Modeling and Reconfiguration of critical Business Processes for the purpose of a Business Continuity Management respecting Security, Risk and Compliance requirements at Credit Suisse using Algebraic Graph Transformation

Christoph Brandt
Computer Science, SECAN-Lab
Université du Luxembourg
Luxembourg, Luxembourg
Email: christoph.brandt@uni.lu

Frank Hermann
Theoretische Informatik/Formale Spezifikation
Technische Universität Berlin
Berlin, Germany
Email: frank@cs.tu-berlin.de

Thomas Engel
Computer Science, SECAN-Lab
Université du Luxembourg
Luxembourg, Luxembourg
Email: thomas.engel@uni.lu

***Abstract*—Critical business processes can fail. Therefore, continuity processes are needed as backup solutions. At the same time business processes are required to comply with security, risk and compliance requirements. In the context discussed here, they should be modeled in a decentralized, local and declarative way, including methodological support by tools.**

By discussing a simplified loan granting process in the context of a Business Continuity Management System at Credit Suisse, we show how algebraic graph transformation can contribute a methodologically sound solution being compatible with all these requirements in a coherent way. As a consequence significant benefits of automation and quality can be realized. The presented contribution is theoretically sound and implementable by the people in the field.

***Keywords*-business continuity management, algebraic graph transformation, event-driven process chains, security, risk, compliance**

I. Introduction

The problem statement seen from the empirical study of Knight and Pretty [13] is about the reconfiguration of business processes (BPs) to cope with possible failures. However, not all implementations of a Business Continuity Management System (BCMS) have shown to be successful. Therefore, we are interested in a methodological solution that can be evaluated in advance, delivering a relevant contribution in comparision to the best-practices used today.

The research question therefore is about the generation of all continuity processes – that are respecting security, risk and compliance side-constraints – given a critical business process and its continuity fragments. The purpose is to enable an optimal choice of optimal continuity processes and to enable case-based decisions.

This paper presents contributions in the area of business continuity management (BCM) with respect to security, risk and compliance and in the area of algebraic graph transformation (AGT). Given a declarative process model and continuity snippets all possible continuity processes that respect given side constraints can be generated. So, for all combinations of modeled failures it is possible to check if sound continuity processes are available. Therefore, it can be tested beforehand if a BCMS is complete as a whole. In doing so, the way of modeling and the nature of models are kept fully compliant with business requirements of Credit Suisse. The solution is required to be fully declarative, minimal, decentralized, formal (in a transparent way) and automatable at the same time. From the point of view of theory AGT analysis techniques are specialized for the given class of problems.

The paper is organized as follows: Firstly, we show how laws, regulations and rules can be mapped to the notion of security, risk and compliance. Secondly, we introduce the notion of a BCMS and reflect the corresponding situation at Credit Suisse (CS). Thirdly, we present a simplified version of a loan granting process (LGP) as an example of a critical BP at CS and draw the link to an underlying BCMS. Forthly, we demonstrate how the set of continuity processes based on the given loan granting processes and continuity fragments can be generate that respect given side-constraints. Finally, we draw our conclusions, point to issues of future work and mention some related work.

II. Laws, Regulations, Rules

Laws, regulations and rules determine the degree of freedom and the possible boundaries a bank can exploit or has to respect. In the context of this study we like to put our focus on concrete requirements regarding security, risk and compliance derived from laws, regulations and rules. In a first step, we will present today's understanding in the banking environment which is best-practice driven. In a second step, we will point out our understanding which is more aligned towards formal methods.

A. Security

From a best-practice point of view at CS, security can be understood as a set of services. These services encompass the protection of persons, assets, physical property, handling policy violations, as well as IT security related issues, etc.

978-1-4244-5564-5/09/$26.00 ©2009 IEEE

From a methodological point of view, we consider security as everything that can be proven based on sound models. As an example the separation of duties is presented next.

1) Separation of duties: The separation of duties is a special security requirement. Its primary objective is the prevention of fraud and error. It can be illustrated as a requirement of two distinct signatures on a contract [7].

Its monitoring and enforcement as a best practice can be realized by the help of an organizational policy that requires contracts to be signed by different persons.

Its monitoring and enforcement from a methodological point of view can be realized already when building organizational models. Therefore, it comes along as a side constraint during the modeling process. Because models can be build using AGT, such requirements can be automatically enforced as graph constraint (GC) checks on the abstract syntax of (formal) models.

B. Risk

Credit Suisse considers different types of risk: market risks, credit risks and operational risks. Here, we only focus on operational risks. An operational risk encompasses inadequate or failed business processes, people or systems caused by certain events that lead to financial losses.

From a best practice point of view, operational risks are managed by organizational solutions like committees and forums, processes and standards, indicators, reports, audits, analysis of loss data, estimation of required risk capital, etc.

