# Modeling and Reconfigurating 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

Christoph Brandt[1)], Frank Hermann[2)] and Thomas Engel[1)]

1 Computer Science, SECAN-Lab
Université du Luxembourg
Luxembourg, Luxembourg
Email: [christoph.brandt,thomas.engel](at)uni.lu,

2 frank(at)cs.tu-berlin.de
Institut für Softwaretechnik und Theoretische Informatik
Technische Universität Berlin, Germany

Bericht-Nr. 2009/_ _
ISSN 1436-9915

# Modeling and Reconfigurating 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

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(at)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. A Business Continuity Management System is a special management system that will define how to recover from such failures and specifies temporary work-arounds to make sure a company is not going out of business in the worst case. However, because today's implementations are primarily organizational best-practice solutions, their security, risk and compliance issues in such a recovery situation are mostly unknown. In contrast to that, algebraic graph theory can be used as a formal method supporting employees when running business processes need to be reconfigured to recover from specific failures. The example discussed is a loan granting process in a real-world banking environment. Because such a process has to respect certain laws, regulations and rules even in emergency situations, we sketch how this can be done during the process reconfiguration by looking at security, risk and compliance issues, compatible with the graph technique.

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

## 1 Introduction

The problem statement can best be described by the empirical study of Knight and Pretty [1]. They show that companies that implemented a Business Continuity Management System (BCMS) are in a better position to survive a disaster that interrupts one of their critical business processes. However, a company's chance to survive is not guaranteed. Sometimes companies fail to recover from a disaster even so they have implemented a BCMS. In the highly regulated environment of banks certain laws, regulations and rules need to be respected as a side constraint even in such a situation which is, to our knowledge, not solved by today.

The research question derived from this situation is about how an effective and efficient BCMS can be put in place that fullfills security, risk and compliance issues derived from the laws, regulations and rules. This question is discussed based on a loan granting process running in the real-world banking environment at Credit Suisse (CS). The challenge is to generate all continuity processes to a given critical business process and its continuity fragments, such that security, risk and compliance side-constraints are respected. 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 and we sketch how these qualities can be measured based on a modeled business process. Secondly, we introduce the notion of a business continuity management system and reflect the corresponding situation at CS. Thirdly, we present a simplified version of a loan granting process as an example of a critical business process at CS and draw the link to an underlying BCMS. Forthly, we give a short introduction to algebraic graph transformation and reference subobject transformation systems that we are going to use to reconfigure the loan granting process in case of a failure of one of its parts. Fifthly, we demonstrate how a concrete loan granting process model can be analyzed and optimized in an efficient and effective way that fulfills security, risk and compliance issues as a real-world side constraint. Beyond that, we illustrate how case based decisions can be supported. Finally, we draw our conclusions, point to issues of future work and mention some related work.

# 2 Laws, Regulations, Rules

Laws, regulations and rules determine the degree of freedom and possible boundaries a bank can exploit or needs to respect. They do limit or enforce certain actions and organizational structures.

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. We use this understanding in the following sections to discuss the handling of a loan granting process by the help of an implemented business continuity management system.

## 2.1 Security

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

From a methodological point of view, we consider security as everything that can be proven based on sound models. We assume that the organizational models and corresponding methods are selected and combined in a way that enables fully automated modelchecking.

As an example the separation of duties is presented next.

### 2.1.1 Separation of duties

The separation of duties is a special security requirement. Its primary objective is the prevention of fraud and error. This is realized by disseminating the tasks and associated privileges for a business process among several persons. It can be illustrated as requirement of two signatures on a contract [2].

Its monitoring and enforcement from a best practice point of view can be realized by the help of an organizational policy that defines that no person should handle more than one type of function and that requires contracts to be signed by two different persons. Such a policy is reviewed, enforced and monitored by a security organization.

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 algebraic graph transformation, such requirements can be automatically enforced as graph constraint checks on the abstract syntax of formal models [3]. This methodological approach has some advantages like better quality assurance and better scalability than the best practice approach. Cognitive business process models [3] that are aligned with their formal counterparts can easily be used by a workflow engine to monitor security rules which is more efficient than using an additional organizational security structure.

## 2.2 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 emergency procedures in the context of a business continuity management in particular.

### 2.2.1 Emergency Procedures

We define emergency procedures (EP) 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.

