 4 shows the activity diagram for the correlation of IMs triggered by the submission of a new customer IM. The four swim lanes correspond to the components described in 5.2.

The customer performs the first activity in the diagram: the sending of an incident report to the FMM. If the customer has an ITS (or is itself an FMM), this report will be in the form of an instance of the class CustIncidManifest. Otherwise, the correlation engine is responsible for parsing the report and creating a customer IM from the contents.

This manifest is stored in the next step in the Customer IM Catalog of the ioCMDB. The service ID from the manifest is then used to look up the SPs or SPCs that may have been responsible for the service in the Service Catalog. In the case of a SP, the correlation engine requests from the provider's ITS whether the provider had actually processed the request identified in the incident report. If so, it requests a ticket ID for the service that was either created when the request was processed or was already open at that time. In a SPCDP, it is possible that a provider did process the request without error because another provider in the service chain encountered the error: in this case the correlation engine requests the provider ID that the provider had sent the request on to and adds it to the list of SPs to query. The correlation engine continues until a ticket has been found. If all potential providers are queried and no ticket exists, then either the customer has made a mistake or there is a serious fault in a provider ITS. In some cases the SPC won't store or provide fault information based on a request identifier, but rather on an outage report. In this type of operational setting, the FMM correlation engine uses a fault discovery algorithm further described in section 6.

In the case that the service may have been provided by a SPC, the provider forwards the Customer IM to the SPC and receives the corresponding SP IM. Otherwise the correlation engine is responsible for constructing an SP IM containing the ticket information. In either case, the SP IM is sent to the customer and is also stored in the SP IM Catalog, and a transaction record is generated.

Example: Customer A notices a failure for request R1 for Service 1. It contacts FMM1 and sends a customer IM (figure 2) containing an incident ID (incid1), the service ID (Service 1) and the request ID (R1), as well as other information as described in section 5.1.1. FMM1 looks up Service 1 in its ioCMDB and notes that ITS1 is responsible for that service. It sends a query to ITS1 for all open tickets associated with Service 1 and impacting R1. ITS1 checks its own CMDB and notes that R1 was not processed because of a fault associated with ticket T1. FMM1 then sends to customer A a SP IM (figure 4) containing the status information from T1. It also records the transaction in its ioCMDB, keyed by incident ID and customer. Assuming that the SPC in the example uses requistID based IMs, FMM2 then repeats the process with ITS4 and if necessary ITS5 until it locates the provider that failed to service R2 and the associated ticket T2. It then sends back to FMM1 a SP IM containing T2's status information, and FMM1 forwards this back to Customer B. In addition, both FMM1 and FMM2 record the transaction in their ioCMDBs.

Customer B also notices a failure, this time for request R2 for Service 3. It issues a customer IM incid2 to FMM1. FMM1 looks up Service 3 in its ioCMDB and notes both that Service 3 is managed by FMM2 and that FMM2 is an FMM and not an ITS. It therefore forwards incidID2 to FMM2. FMM2 looks up Service 3 in its own ioCMDB and determines that the initial step is provided as Service Part 1 by ITS3. FMM2 then requests from ITS3 any tickets associated with Service Part 1 and R2. In this example, ITS3 has no tickets. FMM2 then requests which provider received R2 from Provider 3, and ITS3 consults its CMDB and determines that R2 was passed along to Provider 4.

3) Updates. During the period from the initial incident report until the final resolution, the customer will want periodic status report updates. The data is sent to the customer as SP IMs. Where an incident required communications through a chain of FMMs (as in request R2 in the example above), the updates will still require passing the SP IMs back along the chain: this is to avoid the security and privacy issues involved in having FMMs not directly contracted with a customer sending data directly to the customer.

The scenario illustrates the need for two different types of communication requirements. The customer communicates with an FMM and an FMM communicates with another FMM exclusively through IMs. Communication between FMMs and their SPs or SPCs require a more complicated method for the case that no requestID exists. In these circumstances the fault localization is particularly difficult and, in a case when SPC has large redundancy and provide complex composite services, localization of fault could take a lot of time. Therefore a formalization method is proposed in the next section.

6. Formalization of the SPC service delivery process

As we discussed earlier the information model of service delivery could include tagging the service request and each of atomic services within service fulfillment with the request ID. However it is common for high availability services (such as services described in Section 2) not to include the request ID in their informational model. In these circumstances the fault localization is particularly difficult and, in a case when SPC has large redundancy and provides complex composite services, localization of fault could take a lot of time. In this section we build a probabilistic model for fault discovery and outline major principles for minimizing the length of the search-path and the time necessary for identifying the service provider responsible for the fault.

