17th Int. Conf. on IFSs, Sofia, 1–2 November 2013
Notes on Intuitionistic Fuzzy Sets
Vol. 19, 2013, No. 3, 62–72

Intuitionistic fuzzy component failure impact analysis (IFCFIA) – A gradual method for SLA dependency mapping and bi-polar impact assessment

Roland Schütze

University of Fribourg, Switzerland
e-mail: roland.schuetze@unifr.ch

Abstract: This concept provides a bridge from IT-centric service levels, written in IT technical terms, to business-oriented service achievement. The proposed IFCFIA methodology will help for Service Level Agreements (SLAs) to relate metrics for business applications into measurable parameters for technical services that can be defined and reported against a SLA and monitored under Service Level Management. It allows assessing the complex dependency and impact relationships of low-level backend components to the quality of the frontend service. This work defines dependency couplings in a practical and feasible manner in order to satisfy aspects of the distributed nature of SLAs in a multi-tier-architectural environment and offers transparency into complex impact assessments. IFCFIA starts from the idea of naturally approaching impact relationships by separately envisaging positive and negative aspects with the notion of bipolarity. Performing an intelligent multi-level impact- or fault-tree analysis by means of intuitionistic fuzzy mathematical models it unveils business insights into how service accounts as a whole can improve quality and allows pro-actively tracking measures of backend components to gather the overall SLA quality status of the business service.
Keywords: Service Level, SLA, Business Impact, Services Quality, Intuitionistic Fuzzy Sets.
AMS Classification: 03E72, 03E75

1 The complexity of multi-layered Service Level Requirements

In an increasingly service-oriented world, “best effort” service delivery is not good enough. But how does the business know whether it is getting an adequate service? Service level requirements are set to ensure that the business goals underlying IT services are met. The Service Level Agreements (SLAs) incorporate the expectations and the obligations about the properties of a service. [1],[2]. The most significant part of a SLA is the range of the duties of a service. The SLA objectives are mostly the concerns that are associated with the Quality of a

Service (QoS). To guarantee business-focused SLAs results in optimization problem solving across multiple domains (e.g. networking, computer systems, and software engineering). The landscape of today's IT service providers is inherently integrated. It consists of all kinds of elements, namely networks, servers, storage, and software stacks. The fulfilment of any higher-level objective requires proper enforcements on multiple resources at several levels.

The challenge with such enterprise SLAs is translating metrics for business applications into measurable parameters for technical services that can be defined and reported against an SLA and monitored under Service Level Management (SLM). Service compositions, translation and mappings lies therefore in the core of SLA management, in that it correlates metrics and parameters within and across layers [3]. For example, in order to guarantee certain bounds on the response times for ERP-type, it involves the ERP software, the application and database servers, the network configuration, and more [4]. When knowing the relation and dependency of this backend service to the end-user service (or composite service), service administrators can then pro-actively track and verify these dependencies by periodically polling the measures of individual services and gathering the overall quality status of the end-user service. This will allow administrators responsible for the functioning of a service to monitor its quality based on the measurements typically already done for the infrastructure components.

2 SLA dependency mapping

2.1 The concept of Key Quality and Performance Indicators

Open Group [5] defined the concept of key quality– and performance indicators (KQI/PI). Service Level Specification parameters can be one of two types: Key Quality Indicators (KQIs) and (most technical) Service Performance Indicators (PIs). At the highest level, a KQI or group of KQIs are required to monitor the quality of the business service offered to the end-user. These KQIs will often form part of the contractual SLA. The KQI is derived from a number of sources, including performance metrics of the service or underlying support services with PIs. Different PIs may be assembled to calculate a particular KQI [6].

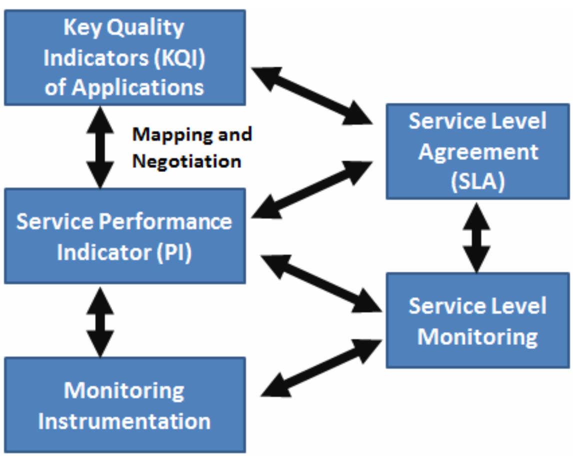

Figure 1. KQI , PI & SLA relationship [5]

