An information model for inter-organizational fault management

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

Thomas Schaaf
Munich Network Management Team
c/o Ludwig-Maximilians-Universität München
Oettingenstr. 67, 80538 Munich, Germany
schaaf@mnm-team.org

***Abstract*—IT service providers outsourcing (parts of) their IT services often have to face the side-effect of losing control over service delivery and support, and thus over the service quality. To respond to this problem, guidance on delivering IT services in inter-organizational environments is needed.**

One of the most important disciplines in this context is fault management. This paper presents the fundamentals of an information model for inter-organizational fault management and shows how this information model is an important component of a comprehensive management architecture for inter-organizational fault management.

I. Introduction

Providing IT services in an inter-organizational environment is a complex and often error-prone challenge. Fault management – as one of the five typical functional management areas (FCAPS) according to the OSI management architecture [1] – aims at restoring normal service operation after an incident occurred; in addition to this reactive task, fault management is about proactively preventing incidents from occurring (or recurring) in order to avoid incidents and their associated loss or damage to the business.

A. Motivation

The above mentioned notion of fault management corresponds to the core objectives of incident management and problem management as described in the IT Infrastructure Library (ITIL) [2] and related frameworks in the area of IT service management (ITSM). But since ITIL and the related ITSM frameworks do not consider specific aspects of inter-organizational (fault) management, they will hardly provide any guidance in this regard. The specific problems of inter-organizational fault management (ioFM) are elaborated and explained in detail in [3] and include:

- the outsourcing problem
- the problem of heterogeneity and autonomy
- the problem of service delivery diversity

B. Approach

To cope with these challenges, an architecture for inter-organizational fault management (ioFMA) is required that addresses all relevant aspects of ioFM in sufficient detail to provide guidance on organizational and technical implementation, i.e. setting up processes/procedures as well as designing or adapting tools/technology. The ioFMA consists of four key models:

1) the organizational model revealing roles and responsibilities required to conduct efficient ioFM,
2) the functional model describing the most important management functionalities,
3) the information model specifying all relevant information objects, and
4) the communication model delivering procedures for information exchange between entities.

The focus of this paper is on the information model which is often regarded as the backbone of a management architecture. In the next section, we will give a brief overview of the core requirements on this part of the ioFMA. Section 3 will then present some core elements of the information model. Related work in the areas of management architectures, fault management in general as well as on information/data models will be presented in Section 4. Section 5 concludes the paper with a short summary and outlook.

II. Requirements

The goal is to develop an information model for ioFM. Starting point for the requirements analysis is the fault life cycle consisting of the following life cycle phases: detection, isolation, repairing/recovery, and forecast/prevention. In [4], requirements for each of the four sub models of the ioFMA are identified and mapped to this life cycle. With respect to the information model, the following requirements have been derived and are relevant for the scope of this paper:

- **IM-01**: A ***common data format*** for fault information is needed in order to facilitate the inter-domain data exchange and the communication. This should consist of a set of common attributes or properties.
- **IM-02**: Another additional or coexisting requirement to the first one is the existence of ***conversion methods*** between the data format in the different domains.
- **IM-03**: ***Interfaces*** across different domains have to be defined.
- **IM-04**: The ioFMA has to support all the ***life cycle phases*** of a fault resolution process (detection, isolation, repairing/recovery, and forecast/prevention).

