IT Business Service-Level-Management—An Intuitionistic Fuzzy Approach

Roland Schuetze

Abstract Bridging from IT-centric service levels, written in IT technical terms, to business-oriented service achievement is a hot topic in today's service research. The proposed 'IFSFIA' 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. The concept starts from the idea of naturally approaching impact relationships by separately envisaging positive and negative aspects with the notion of bipolarity. Performing a multi-level impact 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 a business service.

Keywords Service level management • SLA • Business impact • Intuitionistic fuzzy sets

This concept was first published at ICIFS conference, Nov. 2013, Sofia Bulgaria [1], the following version includes several conceptual extensions.

R. Schuetze (✉)
Department of Informatics, University of Fribourg, Fribourg, Switzerland
e-mail: roland.schuetze@unifr.ch

P. Angelov and S. Sotirov (eds.), *Imprecision and Uncertainty in Information Representation and Processing*, Studies in Fuzziness and Soft Computing 332,
DOI 10.1007/978-3-319-26302-1_16

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. 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 [2]. 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 [3]. 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 [4] defined a 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, whereas the monitoring instrumentation is established for the lower level components to ensure the fulfillment of the service quality objectives (Fig. 1).

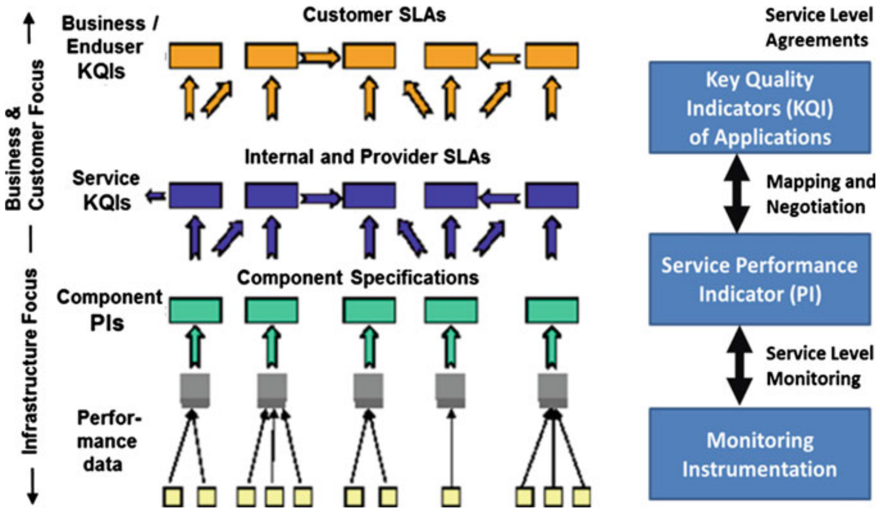


Fig. 1 KQI, PI & SLA relationship [5]

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. 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 for 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 component 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. The goal is to build components that do not have tight dependencies on each other, so that if one service 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 to still continue to work. Loose coupling describes an approach where integration interfaces are developed with minimum assumptions between the sending/receiving parties, thus reducing the risk that failure in one module will affect others. Loose coupling isolates the components of an application so that each component interacts asynchronously and treats others as a “black box”. E.g. 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. 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 e.g. proposed by Dhama [5] or Fenton and Melton [6]. More advanced are object-oriented coupling measures [7] and further several metrics are proposed to evaluate the coupling level real-time by runtime monitoring, introduced as dynamic coupling metrics [8].

2.3 *Application Dependency Discovery Management (ADDM)*

Application discovery is the process of automatically analyzing artefacts of a software application and physical elements that constitute a network (e.g., servers, firewalls, etc.). ADDM products [9] deliver a powerful enabler that minimize IT organizations expend on the information assimilation function and can also provide a basis for further higher level, logical dependency assessments. According to [10] these tool assert networks mainly based on three different approaches: middleware or instrumenting applications; analyzing program configuration files or analyzing application traffic. ADDM products deliver a point-in-time view of the “truth” and unveil dependencies, but do not measure a granular truth value of an impact two service components may have on each other. Dependency graphs created by an automated discovery tool can be leveraged as a great starting point for advanced methods to calculate granular degrees of dependence.

An inductive approach can also be chosen by calculating couplings between servers or services based on historical data collected from the actual server network. As opposite a deductive method would be applicable, where dependencies are not calculated based on data the system produces, but rather the system itself, for

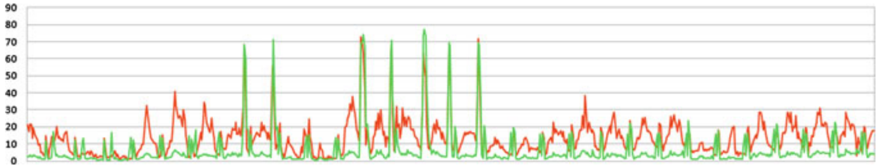


Fig. 2 Inductive coupling assessment between database and application performance

example plans system architects make or comparisons to other systems, which have a similar layout (Fig. 2).

For inductive coupling measurements statistical methods can be applied or an expert can determine coupling effects based on the given data-series and his experience.

2.4 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 behavior 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.

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.

3 Applying the Model of Intuitionistic Fuzzy Sets

3.1 Coupling Statements as Intuitionistic Fuzzy Sets

Let E be a fixed universe and A is a subset of E. The set $A^* = \{(x, \mu A(x), \nu A(x))| x \in E\}$ 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 (Zadeh) [16] 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 $< \mu a, \nu a >$ and $< \mu b, \nu b >$. The tightly coupling a degree of truth is $< \mu a >$ and the degree of falsity $< \nu a >$. The same assessment is done for loosely coupling b where $< \mu b, \nu b >$ represent the degrees of truth and falsity. This maps service quality impacts to the idea behind intuitionistic fuzzy service dependencies, where the level of tightly coupling between service components corresponds to the intuitionistic fuzzy degrees of truth and falsity of the dependency impact and the loosely coupling index assesses the resilience capabilities of a service.

3.2 Defining the Fuzzy Intuitionistic Direct Coupling Between Components

The validities (membership degrees) for tightly and loosely couplings are independently estimated by separate approaches, for ‘tightly’ using the described inter-modular coupling metrics and for ‘loosely’ applying assessed intrinsic component resilience capabilities. In praxis dependencies are naturally expressed by positive forms (membership) only, which is also the way human assessments work. Thus, the proposed method does only require the experts to judge on the validity of the tightly and loosely coupling and to specify a level of certainty of these statements (Fig. 3).

The vagueness is expressed in linguistic terms and mapped into a crisp number with regard to the applied complement function, omitting that $\lambda >= 0$ (Sugeno) or $w <= 1$ (Yager). The non-validity is then automatically set by the fuzzy complement function (Fig. 4).