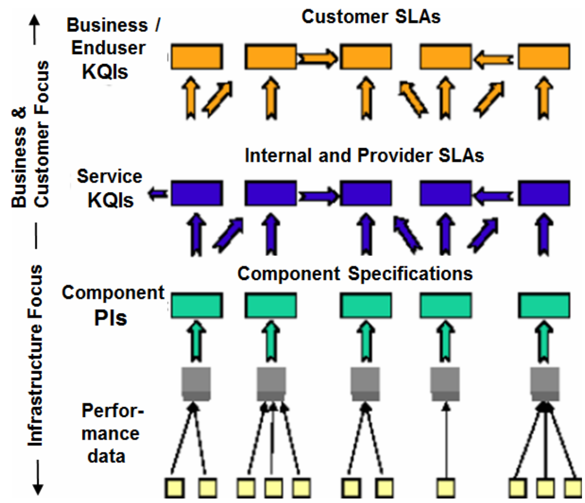

Figure 2. KQI/PI indicator hierarchy

The mapping between the PI and KQI may be simple or complex, empirical or formal. The automated process of translating and correlating high-level requirements and policies of all kinds down to infrastructure level creates a set of related PIs, which is termed now a KQI/PI hierarchy. While the association relationship only relates adjacent sets of KQIs/PIs, the hierarchy establishes associations across the whole stack in a distributed multi-tier architecture. In the following a Coupling *C* association is defined, which can be constructed in a practical and feasible manner in order to satisfy aspects of the different types of interdependencies.

2.2 Dependence coupling as measurement

Dependence Coupling is a measure that we propose to capture how dependent the component or service is on other services or resources for its delivery. In general the goal is to build components that do not have tight dependencies on each other, so that if one component were to die (fail), sleep (not respond) or remain busy (slow to respond) for some reason, the other components in the system are built so as to continue to work as if no failure is happening. Loose coupling describes an approach where integration interfaces are developed with minimum assumptions between the sending/receiving parties, thus reducing the risk that change or failure in one module will effect to others. Loose coupling isolates the components of an application so that each component interacts asynchronously with the others and treats them as a “black box”. For example, in the case of web application architecture, the application server can be isolated from the web server and from the database.

Two new types of a logical relationship are now introduced which expresses the level of inter-dependency between components: ‘is tightly coupled’ and ‘is loosely coupled’. The tightly coupled measurement can be seen as an indicator of the risk resulting from interdependencies where the loosely coupled aspect refers to the mitigation and resilience capabilities of a service. Loose coupling indicates that the service does not have to depend on other services or resources to complete delivery of its service. Tight coupling on the other hand indicates that successful delivery of other services or availability of resources is a prerequisite for the completion of a service. When the dependency is between a service and some resource it uses, coupling will essentially be a function of how often the resource is used. For instance, the dependence of a service on the network layer might be measured by how often it is making a socket call, or how much data it is transferring. The dependence of a database on compute partition will be determined by how much compute resources it needs from that partition. For web-services we can examine environmental coupling which is caused by calling and being called. Traditional components are more tightly and statically integrated and measurements are related mostly to procedural programming languages [7, 8]. More advanced are object-oriented coupling measures [9] and further several metrics are proposed to evaluate the coupling level real-time by runtime monitoring, introduced as dynamic coupling metrics [10].

2.3 Bi-polar coupling aspects

A key principle of the following proposed impact assessment method is the idea of naturally envisaging positive and negative instances of the dependency relation and simultaneous consideration by pulling both strengths together. For a complex IT system the risk are the dependencies through interactions, the controversy mitigation ability are the built-in system resilience capabilities. The simultaneous and free play of contrary forces, dependence and resilience together will define the overall system behaviour and the expected impact to the business. Considering and judging positive and negative aspects isolated will not lead to reliable assessments. This leads to the question whether traditional impact analysis methods can be applied for such integrated model. In general the ITIL v3 methods already cover both aspects [11]. Fault Tree Analysis (FTA), like the word fault tree indicates, work in the "failure space" and looks at system failure combinations. So the FTA method covers the aspect of negative risk of interdependencies and negative impacts on failure. On the other side, the ITIL Component Failure Impact Analysis (CFIA) approach [12] is assessing on the mitigation, restoration and resilience capabilities, which represents the positive aspect of independence.