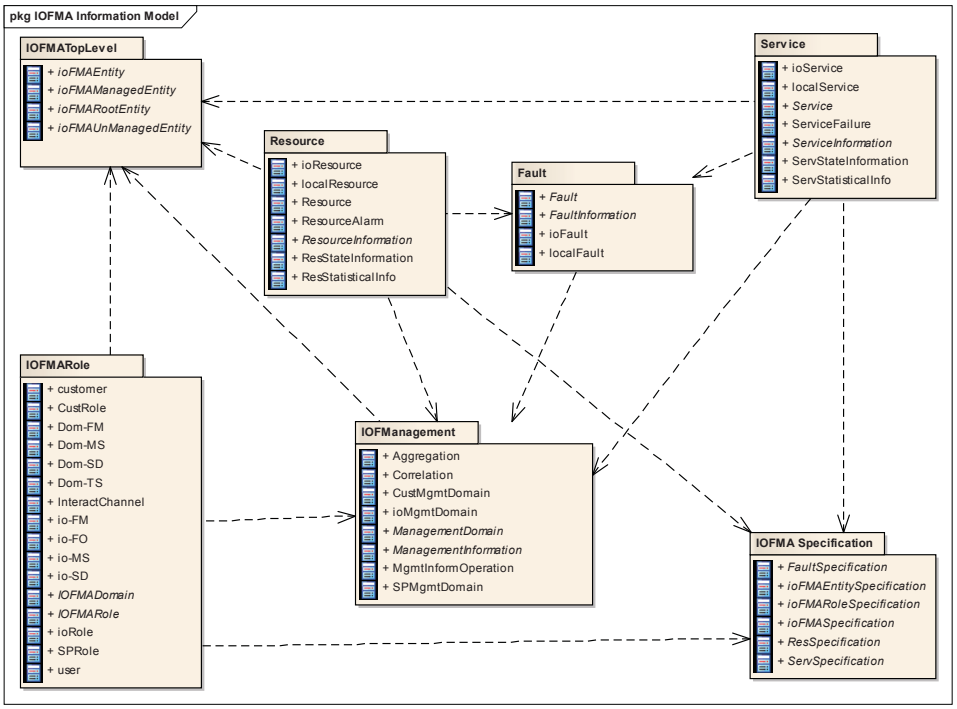

Fig. 1. Overview on the **ioFMA** information model

- **IM-05**: Also, the use of ***standard metrics*** has a supporting role in monitoring, and respectively in reporting. An example of such a set of standard metrics is the IP Performance Metrics (IPPM) [5] (e.g., One Way Delay (OWD [6]), IP Delay Variation ([7]), Packet Loss ([8]), and others).
- **IM-06**: As the correlation/interrelation between the metrics of different domains has to be provided, a suitable ***aggregation function*** has to be defined.

These requirements have been elicited from two real world scenarios: the IntegraTUM scenario as a representative example of hierarchical inter-organizational service delivery, and the GEANT scenario featuring heterarchical service delivery in inter-organizational environments. Use cases have been described serving as the main source of functional requirements (see [4]).

III. Fundamentals of an Information Model for ioFM

As stated before the information model is the core of each management architecture. It provides a common information base for all involved components in order to sustain the interaction respectively the information interchange between different components in their management tasks.

In our approach, the information model consists of separate domains grouping the covered entities in a way that entities related to the same topic area are assigned to the same domain. Thus, an **ioFMA** domain contains entities which are attributed to specific, conceptually closed areas of the ioFM. The **Role** domain is based on the organizational model (see [3]). It describes the diferent organizational domains, roles and interaction channels between the roles involved in the ioFM.

The managed objects of the **ioFMA** MIB are the resources, services and faults in interorganizational environments. Therefore, three model domains **Resource**, **Service** and **Fault** are defined. Within the **Resource** domain, a specialization of the resources into intra-organizational and inter-organizational resources is needed. In addition, this domain covers resource alarms as well as information on the state of a resource. Analogous, the **Service** domain provides an identification of the service, the way it is delivered (intra-organizational or inter-organizational), a description of possible problems with this service as well as service information. The **Fault** domain integrates additional information concerning the specialization of the fault (intra-organizational or inter-organizational) as well as service failure and resource alarm related information.

Due to space restrictions, this paper is focused on presenting the model of the **Fault** domain.

As the specifications of the **ioFMA** entities (resources, services, faults, roles, etc.) are a vital part of the ioFM, they are grouped in a separate domain (**IOFMASpecification**). Fault management related information are covered by a specific domain named **IOFManagement**.

In order to define a generalized set of ioFM entities, a comprehensive domain is introduced. The domain **IOFMATopLevel** contains root entities and abstract superclasses of the **ioFMA** information model in order to generate all entities involved.