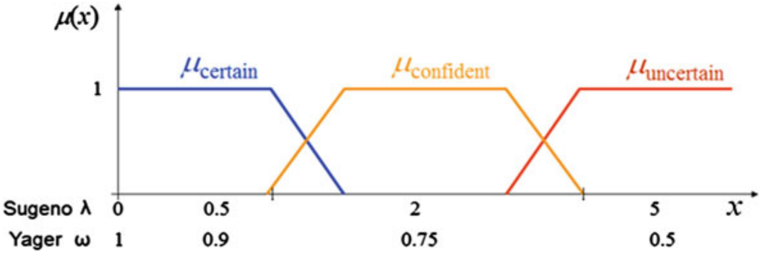


Fig. 3 Certainty mappings to define Sugeno and Yager complements

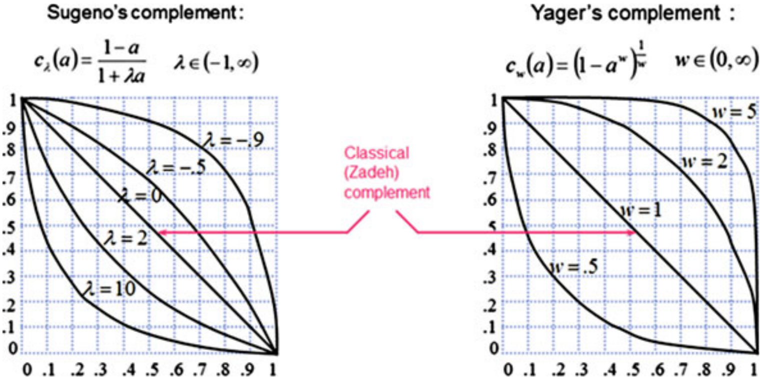


Fig. 4 Sugeno and Yager fuzzy complements

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 in a single IFS. 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 $A@\neg B$ by adding membership 'tightly' with non-membership 'loosely' and vice versa divided by 2. The combined degrees are further referred as μ_D and ν_D for direct coupling index and are called the intuitionistic fuzzy probabilistic direct impact between two related components.

$$\mu_{combined}(x) = \frac{\mu_A(x) + \nu_B(x)}{2} \text{ and } \nu_{combined}(x) = \frac{\nu_A(x) + \mu_B(x)}{2} \tag{1}$$

It implements the idea that although the coupling effects and component resilience are independent, only the simultaneous consideration of both strengths together defines the impact. This implies a beforehand a normalization of the positive and negative effects (even there are independent measurements used) for getting comparable weights, which is a key challenge to get accurate results applying the proposed method.

The direct coupling from component x to component y can now be defined where V is the described evaluating function of the intuitionistic fuzzy coupling statement.

$$V(dirdcpl(x,y)) = \begin{cases} <\mu_D(x,y), \nu_D(x,y)>, & \text{if } <x,y> \in D \\ <1,0>, & \text{if } <x,y> \notin D \end{cases} \tag{2}$$

The defined IFS is further called the fuzzy intuitionistic direct coupling index between the two components x and y.

3.3 Calculation of Indirect Coupling Impacts

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. This concept was developed within the Fault Tree Analysis by Kolev/Ivanov in 2009 [16]. The indirect coupling from component *x* to service *y* can be defined as follows where *i* is the component directly coupled to *y* on the path from *x* to *y*.

$$V(indcpl(x,y)) = \begin{cases} \bigvee_{i,y \in D} indcpl(x,i) \wedge dircpl(i,y), & \text{if } x \neq y \\ <1,0>, & \text{if } x = y \end{cases} \tag{3}$$

Within the KQI/KPI hierarchy model the methodology for calculating the indirect coupling follows the forward dependency direction (Forward Coupling Calculation FCC). In case of an incident this means starting from the failed node in the hierarchy and traversing through its direct or indirect dependants to the business service. 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 B coupled to (depends on)” The second method implies the definition of methodology for calculating indirect impacts starting from the dependant and traversing through its impact arcs in the reverse direction. We refer to this method as Reverse Coupling Calculation (RCC).

$$V(indcpl(x,y)) = \begin{cases} \bigvee_{x,i \in D} dircpl(x,i) \wedge indcpl(i,y), & \text{if } x \neq y \\ <1,0>, & \text{if } x = y \end{cases} \tag{4}$$

The possibility of both, a classical, probabilistic interpretation of the logical operations conjunction (∧) and disjunction (∨) is a key concept in the 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. These IFS operations are proposed for classical, moderate, worst and best case impact analysis [16]:

$$\textbf{Worst Case}\quad \begin{array}{l} V(p \wedge q) = \langle \min(\mu(p), \mu(q)), \max(v(p), v(q)) \rangle \\ V(a \vee b) = \langle \mu(a), \mu(b) - \mu(a).\mu(b), v(a).v(b) \rangle \end{array} \tag{5}$$

$$\textbf{Moderate Case}\quad \begin{array}{l} V(p \wedge q) = \langle \mu(p).\mu(q), \nu(p) + \nu(q) - \nu(p).\nu(q) \rangle \\ V(a \vee b) = \langle \mu(a) + \mu(b) - \mu(a).\mu(b), \nu(a).\nu(b) \rangle \end{array} \tag{6}$$

$$\textbf{Best Case}\quad \begin{array}{l} V(p \wedge q) = \langle \mu(p).\mu(q), \nu(p) + \nu(q) - \nu(p).\nu(q) \rangle \\ V(a \vee b) = \langle \max(\mu(a), \mu(b)), \min(\nu(a), \nu(b)) \rangle \end{array} \tag{7}$$

$$\textbf{Fuzzy Classical}\quad \begin{array}{l} V(p \wedge q) = \langle \min(\mu(p), \mu(q)), \max(v(p), \nu(q)) \rangle \\ V(a \vee b) = \langle \max(\mu(a), \mu(b)), \min(\nu(a), \nu(b)) \rangle \end{array} \tag{8}$$

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. Combinations 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 functioning in case the component PI fails).

4 Intuitionistic Fuzzy Service Failure Impact Analysis (IFSFIA)

A complete methodical assessment approach, which is practically usable in datacentre environments, includes several sequential steps to be processed. It starts from automated exploring the details of the managed resources and backend components, the grouping of components to impacted frontend 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.

4.1 IFSFIA Structured Step-by-Step Approach

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 interdependencies 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. For the later use cases IBM's Tivoli Application Dependency Discovery Manager (TADDM) is chosen as auto-discovery solution that provides in depth automated application dependency mapping and configuration auditing [21].