### 2.2.2 Business Continuity Management

According to EPs, we define a business continuity management to be a special management function that takes care of emergency planing and handling to ensure that a bank is not going out of business in case of major failures in its critical business processes.

## 2.3 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 [4] which is about the accuracy of financial statements and the corresponding top management responsibility. It is not always fully defined of how to comply to certain regulations.

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 to check whether the concrete situation always conforms to the model.

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

### 2.3.1 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.

For example, a person belonging to the investment banking can act as a broker-agent for a client on behalf of a person belonging to the private banking. According to usual information barriers, such an agent is only allowed to access client data relevant to the transaction but it might be the case that he does access other data too.

### 2.3.2 Payment Plans

Payment plans in the context of granted loans need to be monitored to see if clients actively comply to the plans. A plan as a business model does not assure that the concrete behavior of a client will conform to it.

# 3 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 Business Continuity Management System (BCMS). The British Standard Institution (BSI) updated this standard using feedback from industry. The BS 25999-2:2007, published November 2007, summarizes the specifications for a BCM. According to Boehmer [6] more than 5000 industrial ideas have been incorporated during this update, this standard is setting out a high level of maturity. 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). We further define the maximum acceptable outage (MAO), the recovery time to objective (RTO), and the time to recover (TTR).

A disaster is an unforeseen event having a disruptive impact on a critical business process of a company. A business risk is the risk that a disaster potentially has on the business model of a company. A critical business process of a company is a business process the company is running that if interrupted and not recovered in a certain time span causes the company to go out of business in the worst case. A disaster recovery plan of a company is a plan that defines how to recover from the failures in a critical business process that have been caused by a disaster. A business continuity plan is a plan that is applied after the disruption of a critical business process to deliver a minimum level of business activity in order to guarantee the survival of the company during the time the business recovery process is executed.

The MTPD is the maximum time a company can survive without a minimum level of business activity. The MAO is the maximum period of downtime within which the business activities and functions must be fully recovered before the outage compromises business objectives and overall operational viability. The RTO is the maximum time allowed following disaster declaration to return the failed business and IT processes to a minimum level of activity. The TTR is the time taken to fully recover the IT and Business processes.

The BS 25999 is reactive in nature. It comes into play once a catastrophic event has happened. An important control used in the context is the MTPD that defines the tolerable downtime between the disruption of a critical business process and the availability of a minimum level of business activity for this process, defined as RTO, either caused by the business continuity plan or the business recovery plan. It will be assumed that the business continuity plan and the business

recovery plan are started simultaneously once the disruptive event has happened. The control $RTO \leq MTPD$ is one measure to evaluate the effectiveness of a BCM because a company will go out of business if it is not able to partially recover from an unforeseen and disruptive event in the time span given by the MTPD. As a second control $TTR \leq MAO$ can be used. It is a measure to evaluate that the IT and business processes are fully recovered in the time span given by the MAO. Any BCMS includes those business processes that are vital to the company.

From the concrete perspective of Credit Suisse, the bank's BCM can be looked at from a strategic and an operational point of view. In the first case, it coordinates according to the bank's global policy business continuity activities including information gathering, planning, implementation, testing as well as crisis management, to assure survivability of the bank in case of a major operational disruption, crisis and disaster. In addition to that IT disaster recovery differentiates itself from an availability management, by the severity of events and non-resolvability with ordinary management techniques and decision making authorities. In the second case, it consists of global and regional concepts and templates defining concrete tasks, a readiness assessment of the implemented BCMS and an established BCM reporting.

In detail, Credit Suisse's BCMS differentiates between a disaster, a crisis, a major incident and an incident. A disaster is an event that is primarily handled by the activation of the business recovery plan. A crisis is an event that requires critical decisions that cannot be resolved with ordinary management techniques. An incident is an event that may lead to a disruption of a critical business process or a low quality of service of a critical business process. A major incident is an aggregation of events that constitutes a group of incidents for which the consequences might be unknown.

The purpose of Credit Suisse's BCMS is to implement and maintain the organization's resilience to disruption, interruption or loss in supplying critical products and services in the event of a major operational disruption. The policy for BCM regulates roles and responsibilities as well as implementation and maintenance of planning, analysis, readiness assessment, communication and training aspects and regulates crisis management.