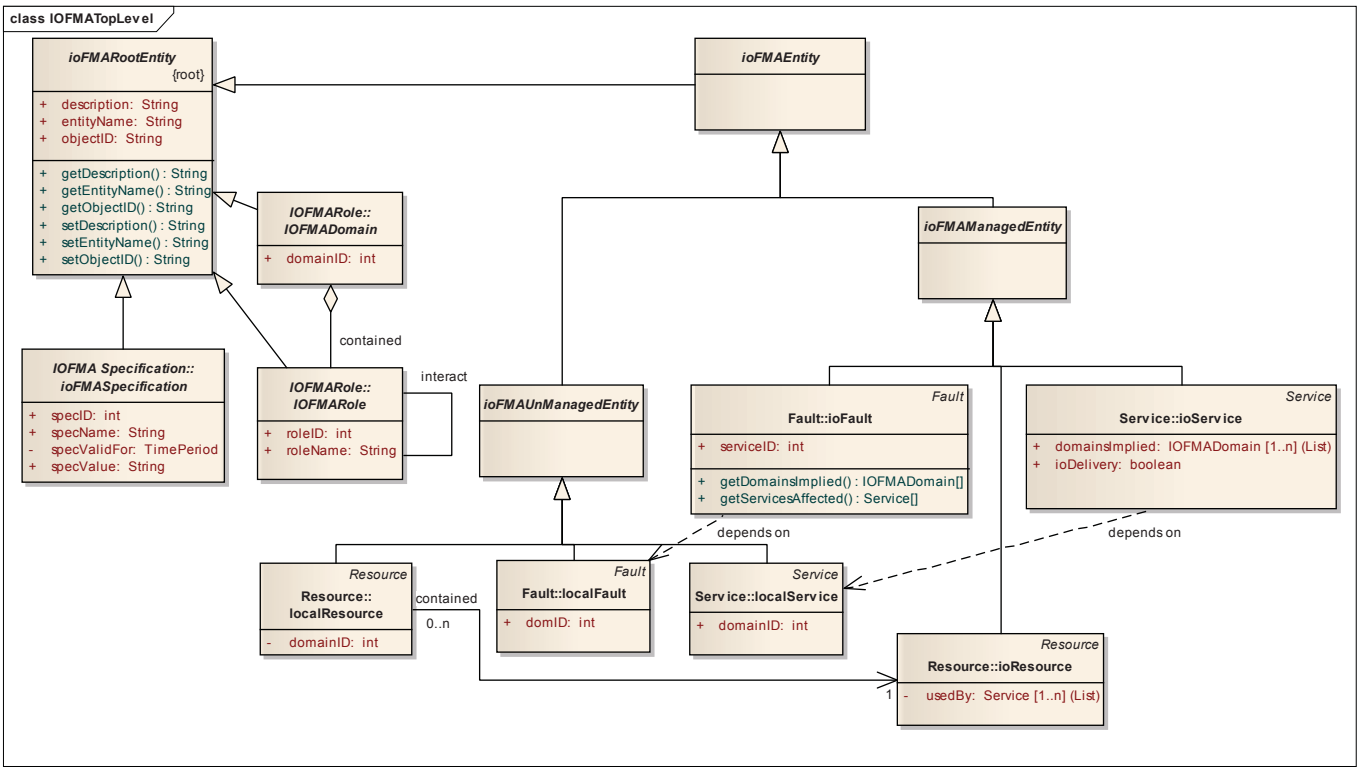

Fig. 2. The `IOFMATopLevel` Domain

A. Overview

The information model specifies a framework for the description of the entities involved in the management process (the managed objects, MOs). For the development and specification of the classes and relationships of **ioFMA** the Unified Modeling Language (UML) will be used.

The building blocks of the **ioFMA** information model are as stated above domains representing groups of entities involved in ioFM. The domain `IOFMATopLevel` contains the common top classes or root entity classes. The domain `IOFManagement` is modeling ioFM with respect to management information. Related to this, the domains `Service`, `Resource` and `Fault` are introduced in order to provide data models for the managed objects of the **ioFMA**. Roles and administrative domains (actually part of the organizational model) can be defined using the domain `IOFMARole`. `IOFMASpecification` is an overview domain concerned with the specification of managed objects. Figure 1 gives an overview on the domains involved in the **ioFMA**-information model.

B. TopLevel domain

The `IOFMATopLevel` domain focuses on the common set of entities, as a basis for the **ioFMA** information model. Figure 2 shows the domain `IOFMATopLevel` with its abstract and generic root entity classes.