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). This enables the identification of critical components (that could cause the failure of multiple IT services) and fragile IT services (that have multiple single points of failure). A basic CFIA will target a specific section of the infrastructure; just looking at simple binary choices (e.g. if we lose component x, will a service stop working? More advanced CFIAs can be expanded to include a number of variables, such as likelihood of failure, repair and recovery time, recovery procedures, organizational assignments and integration into wider service management processes and also can also consider and evaluate for different component failure modes. So within the IFSFIA method in the matrix all data is added which is relevant for the loosely coupling assessment including the business recovery time objectives. The grid is complemented with the evaluated degrees 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: Define the Fuzzy Intuitionistic Direct Impact

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 basic IFS 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 (Fig. 5).

Step 5: Calculating the Fuzzy Intuitionistic Indirect Couplings

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 (depend on)" as Reverse Coupling Calculation (RCC).

In the following example using the forward (FCC) approach for impact assessments in case a component C_2 fails to the business service B_0:

$indcpl(C_2,B_0) = (dircpl(C_2,C_3) \vee (dircpl(C_2,C_4) \wedge dircpl(C_4,C_3))) \wedge dircpl(C_3, B_0)$. *Using* classical operations the $indcpl_{classic}(C_2,B_0) = (0.60,0.30)$, moderate impact $indcpl_{moderate}(C_2,B_0) = (0.43,0.43)$, worst case impact $indcpl_{worst}(C_2, B_0) = (0.60,0.30)$ and best case impact assessment $indcpl_{best}(C_2,B_0) = (0.36,0.51)$.

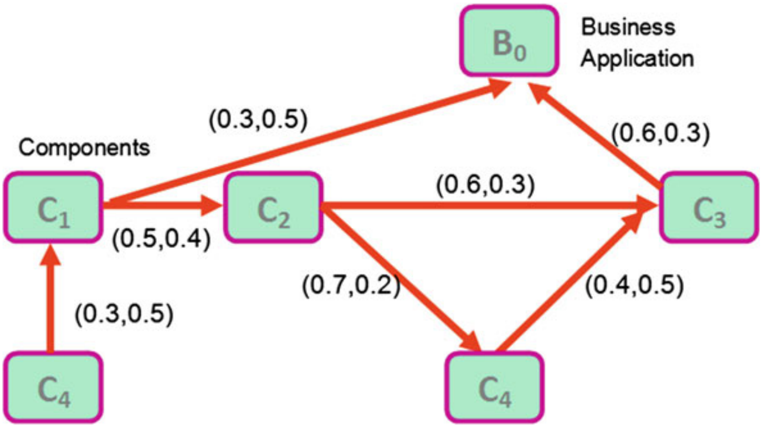


Fig. 5 Directed graph with direct couplings as IFS

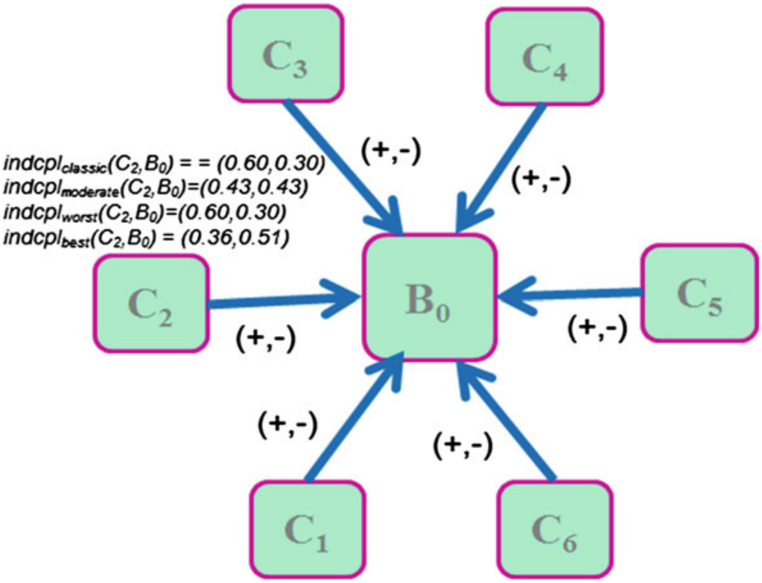

C_3
C_4
(+,-)
$indcpl_{classic}(C_2, B_0) = = (0.60, 0.30)$
$indcpl_{moderate}(C_2, B_0) = (0.43, 0.43)$
$indcpl_{worst}(C_2, B_0) = (0.60, 0.30)$
$indcpl_{best}(C_2, B_0) = (0.36, 0.51)$
(+,-)
C_2
B_0
C_5
(+,-)
(+,-)
(+,-)
(+,-)
C_1
C_6

Fig. 6 One-Level dependency map as star schema

The result of step 5 is the fuzzy intuitionistic coupling index of each component to the business service represented as indirect coupling IFS (Fig. 6).

Step 6 (Optional): Extending the Business View

The IFSFIA 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).

Also business and monetary information can be added to the service like hourly cost of failure or impacted users [18]. This can enable cost calculations based on the number of users concerned and/or amount of lost user processing time or even total cost of unavailability. However, the number of user workstations does not necessarily equate to the number of users at one point in time. So other measurements of costs of failure should complement these numbers, like SLA penalties when service providers fail to deliver the pre-agreed quality, estimation of the financial impact of IT failure against transaction volumes (related to the vital business functions) normally processed during the period of failure. For certain businesses a consequence of IT failure may be even external claims for financial compensation by impacted customers or business partners (Fig. 7).

The created CFIA matrix is expanded to include fields related to the Business Value and the Cost of Failure of a Service. These fields can simply show the hourly failure cost to the business or can map the number of users supported by each business service.