From a methodological point of view, risks can be much better investigated using simulations of possible failures and their consequences based on sound organizational models.

We claim that the case of business processes failures can be backuped to a certain extent by continuity procedures in the context of a BCMS.

1) Continuity Procedures (CP): We define CPs as special micro business processes that are put in place in case that an IT application, a person or a database is not available. It is usually a work-around to guarantee a minimum availability of business services or a certain quality of service.

2) Business Continuity Management (BCM): According to CPs, we define a BCM to be a special management function that takes care of continuity planing to ensure that a bank is not going out of business in case of major failures in one of its critical business processes.

C. Compliance

From a best practice point of view at CS, compliance means conforming to a specification or policy, standard or law. A famous example in this context is the Sarbanes-Oxley Act [19] which is about the accuracy of financial statements and the corresponding top management responsibility.

From a methodological point of view we define compliance as a relationship between certain norms and organizational models and as a relationship between organizational models and the real-world situation. Because a real-world situation does not have to be in line with a model, tests need to be performed.

As examples, we like to point, first, to the behavior of people inside the bank regarding information barriers and, second, outside the bank regarding agreed payment plans.

1) Information Barriers: Information barriers exists between different divisions of a bank for various purposes. At CS such divisions are investment banking, asset management, private banking and shared services. The main purpose is to make sure that confidential information is not passed from the private side of the bank to its public side.

2) Payment Plans: Payment plans in the context of granted loans need to be monitored to see if clients actively comply to the plans.

III. Business Continuity Management

Business Continuity Management (BCM) is introduced here according to the BS 25999 [5] by taking two different perspectives: the first is a general one, the second a specific one, focussing on the concrete situation at Credit Suisse.

From a general perspective BCM is build on the code of practice, the BS 25999-1:2006 standard, introducing the notion of a BCMS. Key elements of a BCM are the notion of a disaster, of a business risk, of a critical business process, of a disaster recovery plan, of a business continuity plan, and of the maximum tolerable period of disruption (MTPD).

A disaster is an unforeseen event having a disruptive impact on a critical business process of a company. As such, it can cause a business risk to a company's business model. Therefore, a business process interrupted by a disaster is named critical if a company goes out of business in case the process does not recover early enough. How the recovering has to happen is defined by a recovery plan. In contrast to that, a continuity plan does backup the critical business process during that time. The backup process must be up and running by the time span defined as MTPD.

Credit Suisse's solution today is primarily an organizational solution. Because of its organizational and informal nature and because continuity fragments are not available as such, no optimization can be performed and case based decisions cannot be supported.

IV. A Loan Granting Process

We assume that a loan granting process (LGP) is a critical business process a bank is running. From the point of view of this paper a possible LGP is simplified to match the scope of this study. It encompasses a client (C), a relationship manager (RM), a credit advisor (CA) and a credit officer (CO). The process is characterized by steps that are performed manually, steps that are executed automatically and steps that are hybrid. The view on the process is the one taken from a workflow engine that runs workflow instances based on their workflow scheme. The notation used is the one

for event-driven process chains, but in a slightly modified version to fit the requirements of the presented scenario in a better way. The process itself covers the whole lifespan of a granted loan, starting with the demand for a loan, ending with its finished payment plan.

We assume that, once a client (C) arrives at the bank, he will be asked a couple of things by the RM (functions F1-F4 in the workflow model in Fig. 2). Firstly, he needs to identify himself and the RM will try to make a first estimation about the possible customer value in an assumed business relationship. Secondly, the RM will record the client's demand. All data is entered into IT systems by the RM. In the following the credit worthiness and a customer rating for C is calculated automatically by two different applications (F5-F6). Based on the rating the CA will make a decision if C will be accepted for a loan (F7). In the next step, an optimized product is created by the CA and the RM for C (F8). Afterwards, the RM creates a contract for C (F9). This contract has to be signed by C and the RM. The credit officer (CO) needs to approve the contract (F10-F12 in Fig. 3). Here the 4-eye principle applies for security reasons because the RM and the CO are not allowed to be the same person. Afterwards, the bank pays the granted loan to C, and C is paying the credit back according to a payment plan up to the moment the contract will be closed (F13-F15).

From the point of view of a BCM this loan granting process can be discussed looking at certain failures that can happen in the process. Because we like to introduce fully automated continuity techniques respecting security, risk and compliance issues as real-world side constraints the perspective on the process is the one of a workflow engine that implements these techniques. Further, we like to base our discussion on a running process instance, not only on the underlying process scheme as it is done today in the context of BCMS. This leads to highly optimized reactions towards certain failures in a critical business process. We assume that for most steps in a critical business process continuity fragments are available to backup those steps. We further assume that the elements that can fail in a process are people, applications and databases. Depending on concrete failures, a workflow engine can select a continuity process based on the calculated set of all continuity processes.