- `ioFMARootEntity` is the root class of the information model class hierarchy. A set of attributes and methods common to all **ioFMA** entities are defined. The attributes `objectID`, `entityName` and `description` are used for the identification, naming and description of the entities, and the `get` and `set` methods are required to read and set their values. This class is inherited by other four other classes: `ioFMAEntity`, `IOFMARole`, `IOFMADomain` and `ioFMASpecification`.
- `ioFMAEntity` is the abstract class from which all managed and unmanaged objects of the **ioFMA** derive through specialization in the classes `ioFMAManagedEntity` and `ioFMAUnManagedEntity`. In this context the managed entities are defined as objects which are critical for the ioFM, though they are directly involved in the management process. On the other hand, the unmanaged entities are those objects that are not manageable, so they cannot create management information by themselves, but can be associated to managed entities.
- The managed entities are `ioService`, `ioResource` and `ioFault`, representing the following inter-organizational entities in the **ioFMA**: services, resources and faults.
- The unmanaged entities of the **ioFMA** defining the local entities are represented by the classes `localService`, `localResource` and `localFault`.

For the fault-related classes a description will be given in the next subsection. The ”local” classes always have an additional attribute `domID` denoting the administrative domain

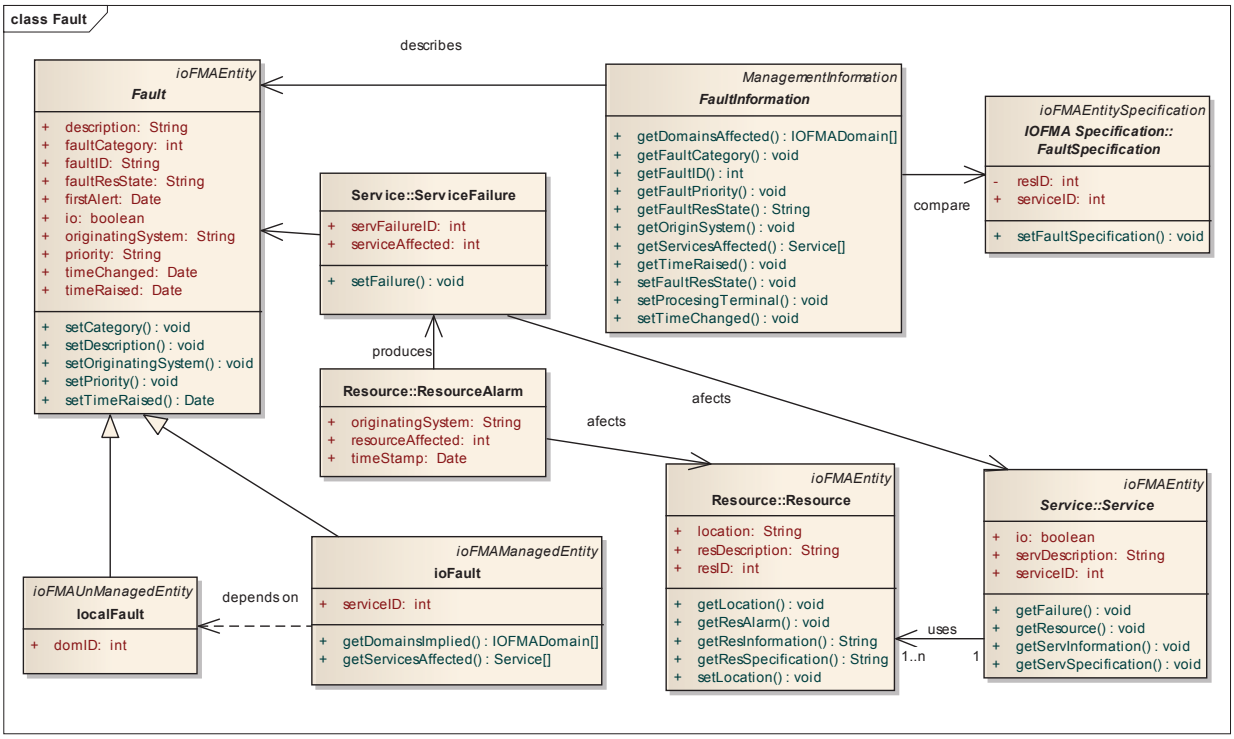

Fig. 3. The `Fault` Domain

they belong to. The `ioResource` class also has an additional multiple attribute `usedBy` representing a list of services that depend on this resource. The `ioService` class has got two more attributes `domainsImplied` as a list of IOFMADomains involved in the service delivery, and `ioDelivery` showing the type of service delivery (hierarchical or heterarchical).