Further the intelligence in any complex system analysis will be the modelling of the indirect dependencies and interactions. There are several scenarios how an incident may interfere indirectly with other components which is mainly resulting out of the combination of the contrary forces. IT systems try to implement strategies that the resilience capabilities of each component should pro-actively limit the inference and impact of the incident to related components or the business services. In praxis impacts are complex which constitutes uncertainty. They involve a multitude of effects that cannot be easily assessed and may involve complex causalities, non-linear relationships as well as interactions between effects [13]. This may render it difficult to determine exactly what may happen. Thus we propose to consider also a level of vagueness, uncertainty and limited or imprecise knowledge as meta-information for assessed degrees of the coupling relationships.

3 Applying the model of intuitionistic fuzzy sets

Let E be a fixed universe and A is a subset of E. The set $A^* = \{(x, \mu_A(x), \nu_A(x)) \mid x \in E\}$, where $0 \leq \mu_A(x) + \nu_A(x) \leq 1$ is called Intuitionistic Fuzzy Set (IFS) [14]. Every element has a degree of membership (validity, etc.) $\mu_A(x)$: $E \rightarrow [0, 1]$ and a degree of non-membership (non-validity, etc.) $\nu_A(x)$: $E \rightarrow [0, 1]$. Intuitionistic Fuzzy Sets have only loosely related membership and non-membership values unlike classical [15] fuzzy sets. An IFS is a generalization of the classical fuzzy set which defines another degree of freedom into the set description, the independent judgment of validity and non-validity. This two-sided view, including the possibility to represent formally also a third aspect of imperfect knowledge could be used to describe many real-world problems in a more adequate way – by independent rating of both, positive and negative aspects – for each variable in the model. For each IFS A in E, $\pi(x) = 1 - \mu_A(x) - \nu_A(x)$ is called the intuitionistic index of x in A which represents the third aspect, the degree of uncertainty, indeterminacy, limited knowledge etc.

In the following approach, let now a be the intuitionistic fuzzy logical statement of tightly coupling and b of loosely coupling with estimations respectively $\langle \mu_a, \nu_a \rangle$ and $\langle \mu_b, \nu_b \rangle$. The tightly coupling degree of truth is $\langle \mu_a \rangle$ and the degree of falsity $\langle \nu_a \rangle$. The same assessment is done for loosely coupling b where $\langle \mu_b, \nu_b \rangle$ represent the degrees of truth and falsity.

The validities and non-validities for tightly and loosely couplings are independently estimated by separate approaches, means for 'tightly' using the described inter-modular coupling measurements and for 'loosely' via the assessed intrinsic component resilience capabilities.

To define now the direct Coupling C association between two components the intuitionistic fuzzy logical statements of tightly coupling and loosely coupling are pulled together. Here several operations over IFS are possible. As tightly and loosely couplings have contrary effects a meaningful operation for building the combined IFS C is for instance $A@\neg B$ where

$$\mu_{combined}(x) = \frac{\mu_A(x) + \nu_B(x)}{2} \quad \text{and} \quad \nu_{combined}(x) = \frac{\nu_A(x) + \mu_B(x)}{2}$$

by adding membership 'tightly' with non-membership 'loosely' and vice versa divided by 2.

This merged IFS is now called the intuitionistic fuzzy probabilistic direct impact between two components. It implements the idea that although coupling effects and component resilience are independent, only the simultaneous consideration of both strengths defines the impact.

In order to satisfy aspects of the distributed nature of SLAs in a multi-tier environment, after assessing the direct couplings the indirect impacts can automatically be calculated. The possibility of both, a classical, probabilistic interpretation of the logical operations conjunction ($\wedge$) and disjunction ($\vee$) is a key concept in the proposed indirect impact calculations. The partial impact between the component PI and business KPI is now expressed by means of intuitionistic fuzzy values carrying probabilistic information. The combination of classical and probabilistic applications of the logical operations can as result be interpreted either as a probabilistic indirect dependency between component PI and the business KQI (means the probability that a KQI breaches the SLA in case the component PI fails) or an ordinary indirect fuzzy dependency (means that the KQI is partially out of specification or degraded in case the component PI fails). The following IFS operations are proposed based on the Fault Tree Analysis concept of Kolev/Ivanov in 2009 [16]: classical, moderate, worst and best case impact analyses.