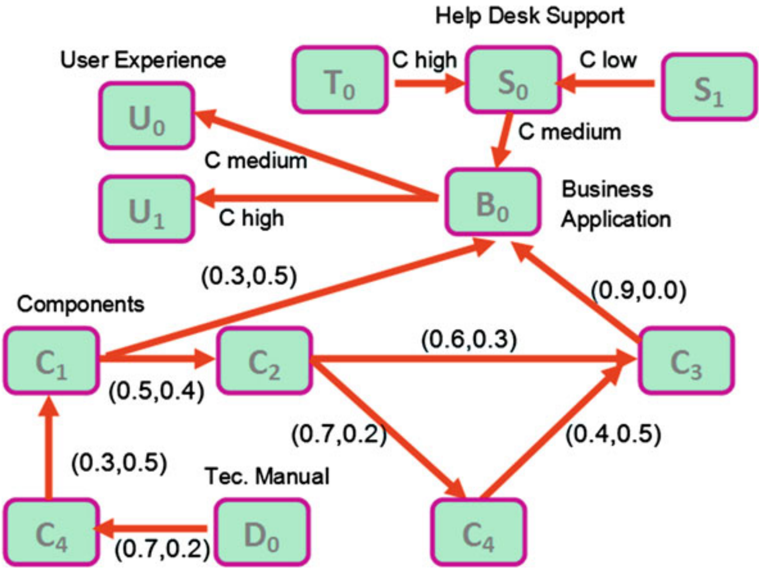


Fig. 7 Extended directed graph with couplings including it enabled services

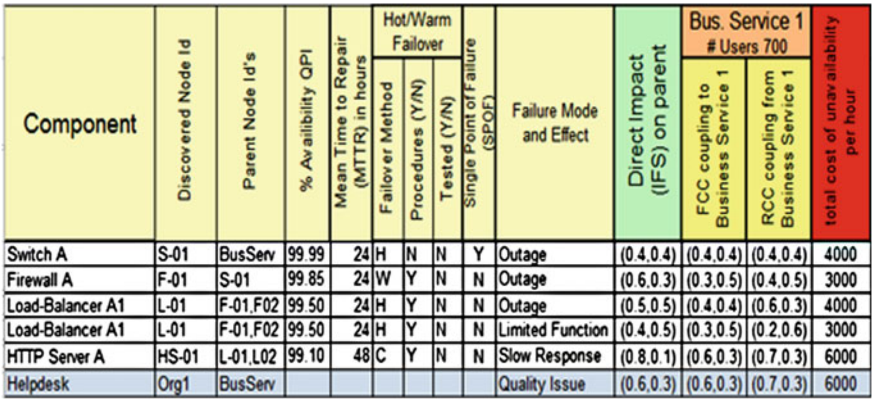
Step 7: Performing Business Impact and Root Cause Analysis

A high tightly coupling index indicates a higher risk to the affected business service, which means this infrastructure component, is vital to business. A high loosely coupling index for a component indicates a strong resilience capability which allows smaller buffer overhead in the individual component's capacity planning and sizing.

The IFSFIA can be used in two principal ways, bottom-up as impact assessment or top-down for root cause analysis.

7(a) Business Impact Analysis (BIA)

Business Impact Analysis identifies vital business functions and their dependencies. These dependencies may include suppliers, business processes, IT Services etc. BIA defines as an output the requirements which include recovery time objectives and minimum Service Level Targets for each IT Service. The impact analysis using the IFSFIA can answer the question "Which are the indirect dependant business services of a particular component x and to which level are they tightly or loosely coupled?" starting from the low-level infrastructure component in the dependency hierarchy and traversing through its direct or indirect dependants to the business application services. The same BIA estimate used during operation to assess the business impact of incidents, can also be used to justify IT Infrastructure improvements by quantifying the total cost to the organization of an IT Service failure(s). These costs can then be used to support a business case for additional IT Infrastructure investment and provide an objective 'cost versus benefit' assessment.



Component	Discovered Node Id	Parent Node Id's	% Availibility QPI	Mean Time to Repair (MTTR) in hours	Hot/Warm Failover			Single Point of Failure (SPOF)	Failure Mode and Effect	Direct Impact (IFS) on parent	Bus. Service 1 # Users 700		total cost of unavailability per hour
					Failover Method	Procedures (Y/N)	Tested (Y/N)				FCC coupling to Business Service 1	RCC coupling from Business Service 1	
Switch A	S-01	BusServ	99.99	24	H	N	N	Y	Outage	(0.4,0.4)	(0.4,0.4)	(0.4,0.4)	4000
Firewall A	F-01	S-01	99.85	24	W	Y	N	N	Outage	(0.6,0.3)	(0.3,0.5)	(0.4,0.5)	3000
Load-Balancer A1	L-01	F-01,F02	99.50	24	H	Y	N	N	Outage	(0.5,0.5)	(0.4,0.4)	(0.6,0.3)	4000
Load-Balancer A1	L-01	F-01,F02	99.50	24	H	Y	N	N	Limited Function	(0.4,0.5)	(0.3,0.5)	(0.2,0.6)	3000
HTTP Server A	HS-01	L-01,L02	99.10	48	C	Y	N	N	Slow Response	(0.8,0.1)	(0.6,0.3)	(0.7,0.3)	6000
Helpdesk	Org1	BusServ							Quality Issue	(0.6,0.3)	(0.6,0.3)	(0.7,0.3)	6000

Fig. 8 Extended IFSFIA matrix with couplings and cost of failure

Since the coupling measurements to the business applications are defined the cost can be computed where *n* is the number of business applications *i, CCI* denotes the hourly cost of a of the component item *Ci, μA(x)i* is the degree of membership of tightly coupling of the component up to the business application *i* and *Ci* denotes the hourly cost of a failure of the business application *i.*

$$C_{CI} = \sum_{i=1..n}^{n} \mu_A(x)_i{}^{*}C_i \tag{9}$$

The calculated total cost of failure per component can then be added as column to the IFCIA grid which allows assessing at one glance the monetary impact (Fig. 8).

In praxis business impact is hard to measure, as it could have several consequences, from financial impact to fuzzy aspects like feeling of dissatisfaction if IT service problems occur. Measurements on business impact of a failure are hard to quantify in monetary value, like “user productivity loss” or “lost business cost” etc.

7(b) Root Cause Analysis (RCA)

A root cause analysis (RCA) is a top down approach and requires the reverse task then the impact analysis to be solved, i.e. “To which components is the business application B coupled to (depends on)”. The IFSFIA analysis procedure takes into account direct and indirect impacts of other components over the failed components. The result of the analysis is an intuitionistic fuzzy distribution of components giving an ordered set of possible root causes. Having the IFSFIA grid created, we simply can sort for the highest level of IFS coupling to get an order for the probability of possible root causes. The infrastructure component with the highest coupling is most likely and should therefore first being considered for causing the impact on a higher business service [17].

RCA implies the calculating of indirect impacts starting from the top and traversing through its impact arcs in the reverse direction. For RCA the Reverse

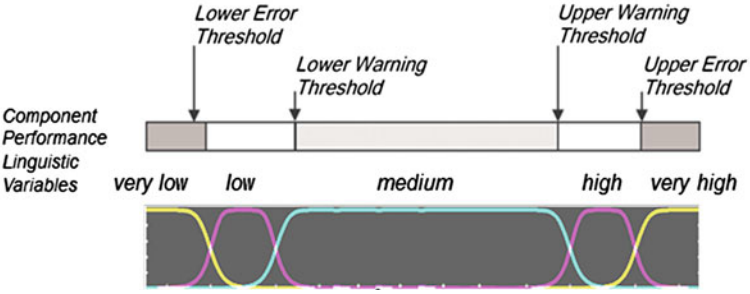


Fig. 9 Mapping of thresholds into linguistic variables

Coupling Calculation (RCC) index in the IFSFIA grid is chosen which may differ from the Forward Coupling Calculation (FCC) index which is applied for bottom-up impact calculations.