- The class `IOFMARole` is the root class for all defined roles of the **ioFMA**. Aggregated they will result in the class `IOFMADomain` representing the administrative domain of the **ioFMA**. These are parts of the `IOFMARole` domain.
- The entire group of entities specifications is based on the class `ioFMASpecification`. A few more attributes are defined as part of this abstract class: `specID` (ID of the specification), `specName` (name of the specification), `specValidFor` (period of time in which the specification is valid) and `specValue` (specification content).

C. Fault domain

This domain contains all fault-related classes of the **ioFMA** and provides the core class `Fault`.

- `Fault` is a specialization of the abstract class `ioFMAEntity`. Some attributes define this class: `faultID`, `description`, `faultCategory`, `priority`, `originatingSystem`, `faultResState` and `io` representing the unique identifier, description, category, priority, originating system of the fault as well as the status/progress of fault resolution. In addition, three time-related attributes are available: `firstAlert`, `timeRaised`, and `timeChanged` which denote the time stamps of the first alarm, of the first fault notification as well as of the last change.
- `localFault` and `ioFault` (specialization of `Fault`) represent the classes of local and inter-organizational faults. The class `ioFault` has an additional attribute `serviceID` which references the affected inter-organizational service. The methods *getDomainsAffected()* and *getServicesAffected()* return the affected domains and services.
- The `FaultInformation` class is responsible for getting or setting (management) information for the `Fault` class. The methods *getFaultID()*, *getFaultCategory()*, *getFaultPriority()*, *getTimeRaised()* and *getOriginSystem()* are used in order to get the ID, category and priority of the fault as well as the time the fault raises at the originating system. The methods *getDomainsAffected()*, *getServicesAffected()* return lists of the affected domains respective services. *getFaultResState()* and *setFaultResState()* get and set the status of the fault resolution process. *setProcessingTerminal()* and *setTimeChanged()* set the responsible terminal for the fault resolution and the time stamp when the status was changed. This is part of the `IOFManagement` domain.
- The `FaultSpecification` class specifies possible faults that may occur in the regarded environments. This specification is made based on historical data.