Worst case impact analysis $V(p \wedge q) = \langle \min(\mu(p), \mu(q)), \max(\nu(p), \nu(q)) \rangle$ $V(a \vee b) = \langle \mu(a) + \mu(b) - \mu(a).\mu(b), \nu(a).\nu(b) \rangle$	**Best case impact analysis** $V(p \wedge q) = \langle \mu(p).\mu(q), \nu(p) + \nu(q) - \nu(p).\nu(q) \rangle$ $V(a \vee b) = \langle \max(\mu(a), \mu(b)), \min(\nu(a), \nu(b)) \rangle$
Moderate impact analysis $V(p \wedge q) = \langle \mu(p).\mu(q), \nu(p) + \nu(q) - \nu(p).\nu(q) \rangle$ $V(a \vee b) = \langle \mu(a) + \mu(b) - \mu(a).\mu(b), \nu(a).\nu(b) \rangle$	**Classical fuzzy impact analysis** $V(p \wedge q) = \langle \min(\mu(p), \mu(q)), \max(\nu(p), \nu(q)) \rangle$ $V(a \vee b) = \langle \max(\mu(a), \mu(b)), \min(\nu(a), \nu(b)) \rangle$

Figure 3. Combined classical and probabilistic IFS operations [16]

Depending on which operations are applied, classical or probabilistic, the results will be greater or smaller. The indirect intuitionistic fuzzy dependencies between components may have different kinds of semantics (functional and probabilistic) depending on the type of information they represent.

4 Intuitionistic Fuzzy Component Failure Impact Analysis (IFCFIA)

A complete methodical assessment approach, which is practically usable in data centre environments, includes several sequential steps to be processed. It starts from automated exploring the details of the managed resources and backend components, the grouping to impacted front-end services and the enrichment in several tasks and calculation steps up to the gradual business impact assessments, including monetary cost-of-failure information and business objectives. The overall frame for incorporating all data is the CFIA grid (described in step 3). This matrix can be freely extended with different kind of variables showing failure modes, reliability parameters, financial data, operational capabilities and techniques and extends the pure system view to include also the processes, tools and people (e.g. helpdesk) that are necessary for functioning of a distributed information system.

Step 1: Auto-discovery by ADDM tools

All infrastructure component items and technical dependencies of a defined scope will be auto-discovered using ADDM (Application Dependency Discovery Management) tools. This provides trust that the discovered information is real by automatically discovering inter-dependencies among applications and underlying systems and minimize IT organizations expend on the complex information assimilation. The discovered components with corresponding relations can be extracted by commercial ADDM tools in a structured data format e.g. XML for further automated processing.

Step 2: Defining the Business Service

The in-scope discovered component items are grouped to form the business applications, as the top level in the component hierarchy is the business service. A business service is the way to group the different kinds of IT resources into a logical group which acts together as one unit to provide the service. Business services can contain any number of the lower-level resources. This grouping step creates implicitly the fault tree to the business service by chaining all directly and indirectly linked components. In case an incident occurs, a list of possible components which may be the root cause of the incident can now be identified.

Step 3: Creating the CFIA Grid

After auto-discovering of the in-scope infrastructure components, there relationships and the configurations, the next step is to create a grid with components on one axis and the IT services which have a dependency on the component. This matrix is called CFIA (Component Failure Impact Analysis) grid. In the matrix all data is shown which is relevant for the loosely coupling assessment including the business repair/recovery time objectives. The grid is complemented with the coupling degrees (calculated or by experts assessed) for loosely and tightly coupling. The tightly coupling index is defined as inter-modular coupling metric, which calculate the coupling between each pair of directly related components. For loosely coupling an intrinsic coupling metric is chosen as this refers to the individual components' resilience capabilities. The CFIA will also verbally indicate the assessed level of certainty.