Step 8 (Optional): Applying Intuitionistic Fuzzy Reasoning
As final step the IFSFIA allows the application for two-sided (intuitionistic) fuzzy reasoning by combining both aspects including the vagueness of the fact into inference rules and logics. Thresholds can be used as natural limits to assign fuzzy linguistic variables to performance values (Fig. 9).

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).

E.g.: *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.*

4.2 Impact Analysis for Gradual Failure Modes

To reduce the complexity compliance for technical performance parameters will in praxis mostly measured bi-modal (either they operate correctly or they fail). This model can now be extended for granular failure impacts or service degradation effects and the consideration of several parallel incidents which causes the total impact. Thus a forecast can be given on the effect of e.g. 80 % SLA achievement or 60 % compliance to performance specifications. The direct coupling dependencies can be visualized within a directed graph representing the direct fuzzy intuitionistic impacts. The map consists of nodes and arcs between nodes. Each node represents a quality characteristic of the system. In the IT landscape model these characteristics could indicate the level of compliance to the SLA quality targets. Each service level specification parameter described as Key Quality Indicator (KQI) represents a node. Each KQI is characterized by a number *Ai* that represents its value and it results from the transformation of the SLA compliance level for which this node stands, in the interval *[0,1].* The tightly coupling model describes the causal relationships between two nodes. A decrease in the value of a quality parameter (QoS) or SLA compliance level would yield a corresponding decrease at the nodes connected to it via tightly coupling relationships, thus soft effects of partial functioning or degraded SLA compliance between IT components can be directly modeled by the same approach. This concept is briefly derived from the mathematical model of cognitive maps. In 1986 Bart Kosko [19] introduced the notion of fuzziness to cognitive maps and created the theory of Fuzzy Cognitive Maps (FCMs). A Fuzzy cognitive map is a cognitive map within which the relations between the elements (e.g. components, IT resources) can be used to compute the "strength of impact" of these elements. FCMs are used in a much wider range of applications [20] which all have to deal with creating and using models of impacts in complex processes and systems. In the IT landscape scenario FCMs can be used to describe mutual dependencies between infrastructure and higher level IT components. The activation level of a quality parameter indicates in this extended model the level of SLA compliance The model of the classical FCM is now leveraged to compute the value of each quality parameter that influenced by the values of the coupled quality indicator with the appropriate weights and by its previous value.

$$A_i = f\Big(\sum_{\substack{j=1 \\ j \neq i}}^{n} A_j W_{ji} \Big) + A_i^{old} \tag{10}$$

So the value *Ai* for each quality indicator *KQIi* can be calculated by the rule where Ai is the activation level of quality parameter *KQIi* at time *t + 1, Aj* is the activation level of quality parameter *KQIj* at time *t, Ai* old is the activation level of quality parameter *KQIi* at time *t*, and *Wji* is the weight of the dependence coupling between *KQIj* and *KQIi*, and *f* is a threshold function. The weights of the dependencies between the *KQIi* and *KQIj* could be positive *(Wji > 0)* which means that an increase in the value of *KQIi* leads to the increase of the value of *KQIj*, and a decrease in the value of KQIi leads to the decrease of the value of *KQIj*. In case of a negative causality *(Wji < 0)* which means that an increase in the value of *KQIi* leads to the decrease of the value of *KQIj* and vice versa (Fig. 10).

By adding also the activation levels of the KQIs, each KQI is characterized by a number *Ai* that represents its value and it results from the transformation of the SLA compliance level for which this KPI stands, in the interval *[0,1].*

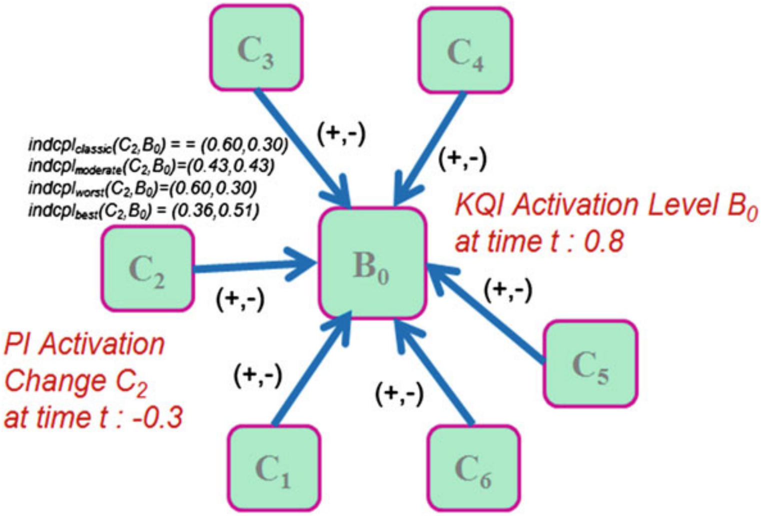

C3
C4
(+,-)
(+,-)
indcplclassic(C2,B0) = = (0.60,0.30)
indcplmoderate(C2,B0)=(0.43,0.43)
indcplworst(C2,B0)=(0.60,0.30)
indcplbest(C2,B0) = (0.36,0.51)
KQI Activation Level B0
at time t : 0.8
C2
B0
(+,-)
(+,-)
PI Activation
Change C2
at time t : -0.3
(+,-)
(+,-)
C5
C1
C6

Fig. 10 Couplings related to KQI activation levels

As example: Using the Forward Coupling Calculation (FCC) method (applicable for Impact Analysis) of *indcpl(C_2,B_0)* depicted in the example graph shows the indirect coupling dependency of the Business Application B_0 on the Component C_2.

- $indcpl_{classic}(C_2,B_0) = (0.60,0.30)$
- $indcpl_{moderate}(C_2,B_0) = (0.43,0.43)$
- $indcpl_{worst}(C_2,B_0) = (0.60,0.30)$
- $indcpl_{best}(C_2,B_0) = (0.36,0.51)$

Now the calculation of the KQI Activation Level for B_0 at time $t + 1$ can be done as follows using an activation level of $KQI_T\ B_0 = 0.8$ at point in time t.

- $KQI_{T+1}\ B_{0\ classic} = (0.8 - 0.3 * 0.6) = 0.62$
- $KQI_{T+1}\ B_{0\ moderate} = (0.8 - 0.3 * 0.43) = 0.671$
- $KQI_{T+1}\ B_{0\ worst} = (0.8 - 0.3 * 0.6) = 0.62$
- $KQI_{T+1}\ B_{0\ best} = (0.8 - 0.3 * 0.36) = 0.692$

In case the performance indicator C_2 decreases of 0.3, an impact between a decrease 0.108 and 0.18 to the quality indicator KQI B_0 is estimated. This simple approach can be helpful where it is required to consider how several smaller improvements at different infrastructure components (e.g. improvements in performance or throughput) in total will impact a business service performance parameter KQI. All impacts will be pulled together so all single impacts are aggregated to the total effect on the business.