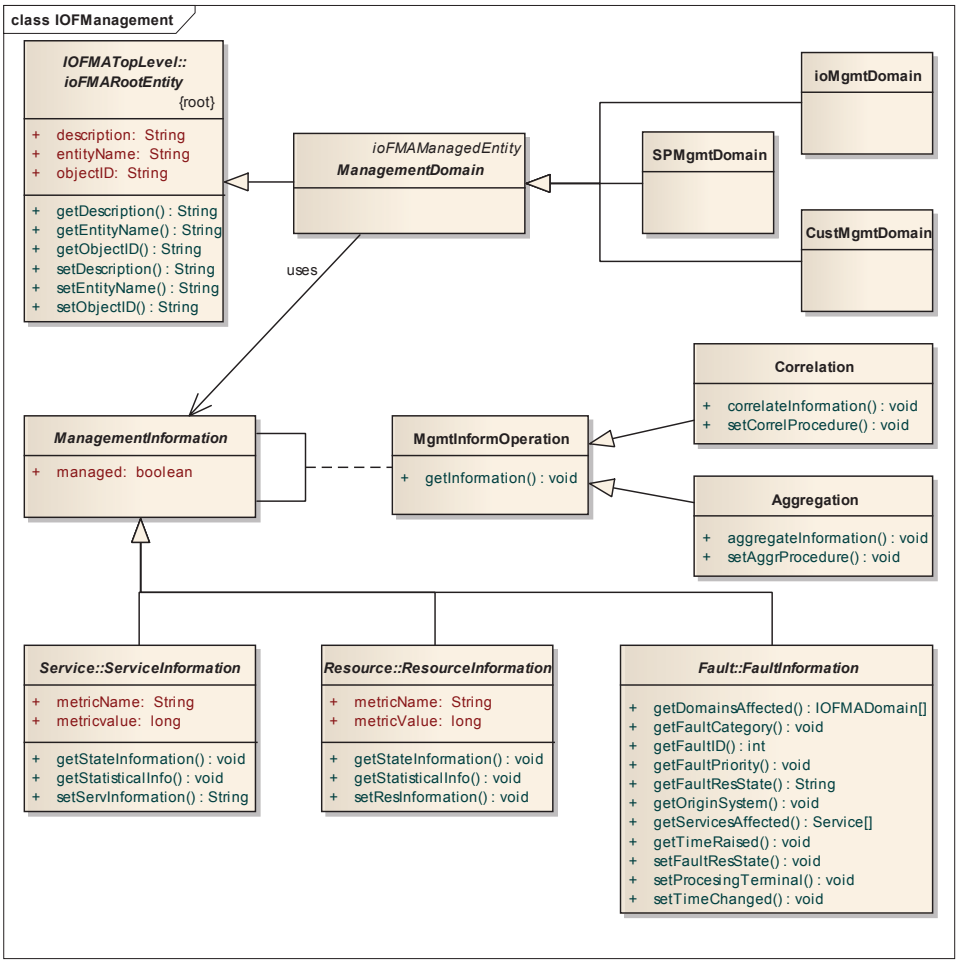

Fig. 4. Die Domne `IOFManagement`

The `resID` and `serviceID` attributes represent the resource or service which could be affected by this fault. It is also accessed by the class `FaultInformation` in order to *compare* the actual information with a possible fault specification.

- A `ResourceAlarm` (connected to a resource) *produces* a `ServiceFailure` (afecting a service) and determines a `Fault`. The resource- and service-related classes are part of the `Service` respectively `Resource` domains.

D. IOFManagement domain

The `IOFManagement` includes fault-management-related issues.

- The abstract class `ManagementDomain` is a specialization of the `ioFMARootEntity` class described in the `IOFMATopLevel` domain. Furthermore, three more classes inherit from this abstract class: `CustMgmtDomain`, `SPMgmtDomain`, and `ioMgmtDomain`. These classes actually represent a part of the functional model of the architecture with its functional areas: inter-organizational management (core part of the **ioFMA**), customer management, and service provider management (without whom no ioFM could be performed at all).
- The `ManagementDomain` class uses the `ManagementInformation` class for providing management information. The `managed` attribute shows whether the information-belonging entity is a managed or unmanaged entity. The management information is classified into `ServiceInformation`, `ResourceInformation`, and `FaultInformation` according to the categories of the ioFMA entities.
- The `ServiceInformation` class characterizes the service with additional information relevant for fault management. It is also part of the `Service` domain. The attributes used here are `metricName` and `metricValue` representing the metric name and the metric value of the management information. The three methods used here are *getStateInformation()* und *getStatisticalInfo()* to get information concerning the status respectively of the statistical metrics as well as *setStateInformation()* to set these information accordingly.
- The `ResourceInformation` class represents the

management information assigned to a resource. It has got two attributes and three methods analog to the `ServiceInformation` class.
- The `FaultInformation` class was already described as part of the `Fault` domain.
- More management information may be combined with each other. This is realized with the help of association classes. Thus, the `MgmtInformOperation` association class is defined. It comes with a *getInformation()* method to gather information from the classes `ServiceInformation`, `ResourceInformation`, and `FaultInformation`. These information will be combined according to the operation classes.
- Two operation classes are defined: `Correlation` and `Aggregation`, each specializing the `MgmtInformOperation` class. The `Correlation` class comes with two methods: *setCorrelProcedure()* to set the correlation algorithm and *correlateInformation()* which actually performs the information correlation. A possible correlation mechanism is described in [9]. The second class (`Aggregation`) provides methods for setting the aggregation algorithm or procedure (*setAggrProcedure*) and performing the actual aggregation of the given information. If necessary, more operation classes can be defined according to the requirements.

E. Further components

In order to achieve a comprehensive information model for the **ioFMA**, the folowing domains have been refined in a similar way as presented above for the `IOFMATopLevel` domain: `Service`, `Resource`, `IOFMARole`, and `IOFMASpecification`. Due to space limitations, the results of these refinements and specifications are not included in this paper.