Step 4: Defining the direct impact as Intuitionistic Fuzzy Set

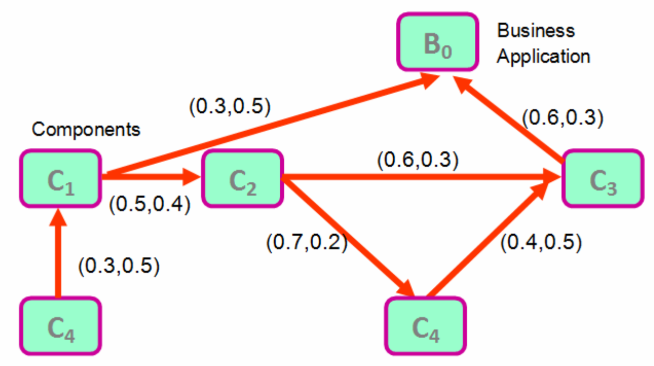

Figure 4. Direct coupling IFS

As next step for the two independent loosely and tightly coupling indexes a combined representation into an integrated Intuitionistic Fuzzy Set (IFS) is created. This requires the two coupling indexes A and B to be normalized and combined by IFS operations (we may choose the fuzzy operation A@¬B). The result of step 4 is the fuzzy intuitionistic direct coupling impact between two components. The direct coupling IFS can be now added to the CFIA grid.

Step 5: Calculating the indirect couplings as IFS

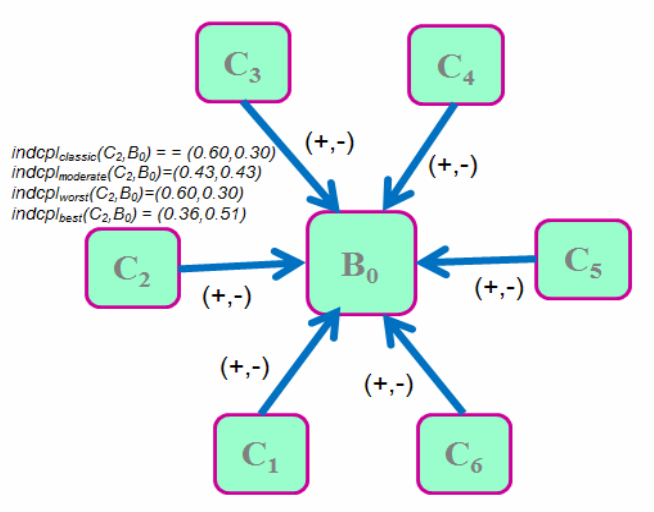

Figure 5. One-Level Dependency Map

Based on the direct couplings, described as inter-modular IFS, the indirect impacts can be calculated. By involving different probabilistic variants of the logical operations when calculating the indirect impacts, the strength of the impact transferred throughout the distributed and multi-tiered system can be modelled. For impact analysis the Forward Coupling Calculation (FCC) is applied which follows the forward dependency direction from the component where the incident occurs and traversing through its direct or indirect dependants. In the KQI/KPI Hierarchy a forward looking coupling calculation means a bottom-up direction. Vice versa, a root cause analysis is a top down approach and requires the reverse task to be solved, i.e. “to which components is the business application coupled to (depends on)” The second method implies the methodology for calculating indirect impacts starting from the dependant and traversing through its impact arcs in the reverse direction. This method is referred as Reverse Coupling Calculation (RCC). For instance following the FCC approach the indirect coupling between component C_2 and business service B_0 can be calculated:

$$indcpl(C_2, B_0) = (dircpl(C_2, C_3) \vee (dircpl(C_2, C_4) \wedge dircpl(C_4, C_3))) \wedge dircpl(C_3, B_0).$$

Applying classical operations $indcpl_{classic}(C_2, B_0)$ = (0.60, 0.30), moderate impact $indcpl_{moderate}(C_2, B_0)$ = (0.43, 0.43), worst case impact assessment $indcpl_{worst}(C_2, B_0)$ = (0.60, 0.30) and best case impact $indcpl_{best}(C_2, B_0)$ = (0.36, 0.51). The result of Step 5 is the coupling index of each component to the business service represented as indirect coupling IFS.

Step 6 (optional): Extending the Business View

The IFCFIA may be optional extended with additional logical dependencies and business impact information. For operation of IT systems we need to know also about dependencies to e.g. IT users and roles, supporting processes or maintenance services. This can be expressed with a coupling relationship like – is coupled to: a procedure, a Service Level Agreement (SLA) or even technical- or user documentation. Also business and monetary information can be added to the service like hourly cost of failure or impacted users. Thus when a component is unavailable, the number of users impacted is understood and an impact calculation based on the assessed cost of unavailability can be performed.