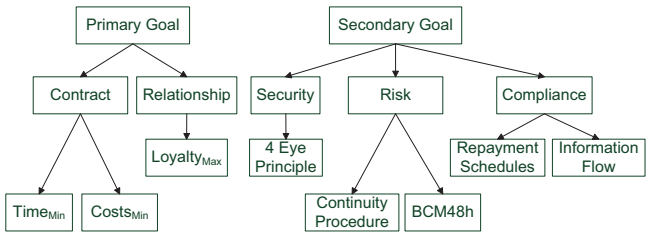

Figure 1. Primary and Secondary Objectives

The presented BP has to meet certain objectives. There are primary and secondary ones. From the point of view of CS, this process needs to generate a contract as well as a long-term client-bank-relationship. The time consumption and the realized costs are the relevant attributes regarding the contract, while the measurable client loyalty is the relevant relationship attribute. Furthermore, certain security, risk and compliance requirements have to be respected.

V. Analysis and Continuity Processes

Event-driven process chain models (EPCs) [20] are widely used for the modeling of business processes. In this section we show how the critical process presented in the previous section can be analyzed on the basis of its EPC model. Thereafter, we present the generation of the continuity processes using an additional set of alternative fragments for particular failures that may occur. Exemplarily, security requirements are checked and we explain how risk and compliance should be analyzed. The full model with all details can be found in [4].

A. Modeling

Standard EPCs specify possible executions of business functions. They consist of the business functions itself, the organizational entities resp. applications performing the functions, the events that trigger the order of the business functions and the data elements which are involved. In our scenario we consider a business process which is partly supported by IT, i.e. by a workflow engine. As such we extend the model by specifying which data is locally cached within the workflow engine and we make explicit where each data element is stored and from where it is accessed. Storage units are either databases, or, alternatively, organizational entities, like concrete persons. We name these extended EPCs workflow engine based and data-flow oriented EPCs - in short WDEPC.

Figures 2 and 3 show the WDEPC language artifact for the presented loan granting process: WDEPC LG. The diagrams are divided into five columns. Starting on the left, there are data storage units. The next column contains the corresponding data items, where solid lines indicate that the data is also locally cached within the workflow engine. This allows us to cover automated and non-automated parts of a business process by one single declarative process model. The third column consists of the business functions, the fourth contains the events and finally the fifth column consists of the organizational entities, i.e. persons or software applications.

B. Graph Grammar for a WDEPC

In the following, we describe the construction of a graph grammar GG to define the operational semantics of the LGP. Thereafter, we show how dependencies that are not caused by the events but by the dependencies on data elements and actors are computed using the constructed grammar.