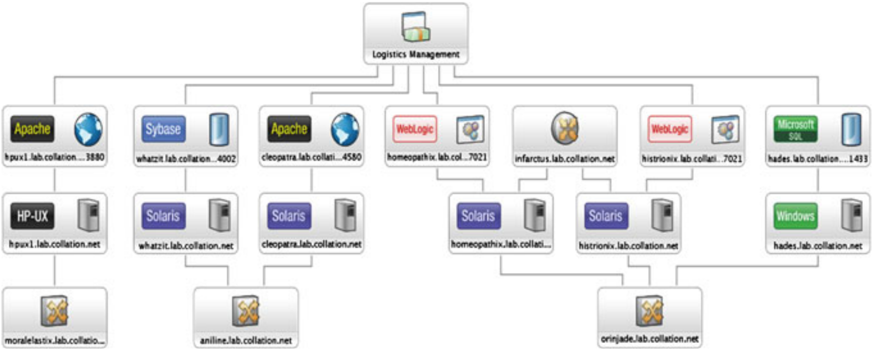

Logistics Management
Apache
hpux1.lab.collation....3880
Sybase
whatzit.lab.collation...4002
Apache
cleopatra.lab.collati...4580
WebLogic
homeopathix.lab.col...7021
infarctus.lab.collation.net
WebLogic
histrionix.lab.collati...7021
Microsoft SQL
hades.lab.collation....1433
HP-UX
hpux1.lab.collation.net
Solaris
whatzit.lab.collation.net
Solaris
cleopatra.lab.collation.net
Solaris
homeopathix.lab.collati...
Solaris
histrionix.lab.collation.net
Windows
hades.lab.collation.net
moralelastix.lab.collatio...
aniline.lab.collation.net
orinjade.lab.collation.net

Fig. 11 Logistics management application physical topology

5 Data Center Use Cases

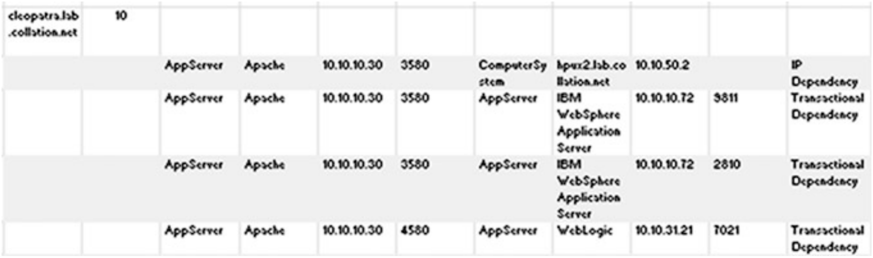
Several real world datacenter use cases have been developed for the IFSFIA framework [21]. These comprise Business Impact Analysis, Root Cause Analysis, Advanced Service Level Monitoring and Capacity Optimization which have been developed as use cases for the business application “Logistics Management” (Fig. 11).

5.1 Performing the IFSFIA Analysis

In the use case the IBM Tivoli Application Dependency Discovery Manager (TADDM) component affinity report extracts all related components which have a dependency (IP dependency, transactional dependency or configuration dependency) on those components which are directly related to the in-scope business services. It creates a table of all servers within the specified scope that are sources of relationships, and the connections from those servers to other server and middleware applications [17] (Fig. 12).

The Intuitionistic Coupling Index is now determined with regard to an appropriate formula (e.g. Dhama’s metric) or alternatively assessed by the experts via inductive monitoring of relevant performance indicators. 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. Both index are normalized and pulled together into a single IFS, the fuzzy intuitionistic direct impact (Fig. 13).

The methodology for calculating the indirect coupling follows the forward dependency direction. Following it the indirect dependants of the failed component x are determined, starting from the node x in the dependency graph and traversing



cleopatra.lab.collation.net	10									
		AppServer	Apache	10.10.10.30	3580	ComputerSystem	hpux2.lab.collation.net	10.10.50.2		IP Dependency
		AppServer	Apache	10.10.10.30	3580	AppServer	IBM WebSphere Application Server	10.10.10.72	9811	Transactional Dependency
		AppServer	Apache	10.10.10.30	3580	AppServer	IBM WebSphere Application Server	10.10.10.72	2810	Transactional Dependency
		AppServer	Apache	10.10.10.30	4580	AppServer	WebLogic	10.10.31.21	7021	Transactional Dependency

Fig. 12 TADDM Server affinity report

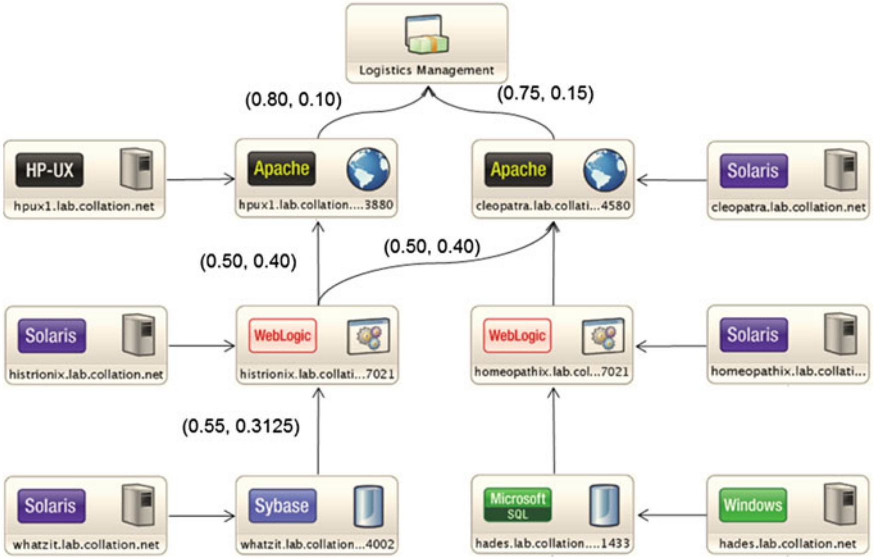
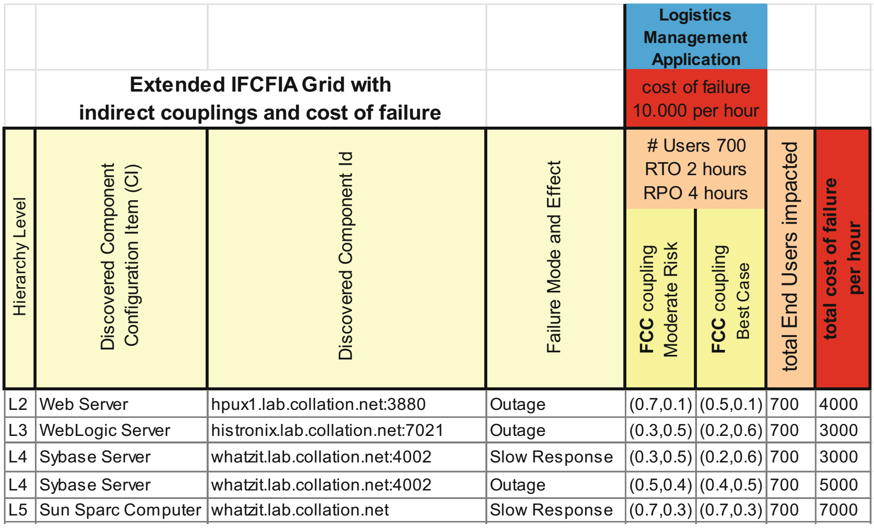