Extended IFCFIA Grid with indirect couplings and cost of failure					cost of failure 10.000 per hour RTO 2 hours RPO 4 hours		cost of failure 3.000 per hour RTO 12 hours RPO 12 hours			
					Bus. Service 1 # Users 700		Bus. Service 2 # Users 300			
Component	Discovered Node Id	Parent Node Id's	Failure Mode and Effect	Direct Impact (IFS) on parent	FCC coupling to Business Service 1	RCC coupling from Business Service 1	FCC coupling to Business Service 2	RCC coupling from Business Service 2	total End Users impacted	total cost of failure per hour
Switch A	S-01	BusServ	Outage	(0.4,0.4)	(0.4,0.4)	(0.4,0.4)			700	4000
Firewall A	F-01	S-01	Outage	(0.6,0.3)	(0.3,0.5)	(0.4,0.5)			700	3000
Load-Balancer A1	L-01	F-01,F02	Outage	(0.5,0.5)	(0.4,0.4)	(0.6,0.3)			700	4000
Load-Balancer A1	L-01	F-01,F02	Limited Function	(0.4,0.5)	(0.3,0.5)	(0.2,0.6)			700	4000
HTTP Server A	HS-01	L-01,L02	Slow Response	(0.8,0.1)	(0.6,0.3)	(0.7,0.3)	(0.5,0.3)	(0.7,0.3)	1000	7500
How-to Manual	Sup1	HS-01	Quality Issue	(0.2,0.6)	(0.3,0.5)	(0.2,0.6)			700	3000
Technical Support	Org2	HS-01	Slow Repair	(0.8,0.1)	(0.8,0.1)	(0.8,0.1)			700	8000
Helpdesk	Org1	BusServ	Quality Issue	(0.6,0.3)	(0.6,0.3)	(0.7,0.3)	(0.6,0.3)	(0.7,0.3)	1000	7800

Figure 6. Extended CFIA with Cost of Failure

Step 7 (optional): Applying Intuitionistic Fuzzy Reasoning

As last step the IFCFIA allows the application for two-sided (intuitionistic) fuzzy reasoning by combining both aspects including the vagueness of the fact into inference rules and logics. Using two-sided fuzzy logic, the complex system behaviour can be closely analysed by considering both contrary coupling aspects simultaneously. Two-sided fuzzy *if-then* rules can consider different interpretations of fuzzy implications, by applying bi-polar operations and interpretations. Once we have determined the fuzzy rules to define the performance measures, we can create linguistic rules for the service that will help to predict the impact to the front-stage service quality (QoS).

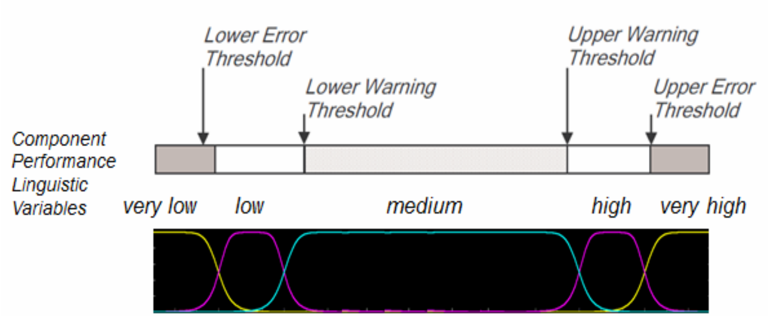

Figure 7. Map thresholds into linguistic variables

For instance: *If {"Component Service" is (tightly coupled > 0.5) and (loosely coupled < 0.4) to "Business Service" and ("Component Service Performance" is LOW or "Component Service Reliability" is LOW)} then "Business Service" performance is LOW*

5 Data center use cases for the logistics management application

Several real world datacenter use cases have been developed for the IFCFIA framework [17]. These comprise Business Impact Analysis, Root Cause Analysis, Advanced Service Level Monitoring and Capacity Optimization in Consumption Based Usage and Charging Models.