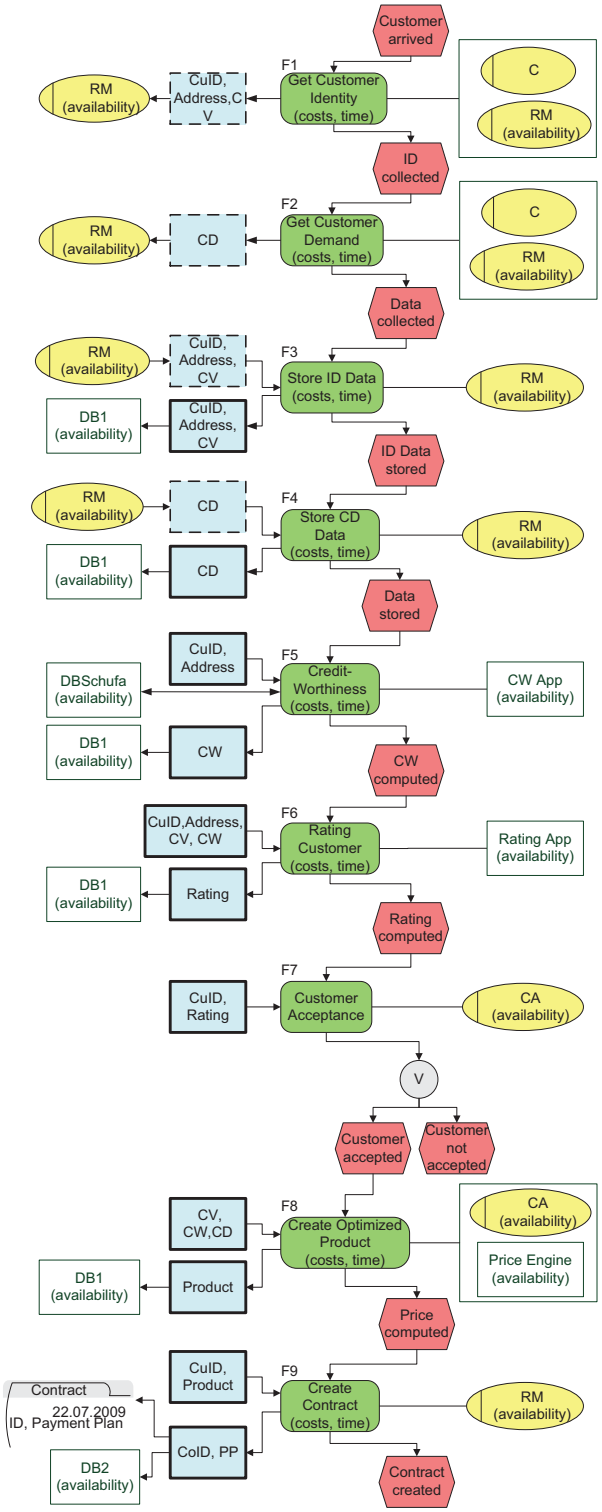

Figure 2. Workflow Part 1 of LG

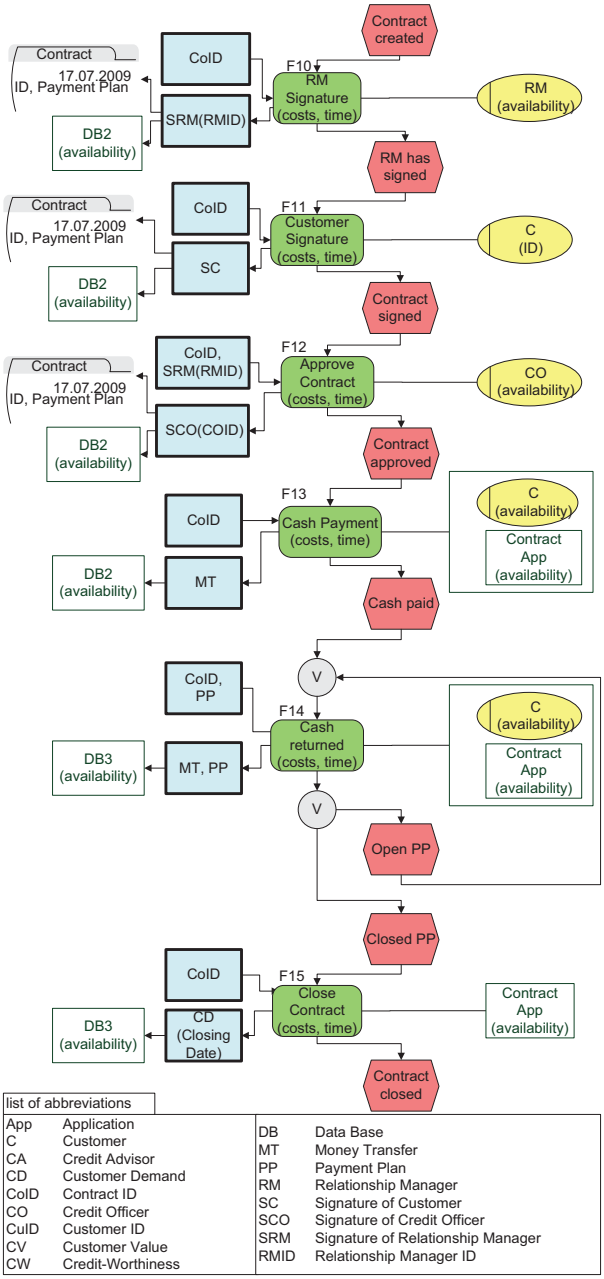

Figure 3. Workflow Part 2 of LG

From the formal point of view a graph grammar $G = (TG, RG, SG)$ consists of a type graph TG, a set of graph transformation rules RG and a start graph SG [9]. The type graph specifies the structure of possible graphs, the rules constructively define how graphs are modified and the start graph is the starting point for each transformation sequence. A graph $G = (V, E, src, tgt)$ is given by a set of vertices V, a set of edges E and functions $src, tgt : E \to V$ defining source and target nodes for each edge. Graph morphisms $m : G_1 \to G_2$ specify relations between graphs, where $m = (m_V, m_E)$ consists of a mapping m_V for vertices and a mapping m_E for edges, which have to be compatible with the source and target functions. A graph transformation rule (production) $p \in RG$ is given by a graph L (left hand side), a graph R (right hand side) and an intermediate graph K with two injective graph morphisms $l : K \to L$ and $r : K \to R$ specifying the mappings between L, K and R. Applying a

rule p to a graph G means to find a match m of L in G and to replace this matched part $m(L)$ by R, which is performed in two steps according to the double pushout approach [9], where first deletion and then creation of nodes and edges is performed. In the present scenario our generated rules do not delete anything ($L = K$), thus we only depict L and R as in Fig. 4.