Logistics Management
(0.80, 0.10)
(0.75, 0.15)
HP-UX
hpux1.lab.collation.net
Apache
hpux1.lab.collation....3880
Apache
cleopatra.lab.collati...4580
Solaris
cleopatra.lab.collation.net
(0.50, 0.40)
(0.50, 0.40)
Solaris
histrionix.lab.collation.net
WebLogic
histrionix.lab.collati...7021
WebLogic
homeopathix.lab.col...7021
Solaris
homeopathix.lab.collati...
(0.55, 0.3125)
Solaris
whatzit.lab.collation.net
Sybase
whatzit.lab.collation...4002
Microsoft SQL
hades.lab.collation....1433
Windows
hades.lab.collation.net

Fig. 13 Logistics management intuitionistic application dependency map

through its direct or indirect dependants. Different types of impact analysis involve the usage of classical or probabilistic variants of the logical operations conjunction and disjunction in calculation of indirect impacts. Depending on which combination of operations will be used, the indirect impacts may be greater or smaller. Within a grid all data relevant for the loosely coupling assessment is shown including the business repair time targets and estimated cost of failures. In the following IFSFIA grid two attitudes are expressed leading to an optimistic (best case) or moderate (mediate case) assessment of the impact caused by an incident situation (Fig. 14).

The result of the IFSFIA analysis is an intuitionistic sorted fuzzy distribution of the components, providing an ordered set by the probability of incident root causes. It can be now a guide for discovering roots for SLA violations and to justify IT investments.



				Logistics Management Application			
	Extended IFCFIA Grid with indirect couplings and cost of failure			cost of failure 10.000 per hour			
				# Users 700 RTO 2 hours RPO 4 hours			
Hierarchy Level	Discovered Component Configuration Item (CI)	Discovered Component Id	Failure Mode and Effect	**FCC** coupling Moderate Risk	**FCC** coupling Best Case	total End Users impacted	**total cost of failure per hour**
L2	Web Server	hpux1.lab.collation.net:3880	Outage	(0.7,0.1)	(0.5,0.1)	700	4000
L3	WebLogic Server	histronix.lab.collation.net:7021	Outage	(0.3,0.5)	(0.2,0.6)	700	3000
L4	Sybase Server	whatzit.lab.collation.net:4002	Slow Response	(0.3,0.5)	(0.2,0.6)	700	3000
L4	Sybase Server	whatzit.lab.collation.net:4002	Outage	(0.5,0.4)	(0.4,0.5)	700	5000
L5	Sun Sparc Computer	whatzit.lab.collation.net	Slow Response	(0.7,0.3)	(0.7,0.3)	700	7000

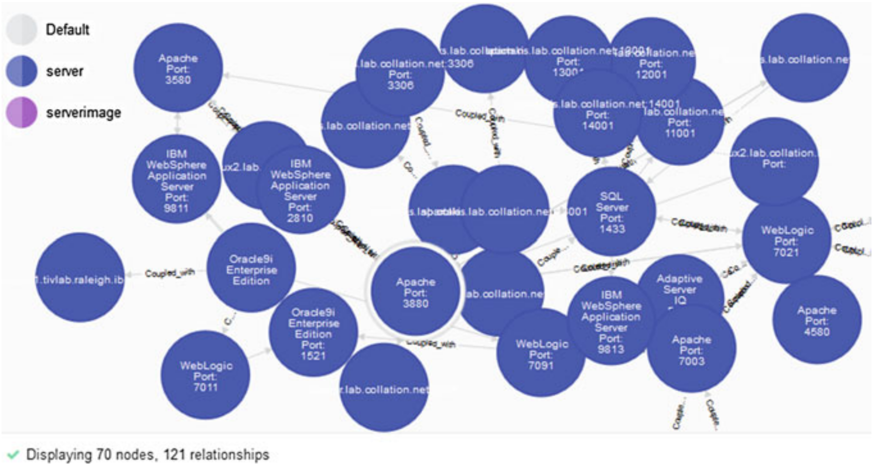
Fig. 14 Monitored failure modes with couplings and costs

5.2 *Indirect Impact Calculation and Visualization Using Python and Neo4j*

As an opposite to the widely known SQL databases, graph databases like Neo4j do not store their information in tables, but rather use graphs consisting of edged and vertices i.e. nodes and relationships to store information. While this approach is not appropriate for all kinds of data, it is a lot more convenient and easier to use, when it comes to graph data that does already consist of data objects and relationships between them. For calculating indirect dependencies in a server networks, graph databases suit perfectly well, since the given data is already in shape of a connected network and actions like path-finding, which are required for the impact calculations, are already implemented in the used graph database Neo4j.

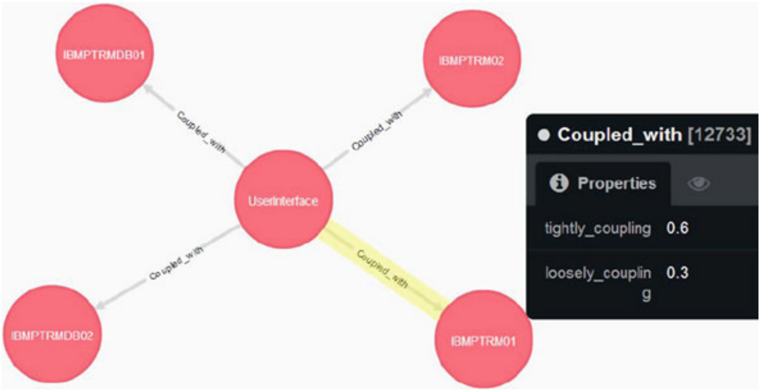
The following image shows the discovered servers of the Logistics Management application including the fuzzy intuitionistic direct impact loaded into the Neo4j database (Fig. 15).