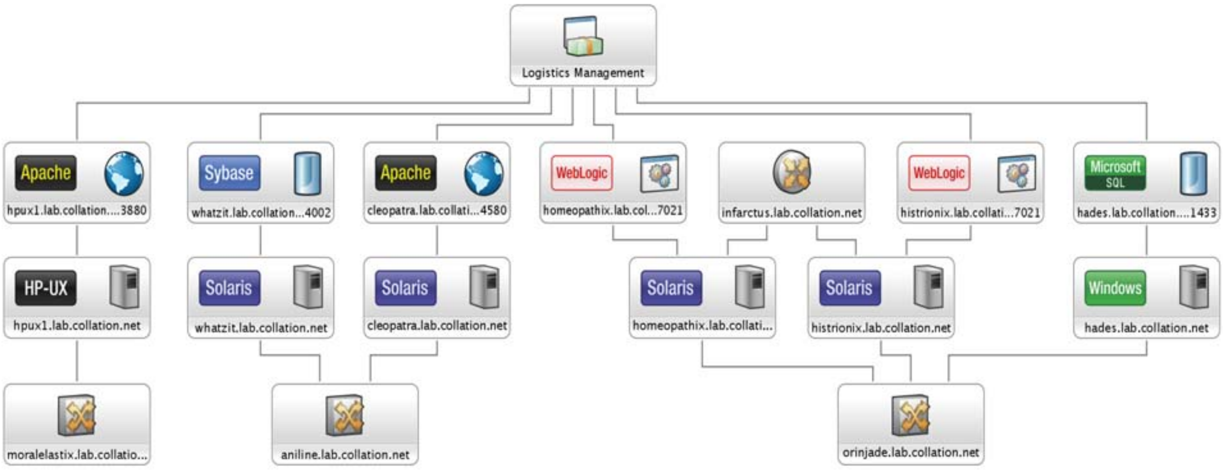

Figure 8. Logistics management application physical topology

Using IFCFIA the back-stage metrics can be related to the front services experienced by the business. The IFCFIA model about a set of fuzzy-coupled com-ponents to a business service with corresponding performance parameters can be utilized to support Service Management to predict on impacts of monitored back-end component failures and incidents.

The result of the IFCFIA analysis is a sorted intuitionistic fuzzy distribution of components providing an ordered set by the probability of incident root causes. It can be a guide in the process of discovering root causes of SLA violations and helps to provide more accurate analyses that are needed for appropriate adjustment decisions at runtime. To justify IT investments IFCFIA can demonstrate how the proposed infrastructure improvements will deliver tangible business benefits by quantifying the impact to the total cost of failures of dependent frontend services. Within ITIL v3 best practices IFCFIA can extend Availability-, Capacity-, Configuration- and Change Management processes by providing the gradual interdependency relationships. Within service operations the ITIL Incident and Problem Management processes can benefit from advanced root cause determination and impact assessments by connecting IT operations to business services, means transformation of availability and performance data into knowledge about the real-time status of business services that allows understanding and communicating the true impact of incidents (such as IT component failures) on the business and vice versa.