IV. Related Work

More areas of interest concerning IT-management-centered aktivities have evolved in the last years. With respect to the **ioFMA**, three categories of approaches are significant: management architectures and their methodological development, ITSM and fault management processes, and information/data models.

A. Management architectures

The fundamentals of management architectures are described in [1]. The primary goal of each management architecture is to establish an *integrated* management framework instead of using several loosely coupled components or tools more or less independently from each other. The **ioFMA** is a management architecture based on this idea.

In [10] and [11], a tool-based management system for supporting Service Level Management tasks is presented. For the realization an MDA-like approach was selected to provide a platform-independent as well as a platform-specific model. Prior to the development of these models, requirements on the management architecture have been elicited from scenario-driven use cases. The developed management architecture was partitioned in four sub models according to [1]. Each of the sub models is examined from two different view points: the process view and the system view. This work accomplishes a construction kit of concrete functional requirements, use cases for a management system, process-related models, and system models with regard to software develeopment and automation aspects. The **ioFMA** is based on a comparable systematic approach, as highlighted earlier.

A similar MDA-based methodology was used in [12] to develop a management architecture for virtual organizations (VO). This work provides a sound basis for the inter-organizational-related development of the **ioFMA**, as VO management exceeds organizational boundaries, too.

B. IT service management/Fault management processes

In the ITSM domain, various frameworks describe management processes and principles. A popular and widespread framework is the IT Infrastructure Library (ITIL) whose third revision has been published in 2007. Incident and problem management are two of ITIL's core reference processes [2] dealing with faults. Besides ITIL, other frameworks or framework-like approaches provide relevant guidance on fault management (excerpt):

- ISO/IEC 20000 [13], [14]: an international standard for ITSM, successor of the BS 15000 standard which was based on ITIL
- Microsoft Operations Framework (MOF) [15]: operational guidance on ITSM including job aids (e.g., document templates), originally based on ITIL
- Enhanced Telecom Operations Map (eTOM): management guidelines, processes and data models especially for telecommunication providers, focused on, but not limited to IT-related management tasks.

Most ITSM frameworks provide general high-level process descriptions for fault-management-related task. For the development of the **ioFMA**, the incident and problem management processes described in [2] respectively in [13] were regarded as the computation independent model, and thus served as basis for the development of the platform independent and platform specific models.

C. Information/Data Models

Another topic of interest is given by different information models from practice and research:

- Common Information Model (CIM) [16] is an object-oriented management information model that aims at providing a common way to represent information about networks and systems as well as services. It defines managed resources as object classes that can be further refined by means of strict inheritance. The Managed Object Format (MOF) as part of CIM is used to describe modeling constructs which can then be processed by automated tools. CIM provides a coherent view on the modeled infrastructure as all CIM classes derive from the

managed element class (core model). However, a linkage of the fault management processes to infrastructure elements is not given.

- Shared Information/Data Model (SID) as an integral part of the NGOSS (New Generation Operations Systems and Software) initiative by TM Forum (Tele Management Forum) [17], [18] provides an object-oriented information model based on CIM and eTOM. It provides a clear distinction between the system and business view on management information. With entities and attributes being described by a mixture of descriptive text, UML diagrams and tables, SID also provides a reasonable level of expressiveness.
- In the last few years a new inintiative of the TMF came into beeing: TM Forum Interface Program (TIP). It aims to connect the multitude of support systems, multi-technology networks, and data that exist in this IT/management environment through open, flexible interfaces. One of these TIPs is the Service Problem Management Information Agreement (SPMIA) [19] developed to support the problem or trouble area of the Business Process Framework of eTOM.

Our appoach is aligned with the core SID principels; in addition, the interfaces given in SPMIA are considered, too.

V. Conclusions and further work

In this paper, we presented some core components of the **ioFMA** information model. This information model was developed following an MDA approach. We gave a brief overview of the fundamental requirements on this information model, followed by a presentation of the (`IOFMATopLevel`, `Fault`, and `IOFManagement`) classes of the information model.

The architecture for inter-organizational fault management **ioFMA** is currently under development, and thus, the information model (including the components presented in this paper) may be subject to changes and enhancements. In the near future, the full architecture will be completed including all four sub models which are – besides the information model – the organizational model, the functional model, and the communication model. The application of the **ioFMA** to a real-world scenario including a final evaluation will be the next logical step.