Given a workflow model in form of a WDEPC the corresponding graph grammar $GG = (TG, R, SG)$ is reconstructed as follows:

- The type graph TG contains the nodes and edges of the WDEPC except the event nodes and its adjacent edges, where nodes with the same label that occur several times in the WDEPC occur only once in TG.
- The start graph SG consists of the nodes for the actors, i.e. the organizational entities, and the resources only.
- Each function is translated into a graph rule (see e.g. function "Store CD Data" Fig. 2 and its corresponding rule in Fig. 4). The left hand side of the rule contains the actors, the input data elements with its resources and the edges between these elements. The right hand side additionally contains the function node, the output data elements, and the edges that connect the nodes as given in the WDEPC.

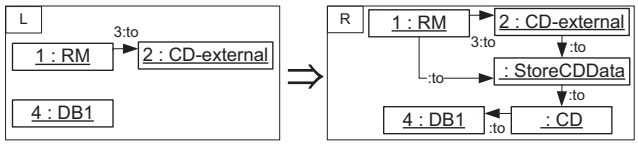

Figure 4. Rule $storeD$

Fig. 4 shows the rule $storeD$ of the grammar GG_{LG}, which specifies the function "Store CD" of the WDEPC LG. The effect of the rule $storeD$ is the creation of a node with type "StoreCDData", which corresponds to the process function "Store CD Data", where CD abbreviates "Customer Demand". Furthermore, edges are inserted to connect the new function node with its actors and data elements. The numbers in the rule denote the mappings between L and R.

A WDEPC can be simulated by applying the generated rules to the start graph according to the order in the WDEPC. Each intermediate graph represents the current state for the execution of the process and each rule application ensures that not only the necessary input data elements are visible but also that they are visible trough the involved resources to which the particular actor has access.

The dependencies between the functions of an WDEPC can be analyzed by the dependencies between the rule applications. Since the derived graph grammar of our example fulfills the additional conditions of a subobject transformation system - a graph grammar, where each rule component is injectively typed - we can apply efficient techniques especially developed for the analysis of dependencies in processes [8], [12].

C. Computation of Dependencies

In the following we explain the analysis of dependencies for the process LG. Consider the first four functions "Get Customer ID", "Get Customer Demand", "Store ID Data" and "Store CD Data". The only dependencies for the corresponding rules $p_1 = getID$, $p_2 = getD$, $p_3 = storeID$ and $p_4 = storeD$ in GG_{LG} are: $p_1 <_{rc} p_3$ and $p_2 <_{rc} p_4$, where "rd" denotes read causality. This means that p_3 uses a structure that is created by p_1 and p_4 uses structures that are created by p_2. Now, the WDEPC LG requires a sequential execution. However, the dependencies based on the rules also allow that first the demand of a customer is determined and stored while the necessary identification information is collected and stored thereafter. This means that the four steps can be executed in several ways - all together 6 variants - only the partial order given by the dependency relation $<_{rc}$ has to be respected. The relation manager shall be able to act upon the customer preferences and upon the course of conversation, such that any of the possible interleavings should be possible. Of course, the possible interleavings can also be achieved by modifying the EPC, but during the modeling of an EPC for a business process several possibilities of concurrency will not be detected, because the real actors are asked to specify the standard execution.

Now, have a look at the end of the example process LG where functions "Customer Signature" and "Approve Contract" occur. The corresponding rules are $p_{11} = customerS$ and $p_{12} = approve$. There is no dependency between these rules implying that the customer may sign the contract before or after the contract is approved by the credit officer. Consider the case that the customer may want to see both signatures on the contract before he signs. Thus, this inverse order is relevant. Note that it is not trivial to find this partial independence while building an EPC model by hand.

D. Computation of Alternatives

In order to construct complete continuity processes for a combination of failures we first show how process fragments are replaced and composed: Consider that we have process parts P_1 and P_2. They are composable, if the start event of P_2 occurs in P_1 and we glue the processes together at this event resulting in the new part $Q = P_1; P_2$. In order to ensure that Q can be executed we check that each left hand side of a rule p_x of the corresponding grammar GG_Q is included in the start graph joined with the right hand sides of the rules that correspond to the applied preceding steps. This condition is sufficient, because the constructed rules do not delete anything. If P_1 is already executable then the check can be reduced to the rules of GG_{P_2}.

As soon as a resource or an actor is not available the process execution has to be replaced by an alternative execution sequence, which contains suitable alternative process parts, such that the alternative execution is possible and fulfills all requirements. Consider the following failure in the

present scenario: the rating application in the WDEPC "LG" is not available, which implies that the function "Rating Customer" cannot be executed. In this case the alternative function "Rating Customer (C)" in Fig. 5 can be executed, where "(C)" denotes that it is a continuity function for a certain failure of a resource. Exchanging function "Rating Customer" with function "Rating Customer (C)" may cause conflicts with other functions. The underlying dependencies with respect to the other functions of the current chain of process steps can be analyzed using the corresponding graph transformation rules. This analysis can be performed statically, i.e. before a failure occurs, and the results can be stored and remain valid during a process execution.