We model a SPC service delivery process (SPCDP) as a directed acyclic graph (DAG), where nodes represent services or tasks which we may treat as atomic services (AS's) executed by a single provider, and where directed edges model data or discreet events passed from one node (source) to another node (target). We assume that a SPCDP has one entrance point and one exit point. We also assume that each AS has one unvarying set of inputs and one unvarying set of outputs.In addition to AS's and directed edges we consider parallelization elements: forks, joins and merges. A fork is an element that enables continuation of the service process by several AS's in parallel at the same time. Join is a convergence of the outputs of two or more AS's into a single AS such that data is passed to the subsequent AS if and only if every one of the incoming AS's outputs data. A merge enables continuation of two or more AS's as a single AS as soon as at least one incoming AS outputs data. A merge is the key element used in delivery processes in order to provide high availability services by parallelizing the execution of tasks.

6.1. Fault Discovery

We assume that the fault discovery process in a SPCDP starts when a customer reports a service fault. Fault discovery and recovery could be done with a number of different goals in mind: one is the discovery of all possible faults, another one is to find and repair as soon as possible at least those faults that would allow SPCDP to restore service delivery, called a *service-path*. These goals result in different discovery policies, of which we consider these two defined by taking in account the probabilistic nature of the faults' occurrences:

- Discover-service-path policy: Find the method that on average, most quickly discovers the faults which, when repaired, are sufficient to restore SPCDP.
- Discover-All policy: Find the method that on average, most quickly discovers all possible faults within SPCDP.

Note that these policies are the most natural policies although they do not cover all possible policies.

1) Average Search Length. To measure the effectiveness of search algorithms we introduce a notion of average search length (ASL) as following $ASL(p) = \lim_{\#searches\to\infty}(\frac{\textit{length of searches}}{\#searches})$. The convergence established in prior art. The proof is based on the following proposition:

Proposition 1. Calculation of ASL. Let $l_p(S)$ be the length of a search (a number of queries made) for a combination of faults $S = \{s_{i_1}, s_{i_2}, ..., s_{i_k}\} \subset SPCDP$. Then$ASL(P) = \sum_{S\subset SPCDP} l_p(S)p(S)$. Here $\sum$ is taken over all subsets of the nodes in SPCDP and $p(S)$ is a probability of algorithm stopping on the subset S.

Example. The ASL evaluation for the service-path policy. Possible combination of faults for a failed sample SPCDP are $\{s_1\}$, $\{s_2, s_3\}$, $\{s_1, s_2\}$,$\{s_1, s_3\}$, $\{s_1, s_2, s_3\}$. It is sufficient to query service s_1 and either service s_2 or s_3 for discovery of SPCDP faults. If the probability for subsets s_1, s_2 and s_1, s_3 has a tie, we choose one with the higher probability of faults based on historical data, if not, then randomly choose one of two. Say s_2 has a higher probability of fault than s_3. Now the question is which of $s_1 or s_2$ should be queried first. This decision is made based on the known fault probability of both s_1 and s_2. For the service s_1, it is $p^{(1)} = p(s_1, s_3)+p(s_1)$ and for the service s_2 it is $p^{(2)} = p(s_2, s_3)$. Suppose that $p^{(1)} > p^{(2)}$, then the optimal algorithm is:

1) Query s_1. If s_1 is faulty, we found the point of failure for SPCDC, otherwise
2) Query s_2.

The ASL for this algorithm is $1 \cdot p^{(1)} + 2 \cdot p^{(2)} on in case of independent faults 1 p_1 + 2(1 - p_1)p_2p_3$.

6.2. Service-path faults in SPCDP

Recall that a service-path of a SPCDP does not have a failure if none of its elements have faults. This implies that a service-path (SP) consists only of sequentially executed AS's and joins of AS's and has the same start and end points as the SPCDP. In this section we presume that faults are independent. The probability of failure of a service-path is$p_{sp} = 1 - \prod_{i=1}^{k}(1 - p_i)$, here p_i is the probability of a fault of an AS in the service-path. Suppose that we have some algorithm P. We suppose that algorithm P is memoryless and deterministic. Than $ASL(P)$ can be calculated as $ASL = 1p_1 \prod_{i=2}^{n}(1 - p_i) + 2p_2 \prod_{i=3}^{n}(1 - p_i) + ... + np_n = \sum_{i=1}^{n} iF_i(p_i, p_{i+1}, ...p_n)$, where F_i is defined as $p_i(1 - p_{i+1})(1 - p_{i+2})(1 - p_n)$.

The following Proposition 2 outlines optimal algorithm for the fault search.

Proposition 2. Optimal faults search for service-path. Minimal ASL is reached for the algorithm that queries AS's in order of their fault probability decreasing.The proof of the proposition may be found in the extended revision of this paper.