Credit Suisse's solution today is primarily an organizational solution. The current BCMS is based on a mixed plan and meta-plan-like concept that defines procedures of how to establish concrete plans in the case of a disruption of a critical business process. This plan and meta-plan-like concept is complemented by an organizational structure intended to handle emergency situations, characterized by check-lists and ad-hoc management procedures. The response time in this context is not always known and automatically generated decision alternatives for the given emergency situation are not available. Therefore, the current approach does not scale well. It is underspecified and cannot be smoothly split into orthogonal models. It focuses primarily certain types of disasters, not failures in business and IT processes. Therefore, combinations of failures caused by different disasters are not reflected as such. It lacks decision support and dynamic flexibility depending on the emergency situation. It does not have a sound concept of handling failures at different level of granularity and abstraction. 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.

# 4 A Loan Granting Process

A loan granting process is presented here as a critical business process a bank is running. In a first step, it is introduced from the general point of view of Credit Suisse. In a second step, a more concrete model is introduced that is discussed in the following section 6 more formally. At the end, this model is put in the context of a BCMS.

Credit Suisse (CS) runs its lending business actively. One business objective is to increase the client profitability. To realize this objective the full potential of an existing bank-client-relationship needs to be known to realize possible advantages for both sides. Therefore, a professional advisory service and individual financial solutions tailored to a client's personal situation are offered. Other business objectives are to strengthen the loyalty of existing clients and to acquire new clients.

Potential clients are natural or legal persons. For economic reasons a certain minimum business volume as well as a certain diversity of business activities between the bank and a client is expected.

For security reasons the customer relationship management process, the credit approval and administration process and the credit monitoring process are separated. Therefore, there are four distinct parties in a loan granting process to a client: the client relationship manager, the credit advisor, the credit officer and the credit unit. The client relationship manager is an expert in the concrete client relationship, the credit advisor is an expert in the bank's portfolio of credit products and services, the credit officer is charged with the responsibility of approving credit transactions, and finally the credit unit is handling exceptions in the credit business process as well as the ongoing administration, monitoring and reporting.

A loan is usually granted on a fully secured basis. It can be secured against marketable securities or against banking securities. Depending on the concrete case the percentage of coverage and the date up to which the securities need to be available can change. Because some securities are volatile an ongoing monitoring is needed.

The relationship manager (RM) is responsible for the entire client relationship to assure the highest possible client satisfaction, to develop a long-term business relationship and to optimize the profitability of this relationship. The RM will take care of collateral shortfalls, limit excesses and acount overdrafts.

The credit advisor (CA) is responsible for the lending business products. In particular, the CA is responsible for the profits generated by the lending products. His duty is to promote the lending business, to administrate specific requirements of clients, to assess the credit worthiness and to handle loan applications. The CA will – in cooperation with the RM – provide advice to the client. For example, he will point out the opportunities available regarding the various lending products and services. The CA advices the RM regarding specific conditions of products and services as well as their appropriate handling. He is responsible for the renewal of credit approvals.

The credit officer (CO) is responsible for credit approvals as well as limit accesses or account overdrafts that are beyond the defined tolerances.

The credit unit (CU) is responsible for the ongoing monitoring, reporting and control of running credit processes, their corresponding shortfalls and customer positions. The CU is assuring compliance with credit limits and repayment schedules as well as credit settlements at the maturity date. At CS, the credit monitoring is mainly based on reports automatically delivered by the IT systems in place. The CU can take actions in the event of emerging difficulties during a credit process.

From the point of view of this paper this loan granting process is simplified to match the scope of this study. It encompasses a client, a relationship manager, a credit advisor and a credit officer. 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. 3). 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. 4). 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 BCM. 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 an emergency process based on the calculated set of all continuity processes. It can reconfigure the running process instance in order to optimize time and cost functions while fulfilling security, risk and compliance side constraints.

# 5 Algebraic Graph Theory

In order to support the development of business recovery and business maintenance plans for different scenarios we propose to apply algebraic graph transformation [7], which is a formal, visual and intuitive technique for the rule based reconfiguration of object structures. Graph transformation offers analysis techniques for dependencies between transformation steps, which specify the actions of a business process in our scenario. This allows us to show how possible modifications of the process steps can be automatically computed.