Acknowledgments

This work was in parts supported by the EC FP7 coordination and support action gSLM (Service Delivery and Service Level Management in Grid Infrastructures).

The authors wish to thank the members of the Munich Network Management Team (MNM-Team) for helpful discussions and valuable comments on previous versions of this paper. The MNM Team directed by Prof. Dr. Dieter Kranzlmüller and Prof. Dr. Heinz-Gerd Hegering is a group of researchers at Ludwig-Maximilians-Universität München, Technische Universität München, the University of the Federal Armed Forces and the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences and Humanities. See http://www.mnm-team.org/.

References

[1] H.-G. Hegering, S. Abeck, and B. Neumair, *Integrated Management of Networked Systems - Concepts, Architectures and their Operational Application*. Morgan Kaufmann Publishers, 1999.

[2] OGC, Ed., *Service Operation*, ser. IT Infrastructure Library v3 (ITIL v3). Norwich, UK: The Stationary Office, 2007.

[3] P. Marcu and W. Hommel, "Inter–organizational fault management: Functional and organizational core aspects of management architectures," *International journal of Computer Networks & Communications (IJCNC)*, vol. 1, Jan. 2011.

[4] P. Marcu and W. Hommel, "Requirements and concepts for an inter–organizational fault management architecture," in *Proceedings of the 9th RoEduNet International Conference*. Sibiu (Hermannstadt), Romania: IEEE Computer Society, Jun. 2010.

[5] "IP Performance Metrics Working Group." [Online]. Available: http://tools.ietf.org/wg/ippm/

[6] Almes, G. and Kalidindi, S. and Zekauskas, M., "A One-way Delay Metric for IPPM," USA, Tech. Rep., 1999.

[7] Demichelis, C. and Chimento, P., "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)," USA, Tech. Rep., 2002.

[8] Almes, G. and Kalidindi, S. and Zekauskas, M., "A One-way Packet Loss Metric for IPPM," USA, Tech. Rep., 1999.

[9] P. Marcu, L. Shwartz, G. Grabarnik, and D. Loewenstern, "Managing Faults in the Service Delivery Process of Service Provider Coalitions," in *IEEE International Conference on Service Computing (SCC 2009)*, Bangalore, India, 2009.

[10] M. Brenner and T. Schaaf, "On Tool Support for Service Level Management: From Requirements to System Specifications," in *Proceedings of 3rd IEEE/IFIP International Workshop on Business–driven IT Management*, vol. 2008, Salvador, Brasilien, Apr. 2008.

[11] T. Schaaf, "IT–gestütztes Service–Level–Management — Anforderungen und Spezifikation einer Managementarchitektur," Dissertation, Ludwig–Maximilians–Universitt Mnchen, Dec. 2008. [Online]. Available: http://d-nb.info/992790255

[12] M. Schiffers, "Management dynamischer virtueller organisationen in grids," Dissertation, Ludwig–Maximilians–Universitt Mnchen, Jul. 2007.

[13] "ISO/IEC 20000-1:2005 - Information Technology - Service Management - Part 1: Specification," International Organization for Standardization, Tech. Rep., Dec. 2005.

[14] "ISO/IEC 20000-2:2005 - Information Technology - Service Management - Part 2: Code of Practice," International Organization for Standardization, Tech. Rep., Dec. 2005.

[15] "Microsoft Operations Framework – Version 4.0," Microsoft Corporation, Tech. Rep., 2008. [Online]. Available: www.microsoft.com/MOF

[16] "Common Information Model (CIM) Version 2.27.0," Distributed Management Task Force, Tech. Rep., Nov. 2010. [Online]. Available: http://dmtf.org/standards/cim

[17] "Shared Information/Data (SID) Model, Addendum 4S0 - Service Overview Business Entity Definitions," TeleManagement Forum, NGOSS Release 9.0, Oct. 2010.

[18] "Shared Information/Data (SID) Model, Addendum 1R – Common Business Entity Definitions – RootBusiness Entities," TeleManagement Forum, NGOSS Release 9.0, Oct. 2010.

[19] "TM Forum Interface Program (TIP) – Service Problem Management Information Agreement (SPM IA)," TeleManagement Forum, TIP Release 1.0, Sep. 2010.