7. Conclusions and Future Work

In our approach we realize a generic information model that consolidates and supports inter-organizational Incident Management, which is necessary for identifying the service provider responsible for the fault. We further consider different service delivery models to understand how ioITSM complies with their inter-organizational needs. A typical example of such collaboration is the End-To-End link service in the large multinational network supporting the Large Hadron Collider (LHC) in CERN, Switzerland. In this paper we outline inter-organizational collaboration and introduce necessary data structures and realize a generic information model that consolidates and supports the inter-organizational Incident Management. We also built probabilistic model for fault discovery and outline major principals for minimizing search-path and the time, which are necessary for identifying service provider responsible for the fault. Although the problems which appear in the implementation process in GÉANT2 network where manifold: cultural, human, language factors were predominant but also the gaps between the different stages of the technological evolution of each network.

In the future we plan to extend the simulation to reflect the all-faults policy algorithm and represent a wider range of possible SPCDP configurations including different exit criteria for the SPCDP correlation process. We will also test our methods against fault data from a real service delivery environment.

Acknowledgments

This paper is a result of collaboration with researchers at the IBM T.J. Watson Research Center that was initiated during a summer internship in 2008.

The authors wish to thank the members of the MNM Team for helpful discussions and valuable comments on previous versions of the paper. The MNM Team, directed by Prof. Dr. Heinz-Gerd Hegering and Prof. Dr. Dieter Kranzlmüller, is a group of researchers of the Ludwig Maximilians Universität München, the Technische Universität München, the Universität der Bundeswehr München and the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences. Its web server is located at http://www.mnm-team.org

References

[1] "IT Infrastructure Library, Office of Government Commerce (UK)," http://www.itil.co.uk.

[2] W. Hommel and S. Knittl, "An Inter–Organizational Configuration Management Database as Key Enabler for Future IT Service Management Processes ," in *eChallenges 2008*, vol. 2008, Stockholm, Schweden, Oct. 2008.

[3] E. Apted, J. Chevers, M. G. Vidondo, and S. Tyley, "Report on GÉANT2 Advanced Services - Lambdas and Switched Optical," GN2 Project, Tech. Rep. Deliverable GN2-08-224v3, Feb. 2009.

[4] W. M. P. V. D. Aalst, "Process-oriented architectures for electronic commerce and interorganizational workflow," *Information Systems*, vol. 24, no. 9, pp. 639 – 671, December 1999.

[5] G. Hedlund, "Assumptions of hierarchy and heterarchy, with applications to the management of the multinational corporation," in *Organizational Theory and the Multinational Corporation*, second edition ed., S. Ghoshal and E. Westney, Eds., London, 2005, pp. 198–221.

[6] M. Hamm, P. Marcu, and M. Yampolskiy, "Beyond hierarchy: Towards a service model supporting new sourcing strategies for it services," in *Proceedings of the 2008 Workshop of HP Software University Association (HP-SUA), Infonomics-Consulting, Hewlett-Packard*, Marrakech, Morocco, June 2008.

[7] P. Marcu, L. Shwartz, and G. Grabarnik, "Model for Incident Ticket Correlation for Inter-Organizational Service Delivery," in *eChallenges 2009*, vol. 2009, Istanbul, Turkey, Oct. 2009, to apear.

[8] P. Marcu, G. Grabarnik, L. Luan, D. Rosu, L. Shwartz, and C. Ward, "Towards an Optimized Model of Incident Ticket Correlation," in *Proceedings of the 11th IFIP/IEEE Symposium on Integrated Management (IM 2009), IEEE Computer Society Press*, New York, USA, June 2009.

[9] M. Brodie, I. Rish, S. Ma, G. Grabarnik, and N. Odintsova, "Active probing," *IBM Technical Report RC22817*, 2002.

[10] M. Langer and M. Nerb, "Defining a Trouble Report Format for the Seamless Integration of Problem Management into Customer Service Management," in *Proceedings of the 6th International Workshop of the HP OpenView University Association (HPOVUA99)*. Bologna, Italy: HPOVUA, June 1999.

[11] S. Tai, N. Desai, and P. Mazzoleni, "Service communities: applications and middleware," in *Proceedings of the 6th International Workshop on Software Engineering and Middleware, SEM*, Portland, Oregon,USA, nov 2006, pp. 17–22.

[12] Y. B. Udupi and M. P. Singh, "Governance of cross-organizational service agreements:a policy-based approach," in *Proceedings of the IEEE International Conference on Services Computing (SCC 2007, volume II)*, Salt Lake City, UT,USA, july 2007, pp. 36–43.

[13] G. Grabarnik, H. Ludwig, and L. Shwartz, "Management of service process QoS in a service provider - service supplier environment," in *Proc. CEC/EEE-2007*, Tokyo, Japan, July 2007, pp. 543 – 550.