From the formal point of view a graph grammar $G = (TG, R, SG)$ consists of a type graph $TG$, a set of transformation rules $R$ and a start graph $SG$. The type graph specifies the structure of possible object structures and the rules constructively define how models are modified. The start graph is typed over $TG$ and is the starting point for each transformation. Each 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. Graphs can be related by graph morphisms $m : G_1 \to G_2$, 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 of $G_1$ and $G_2$. Note that we can also use modeling features like attribution and node type inheritance as presented in [7].

The core of a graph transformation rule consists of a left-hand side $L$, an interface $K$, a right-hand side $R$, and two injective graph morphisms $K \xrightarrow{l} L$ and $K \xrightarrow{r} R$. Interface $K$ contains the graph objects which are not changed by the rule and hence occur both in $L$ and in $R$. Applying 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 the corresponding right-hand side $R$ of the rule. By $G \overset{p,m}{\Longrightarrow} H$ we denote the graph transformation step where rule $p$ is applied to $G$ with match $m$ leading to the result $H$. The formal construction of a transformation step is a double-pushout (DPO), which is shown in the diagram above with pushouts $(PO1)$ and $(PO2)$ in the category of graphs. $D$ is the intermediate graph after removing $m(L)$ and $H$ is constructed as gluing of $D$ and $R$ along $K$.

$$\begin{array}{ccccc} L & \xleftarrow{l} & K & \xrightarrow{r} & R \\ {\scriptstyle m}\downarrow & (PO_1) & \downarrow & (PO_2) & \downarrow{\scriptstyle m^*} \\ G & \longleftarrow & D & \longrightarrow & H \end{array}$$

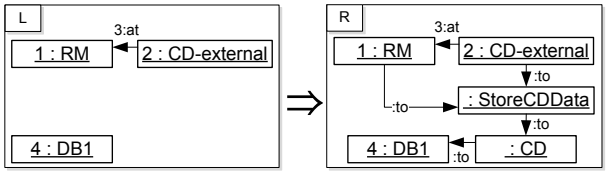


Figure 1: Rule *storeD*

Fig. 1 shows the rule *storeD*, which specifies the function "Store CD" of the EPC for the loan granting process in Sec. 6. Since the rule *storeD* is nondeleting we have that in this case $K = L$. This will also be the case for all derived rules in our scenario. The effect of the rule 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 morphisms $l : K \rightarrow L$ and $r : K \rightarrow R$ of the rule are denoted by numbers for nodes and edges, i.e. each number specifies a mapping between two elements.

# 6 Analysis and Optimization

The critical business process about granting a loan is introduced next from the point of view of a workflow engine using a slightly modified version of an event driven process chain (EPC) [8].

This type of EPC consists of business functions, events, organizational entities and applications executing business functions, as well as the dataflow into and out of a business function from or to certain data storage units. Such storage units are either databases, or, alternatively, organizational entities, like concrete persons. Because we take the perspective of a workflow engine, this scheme will run as an instance when performing a concrete business process. As such, a workflow instance owns local storage to remember certain data values. Data values that are kept in local storage of the workflow instance are modeled by dark-blue boxes. Data values that are not kept in local storage are put into light-blue boxes. This allows to cover automated and non-automated parts of a business process by the help of one single process model. In case that a data value is still known to the workflow instance, it has not to be re-loaded from a database or entered by a person. In case that a failure occurs, data values can be re-loaded on demand. We like to name this a data-flow oriented EPC from the point of view of a workflow engine, or in short: WDEPC. A central advantage out of this understanding is that business functions can be shifted back and forth in the process chain depending on the data values they require. So, in contrast to today's understanding of EPCs, data flow dependencies can be handfeed orthogonal to event chain dependencies.

Further, the notion of local storage of a workflow instance enables work-arounds of temporarily non-available data storage units. A business function is performed by a person, an application or any combination out of these. Data values not relevant to the workflow are abstracted.

## 6.1 Example Scenario

The presented business process – as introduced in section 4 – has to meet certain objectives. There are primary and secondary goals. From the point of view of CS, this process needs to generate a contract and leads to a long-term client-bank-relationship. Relevant attributes regarding the contract are the time required and the realized costs, regarding the relationship the relevant attribute is the measurable client loyalty. It further requires to respect certain security, risk and compliance requirements.