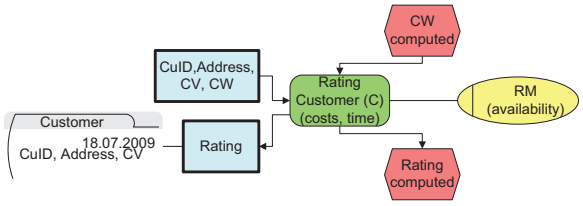

Figure 5. Alternative Function "Rating Customer (C)"

"Rating Customer (C)" needs the availability of "CuID, Adress, CV" and "CW", which are provided by the functions "Get Customer Identity" and "Credit Worthiness". These dependencies are present for the corresponding rules $p_1 = getCID, p_5 = creditWorthiness, p_6 = ratingCustomerE$. Furthermore, we have to ensure that all elements that are necessary for the succeeding steps of "Rating Customer (C)" are present. Thus, we have to ensure that each element, that is created by function "Rating Customer" is either created by "Rating Customer (C)" as well or not needed by a succeeding step. The complete business continuity process is constructed stepwise and for each step the following condition (1) ensures that the succeeding steps can access the elements they need. In more detail, the rule $p_i = (L_i \leftarrow K_i \rightarrow R_i)$ of condition (1) below corresponds to the $i(th)$ function of an WDEPC and p_i shall be replaced by the alternative rule $p'_i = (L'_i \leftarrow K'_i \rightarrow R'_i)$. The elements in the set $(R_i \setminus K_i)$ are the nodes and edges that are created by the rule p_i.

$$\left[(R_i \setminus K_i) \cap \bigcup_{j>i} L_j\right] \subseteq R'_i \setminus K'_i \qquad (1)$$

Fortunately, this condition is fulfilled for "Rating Customer (C)" in Fig. 5 and we can use this fragment. Furthermore, independent succeeding steps can be moved to preceed the critical function, which delays the execution of the continuity fragment - e.g. in Fig. 6 the steps $a7$, $a8$ are moved in front of $a6$, which is going to be replaced by $a6'$. If the missing resource is available again and the delayed function is still not executed then the original function can be executed instead. This is an important advantage of the automatic analysis capabilities and the generation of possible continuity processes.

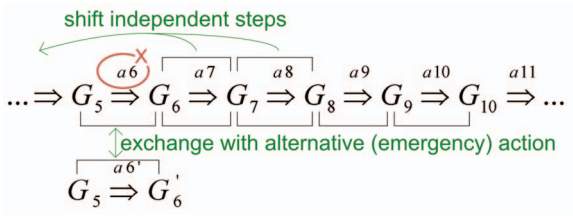

Figure 6. Automatic generation of alternatives

Alternative process parts may contain several steps that furthermore may only replace parts of the original steps or cause conflicts with other steps, which implies that additional alternatives have to be used to build up a complete alternative. In Fig. 7, two alternative parts are composable with the original process by exchanging it with steps $a1$ to $a4$. The step $a2$ is not completely covered by one alternative fragment but by the composition of the two fragments. In order to find optimal continuity processes annotated costs and time values of the functions can be used.

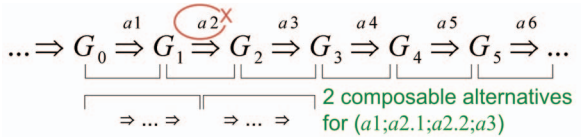

Figure 7. Complex alternatives

E. Validation of Objectives

In order to validate that the non-functional objectives are fulfilled by the generated alternative processes the requirements are vizualized and formalized as graph constraints. They are checked automatically to be fulfilled by the formal graph model. Consider for example the security requirement that the credit officer who approves the contract shall not be the same person as the relationship manager that also signs the contract. In the WDEPC LG both persons are distinguished by their names. Thus, we have to ensure this property on the instance level, i.e. when the process is executed by a workflow engine. In this situation we can check the identities of the objects, which are concrete actors in the process execution and we analyze the derived grammar, where all actors are of type "Person".

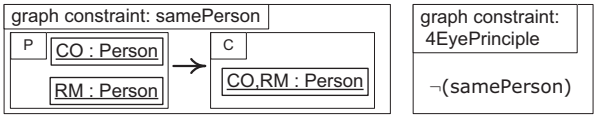

Figure 8. Graph Constraint: 4 Eye Principle

Fig. 8 shows the graph constraint "4EyePrinciple" that ensures that for all intermediate states of the process we have that the credit officer and the relationship manager are different persons. The constraint "4EyePrinciple" is based on the basic constraint "samePerson", which we explain

first. The premise P specifies the pattern of an object structure with two persons. Its conclusion C requires that both persons are the same. More formally, a graph G fulfills this basic constraint, if for any occurrence of P (given by a morphism $p : P \rightarrow G$) we have that there is also a compatible occurrence of C (given by an injective morphism $q : C \rightarrow G$, such that $p = q \circ c$ for the constraint morphism $c : P \rightarrow C$). The constraint "4EyePrinciple" is the negation of "samePerson", which means that it is not allowed that the two roles with labels "CO" and "RM" are the same person. In this context we require that the labels, which occur in the process, guide the matching for the intermediate graphs. However, these labels can also be specified as attributes of the node type "Person" to indicate the concrete role of a person. Those security rules can be defined declaratively. Furthermore, if a condition shall be ensured only locally, i.e. for a single function like "Approve Contract", the constraint can be formulated as an application condition for the corresponding rule [9].