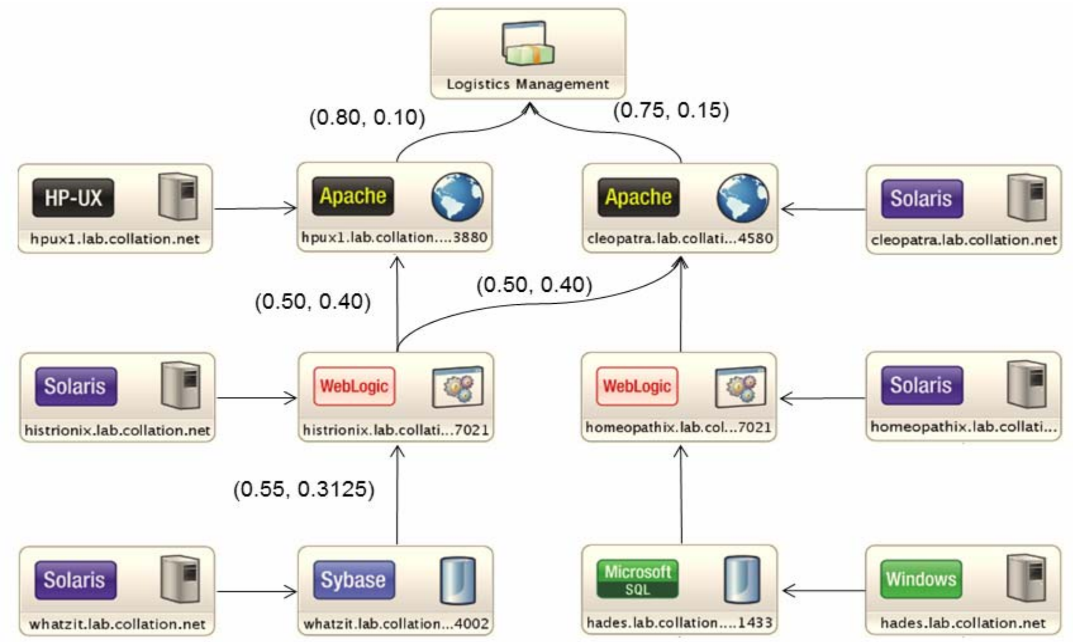

Figure 9. Logistics management application physical topology

References

[1] Bianco, P., G. Lewis, P. Merson. *Service Level Agreements in Service-Oriented Architecture Environments Software Architecture Technology Initiative* CMU/SEI-2008-TN-021 Sept. 2008.

[2] Service Level Agreement Zone, 2007. `http://www.sla-zone.co.uk`

[3] Hui Li, Challenges in SLA Translation – SLA@SOI European Commission Seventh Framework Programme (2007–2013) *SAP Research*, Dec. 2009. `http://sla-at-soi.eu/2009/12/challenges-in-sla-translation/`

[4] Joshi, K. P., A. Joshi, Y. Yesha, Managing the Quality of Virtualized Services. *Proc. of the SRII Service Research Global Conference*, 2011. `http://ebiquity.umbc.edu/get/a/publication/541.pdf`

[5] TeleManagement Forum: SLA Management Handbook – Volume 4: Enterprise Perspective, ISBN: 1-931624-51-8. `http://www.afutt.org/Qostic/qostic1/SLA-DI-USG-TMF-060091-SLA_TMForum.pdf`

[6] Rud, D., A. Schmietendorf Reine. Resource Metrics for Service-Oriented Infrastructures 2007. http://www.cs.uni-magdeburg.de/~rud/papers/Rud-13.pdf

[7] Dhama, H. Quantitative models of cohesion and coupling in software, *Journal of Systems and Software*, Volume 29, April 1995, Issue 1, 65–74.

[8] Fenton, N., A. Melton. Deriving structurally based software measures. *Journal of Systems and Software*, Vol. 12, 1990, Issue 3, 177–187.

[9] Alghamdi, J. S. Measuring software coupling. *Proceedings of the 6th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems, SEPADS'07*, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS), 2007, 6–12.

[10] Pham Thi Quynh, Huynh Quyet Thang. Dynamic Coupling Metrics for Service–Oriented Software. http://www.waset.org/journals/ijeee/v3/v3-5-46.pdf

[11] Van Haren 08: Continual Service Improvement Based On ITIL V3: A Management Guide (Best Practice). http://www.openisbn.com/isbn/9087531281/

[12] Bailey, D., E. Frank-Schultz, P. Lindeque, J. L. Temple, III. Three reliability engineering techniques and their application to evaluating the availability of IT systems, *IBM Systems Journal*, Vol. 47, 2008, No. 4, 577–589.

[13] Hong Yul Yang. Measuring Indirect Coupling. Doctor Thesis at Department of Computer Science, University of Auckland, New Zealand, 2010.

[14] Atanassov, K. On Intuitionistic Fuzzy Sets Theory (Studies in Fuzziness and Soft Computing) Springer Berlin Heidelberg, 2011.

[15] Zadeh, L. A. *Soft Computing and Fuzzy Logic*, University of Berkeley, California USA

[16] Kolev, B., I. Ivanov. Fault Tree Analysis in an Intuitionistic Fuzzy Configuration Management Database, *Notes on Intuitionistic Fizzy Sets*, Vol. 15, 2009, No. 2, 10–17.

[17] Schütze, R. IFCFIA, a gradual method for SLA dependency mapping and bi-polar impact assessment. *Technical Report 07/2013*, University of Fribourg, Switzerland, Department of Informatics.