Being able to calculate the indirect dependency index for the discovered network, the impact of any component to any other can be expressed as fuzzy intuitionistic indirect impact by either getting the direct coupling for adjacent servers or calculating the indirect coupling based on the chosen IFS operations. To present the results to the user, the Neo4j browser is used, where a temporary graph is inserted into the database, which forms a star showing the chosen service in the center and all other components connected to it with the calculated indirect coupling levels (Fig. 16).



✓ Displaying 70 nodes, 121 relationships

Fig. 15 Loaded components and direct dependencies into Neo4j



● Coupled_with [12733]

Properties

tightly_coupling	0.6
loosely_couplin g	0.3

Fig. 16 Star representation of indirect dependencies

6 Conclusions and Ongoing Work

Managing the quality of virtualized, distributed and multi-tiered services is a key challenge in today’s service management. Traditional approaches are measured bi-modal (means either operate correctly or fail) and concentrate on local technical IT performance measurements rather than with business-oriented service achievement. There are some more advanced approaches [22], including proposed models of QoS ontologies [23] or works that are based on Fuzzy Rules [3] e.g. Performance Relation Rules and Artificial Intelligence. The novelty of our approach lies in an integrated step-wise methodology, including automated information assimilation, support of gradual failures or service degradations (e.g. predicting a percentual SLA achievement) and bi-polar fuzzy intuitionistic impact assessments. Combining academic research with practice oriented business scenarios by expanding IT reliability engineering with fuzzy mathematical models provides high value to the service business, especially as the framework is general enough to be applied to any type of IT service. In this paper we presented an intuitionistic fuzzy methodical framework, which can be used to granularly relate performance metrics of the backstage in a service orchestration to the metrics used within Service Level Agreements. This model about a set of fuzzy-related components 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 to business services. Further, it can be a guide in the process of discovering the root cause of SLA violations and may help to provide more accurate analyses that are needed to make appropriate adjustment decisions at runtime. Within ITIL v3 best practices IFSFIA can help Configuration Management and Problem Management processes can benefit from advanced root cause determination and impact assessments. The proposed IFSFIA framework enables 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 on the business and vice versa.

In the ongoing work, we seek to validate the framework by applying it to larger amounts of historical and monitored usage data of datacenter environments compared to frontend quality parameters and business SLA's. Also these research ideas are implemented with prototypes that supports the steps of data assimilations, the indirect impact calculations and the visualization of the couplings within the dependency graph. Further the prototype can be extended to solicit rules based on the derived impacts to predict effects of incidents on the business services.

Acknowledgment A scientific paper organically evolves from a primeval soup of ideas originating from several individuals past and present. In this light, I thank all who made this paper possible, my first advisor, Andreas Meier, for supporting me to conduct scientific research after years deep in practical service business. Second Boyan Kolev and Ivaylo Ivanov for feeding the great idea how to perform indirect fuzzy dependency calculations and Krassimir Attanassov for inventing the principal concept and mathematical foundation about intuitionistic fuzzy sets.

References

1. Schütze, R.: ICIFS Conference Nov. 13 Sofia, Bulgaria—Intuitionistic Component Failure Impact Analysis "Notes on IFS", **19**(3), 62–72 (2013)
2. Li, H.: Challenges in SLA Translation—SLA@SOI European Commission Seventh Framework Programme (2007–2013) SAP Research (2009)
3. Joshi, K.P., Joshi, A. Yesha, Y.: Managing the quality of virtualized services. In: Proceedings of the SRII Service Research Conference (2011)
4. Tele Forum: SLA Management Handbook—Vol. 4: Enterprise Perspective ISBN: 1-931624-51-8
5. Dhama, H.: Quantitative models. J. Syst. Softw. **29**, 65–74 (1995)

6. Fenton, N.E., Melton, A.: Deriving software measures. J. Syst. Softw. **12**, 177–187 (1990)
7. Alghamdi, J.S.: Measuring software coupling. In Proceedings of the 6th WSEAS Conference on Software Engineering (2007)
8. Quynh, T.: Dynamic Coupling Metrics for Service–Oriented Software 2009. University of Science, Hanoi Vietnam (2009)
9. Craig, J.: EMA Radar for application discovery and dependency mapping (ADDM): Q4 2010 summary and vendor profile (2010)
10. Chen, X., Zhang, M., Mao, Z.M., Bahl, P.: Automating Network Application Dependency Discovery: Experiences, Limitations, and New Solutions (2008)
11. Van Haren: Continual Service Improvement Based On ITIL V3 ISBN 9087532407 (2008)
12. Three reliability engineering techniques, IBM Systems Journal, Vol 47, No 4, 2008
13. Yang, H.Y.: Measuring indirect coupling. Doctor Thesis, University of Auckland, New Zealand (2010)
14. Atanassov, K.: On Intuitionistic Fuzzy Sets Theory (Studies in Fuzziness and Soft Computing), 1st edn. Springer, Berlin (1999 edn: 2010)
15. Zadeh, Lofti, Computing, Soft, Logic, Fuzzy: IEEE Softw. **11**(6), 48–56 (1994)
16. Kolev, I.: Fault Tree Analysis in an Intuitionistic Fuzzy Configuration Management Database 2009. In: International Conference on Intuitionistic Fuzzy Sets, Sofia Bulgaria (2009)
17. Jacob, B., Adhia, B., Badr, K., Huang, Q.C., Lawrence, C.S., Marino, M., Unglaub-Lloyd, P.: IBM Tivoli Application Dependency Discovery Manager: Capabilities and Best Practices. IBM Redbook, Poughkeepsie (2008)
18. Kieninger, A., Berghoff, F., Fromm, H., Satzger, G.: Simulation-based quantification of business impacts caused by service incidents. In: Proceeding of the Fourth International Conference on Exploring Service Science (Portugal), LNBIP vol. 143, pp. 170–185. Springer (2013)
19. Kosko, B.: Fuzzy cognitive maps. Int. J. Man Mach. Stud. **24**, 65–75 (1986)
20. Stylios, C., Georgopoulos, V., Groumpos, P.: The Use of Fuzzy Cognitive Maps in Modeling Systems. Department of Electrical and Computer Engineering, University of Patras (1997)
21. Schütze, R.: IFCFIA, a gradual method for SLA dependency mapping and bi-polar impact assessment. Technical report 07/2013, Information Systems, University of Fribourg, Switzerland, (2013)
22. Robak, P.: Fuzzy modeling of QoS. J. Comput. Sci. **16**, 69–79 (2008)
23. Sora, T.: Translating User Preferences into Fuzzy Rules. Politehnica University, Romania (2009)