Summing up, once a business process is modeled its corresponding graph grammar can be derived automatically, and graph constraint checks can be performed to ensure structural security requirements. Therefore, it can be proven that a certain security requirement is valid for a certain model. By adding snippets of continuity procedures, continuity processes can be generated. Technically, this is done using process composition based on algebraic graph transformation. Continuity processes can be created in general for all possible combinations of failures from the point of view of a workflow scheme, or on a case-by-case basis from the point of view of a running workflow instance. For every generated process alternative its satisfiability is checked. The positively evaluated process schemes can be used to simulate all kinds of failures and corresponding consequences of a process instance in terms of time, operational costs and financial losses. By doing this we are able to discuss risks from a methodological point of view, not only based on organizational best-practices. Therefore, we can make informed decisions about alternatives that are fully or partially respecting the side constraints regarding security, risk and compliance. Knowing all possible continuity processes for a given critical business process we can simulate BCM risks.

VI. Related Work

In [15] the importance of a resource and data driven analysis of business processes is stressed. But the authors do not deliver a formal solution suitable to be fully automated as requested by Credit Suisse (CS). In [21] disaster recovery plans are evaluated based on ARIS methodology. This solution does not show how to generate the full configuration space that fulfills possible side-constraints as requested by CS. In [10] an organizational solution to address information security management problems is presented. But this solution cannot be automated as requested by CS. In [16] and [6] the claim is made that continuity processes need to be checked for security, risk and compliance, and that BCMs and risk solutions should be soundly integrated. This claim is fully compatible with the view of CS. In [1] a solution using EPC to simulate processes regarding their risks and costs is proposed by the help of a goal-risk framework. But the complete process configuration space can not be generated and checked for side-constraints as requested by CS.

In [14] the workflow system AgentWork is able to support dynamic workflows based on event-condition-action rules. In contrast to that, CS requested that workflow adaptions should be handled based on declarative continuity snippets only. Given such snippets, we can apply our modification technique automatically. Therefore, such rules do not need to be specified. In [17] a solution guaranteeing the structural correctness of a process model is presented while applying dynamic changes. However, this case is different from the CS scenario where a set of continuity processes is generated in advance based on continuity snippets to enable optimizations and case based decisions, assumed that given side-constraints are respected. In [22] change patterns are proposed as a means to handle modifications of a workflow model and in [18] important correctness problems regarding general modifications are discussed in a comparative survey. In the present scenario already well-formed subprocesses are given. The presented generation technique composes these sub-processes in a controlled way, such that the well-formednes is preserved, which represents an important correctness issue. In addition to that, CS requested to check side-constraints. We do that by the help of graph constraint checks. Therefore, modification rules need not to be maintained and side-constraints can be modeled globally. In [3] a framework for service, process and rule models in the context of enterprise engineering is presented. The techniques presented in this paper are kept fully compatible with this approach as requested by CS.

In [11] and [2] the use of graph transformation and graph substitution techniques is discussed. However, our focus is different. The reconstructed graph grammar formalizes the operational semantics. So, there is no need to model dependencies. They can be automatically derived from the descriptive EPC model. Therefore, the overall modeling effort can be minimized as requested by CS.

VII. Conclusions and Future Work

BCMSs have to support the execution of alternatives for regular business processes in case of failures. For this purpose, these alternatives have to be modeled and maintained. However, the modeling of complete alternatives for all combinations of failures is not practicable and inconsistencies may easily occur. Furthermore, security, risk and compliance shall also be ensured for all these alternatives.

The presented solution dramatically reduces the necessary efforts and supports an automatic validation of the objectives

in an intuitive and formal way. Alternatives are generated automatically based on a set of declarative fragments that replace regular process parts for particular failures. Complete alternatives for combinations of failures can therefore be derived using the same set of fragments. Therefore, the presented technique is practicable, easy to maintain and supports a formal validation of the results. Future work will encompass the implementation of the presented graph techniques for process optimization and composition. It will further address more cases as well as their validation.

ACKNOWLEDGMENTS

We would like to thank Credit Suisse and the FNR for co-founding this work and we are grateful to the anonymous reviewers for their suggestions on how to improve the paper.

REFERENCES

[1] Y. Asnar and P. Giorgini. Analyzing business continuity through a multi-layers model. In *Business Process Management, Lecture Notes in Computer Science*, volume 5240, pages 212–227, Berlin/Heidelberg, 2008. Springer.

[2] D. P. Bogia and S. M. Kaplan. Flexibility and control for dynamic workflows in the worlds environment. In *COCS '95: Proceedings of conference on Organizational computing systems*, pages 148–159, New York, NY, USA, 1995. ACM.

[3] C. Brandt, T. Engel, and F. Hermann. Security and Consistency of IT and Business Models at Credit Suisse realized by Graph Constraints, Transformation and Integration using Algebraic Graph Theory. In *BPMDS 2009 and EMMSAD 2009, LNBIP 29*, pages 339–352, Berlin/Heidelberg, 2009. Springer.

[4] C. Brandt, T. Engel, and F. Hermann. Modeling and Reconfiguration of critical Business Processes for the purpose of a Business Continuity Management respecting Security, Risk and Compliance requirements at Credit Suisse using Algebraic Graph Transformation: Long Version. Technical report, Technische Universität Berlin,Fakultät IV, (to appear 2009). tfs.cs.tu-berlin.de/publikationen/Papers09/BEH09.pdf.

[5] BSi. Business continuity management. bsi 25999-1. British Standards Institution, 2006.

[6] S.-C. Cha, P.-W. Juo, L.-T. Liu, and W.-N. Chen. Riskpatrol: A risk management system considering the integration risk management with business continuity processes. In *Intelligence and Security Informatics*, pages 110 – 115, Taipei, 2008. IEEE.

[7] R. Chandramouli. Enterprise access policy enforcement for applications through hybrid models and xslt technologies. In *ICEC '04: Proc. 6th Int. Conf. on Electronic commerce*, pages 490–499, New York, NY, USA, 2004. ACM.

[8] A. Corradini, F. Hermann, and P. Sobociński. Subobject Transformation Systems. *Applied Categorical Structures*, 16(3):389–419, February 2008.

[9] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. EATCS Monographs in Theoretical Computer Science. Springer, 2006.

[10] J. H. P. Eloff and M. Eloff. Information security management: a new paradigm. In *Proc. research conf. of the South African Institute of Computer Scientists and Information Technologists on enablement through technology (SAICSIT '03)*, pages 130–136, Republic of South Africa, 2003. South African Institute for Computer Scientists and Information Technologists.

[11] P. Heimann, G. Joeris, C.-A. Krapp, and B. Westfechtel. DYNAMITE: dynamic task nets for software process management. In *ICSE '96: Proceedings of the 18th international conference on Software engineering*, pages 331–341, Washington, DC, USA, 1996. IEEE Computer Society.

[12] F. Hermann. Permutation Equivalence of DPO Derivations with Negative Application Conditions based on Subobject Transformation Systems. In *International Conference on Graph Transformation 2008 - Doctoral Symposium*, volume 16. ECEASST, 2009.

[13] Knight and Pretty. The impact of catastrophes on shareholder value. In *The oxford executive research briefings*, University of Oxford, Oxford, England, 1996. Templeton College.

[14] R. Müller, U. Greiner, and E. Rahm. AGENT WORK: a workflow system supporting rule-based workflow adaptation. *Data Knowl. Eng.*, 51(2):223–256, 2004.

[15] A. Nigam and N. S. Caswell. Business artifacts: An approach to operational specification. *IBM Systems Journal*, 42(3), 2003.

[16] G. Quirchmayr. Survivability and business continuity management. In *Proc. of the 2nd WS on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation (ACSW Frontiers '04)*, pages 3–6, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[17] M. Reichert and P. Dadam. ADEPT flex -supporting dynamic changes of workflows without losing control. *Journal of Intelligent Information Systems*, 10(2):93–129, 1998.

[18] S. Rinderle, M. Reichert, and P. Dadam. Correctness criteria for dynamic changes in workflow systems: a survey. *Data Knowl. Eng.*, 50(1):9–34, 2004.

[19] P. Sarbanes and M. Oxley. Public company accounting reform and investor protection act. Washington, 2002. Government Printing Office.

[20] A.-W. Scheer. *ARIS-Modellierungs-Methoden, Metamodelle, Anwendungen*. Springer, Berlin/Heidelberg, 2001.

[21] P. Sztandera, M. Ludzia, and M. Zalewski. Modeling and analyzing disaster recovery plans as business processes. In *Computer Safety, Reliability, and Security, Lecture Notes in Computer Science*, volume 5219, pages 113–125, Berlin/Heidelberg, 2008. Springer.

[22] B. Weber, M. Reichert, and S. Rinderle-Ma. Change patterns and change support features - enhancing flexibility in process-aware information systems. *Data Knowl. Eng.*, 66(3):438–466, 2008.