In the given case, security will be proven by the help of a graph constraint check assuring the four eye principle when signing the loan contract. Risk will be simulated from a perspective of the workflow scheme discussing possible failures and recovery strategies in the limits of a 48h MTPD

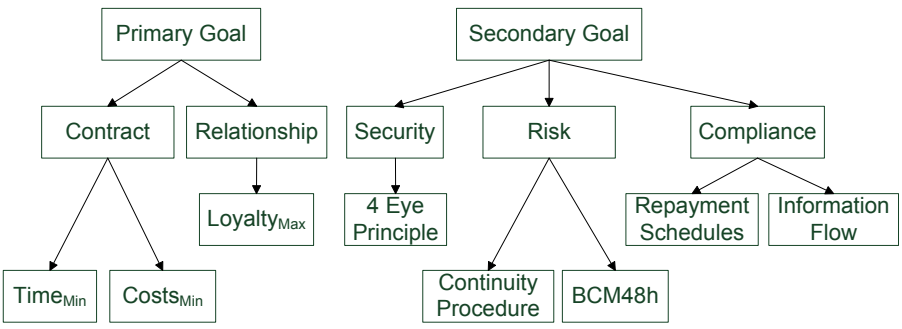


Figure 2: Primary and Secondary Objectives

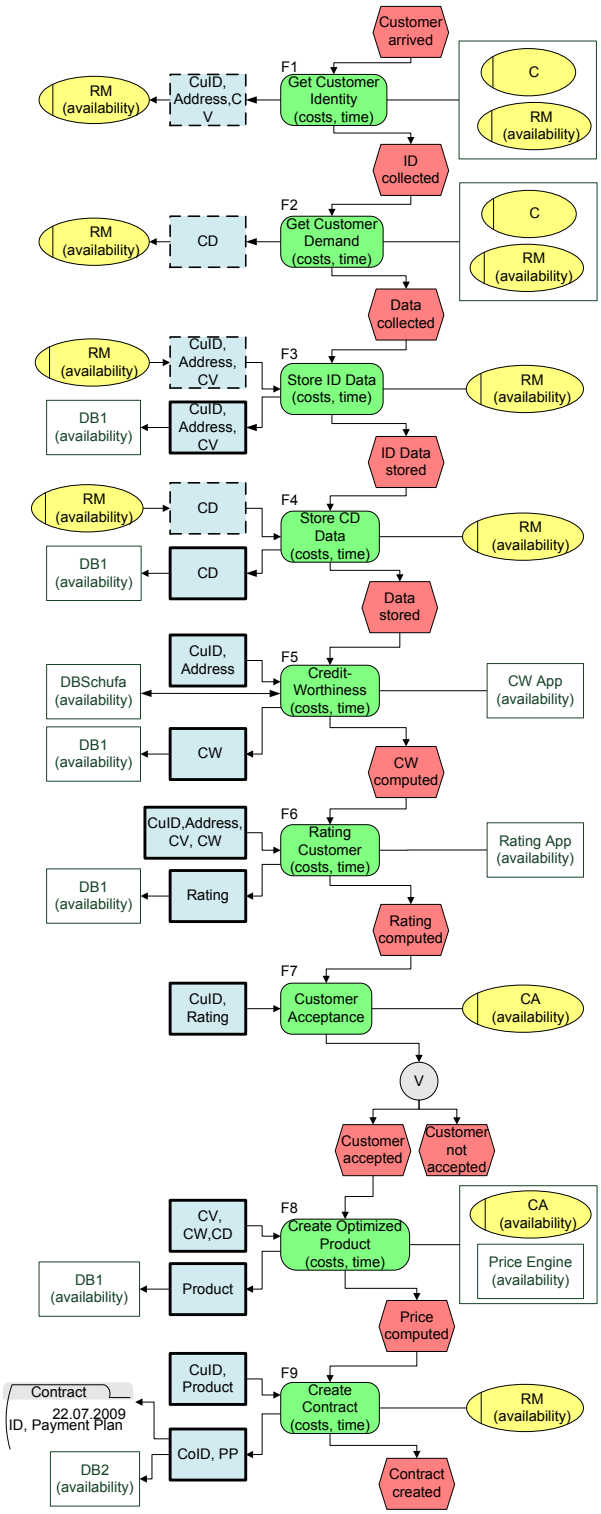


Figure 3: Workflow Part 1 of *LG*

BCM baseline ($RTO \leq MTPD$). Finally, compliance is assured by testing the workflow instance against the real-world situation.

Figures 3 and 4 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.

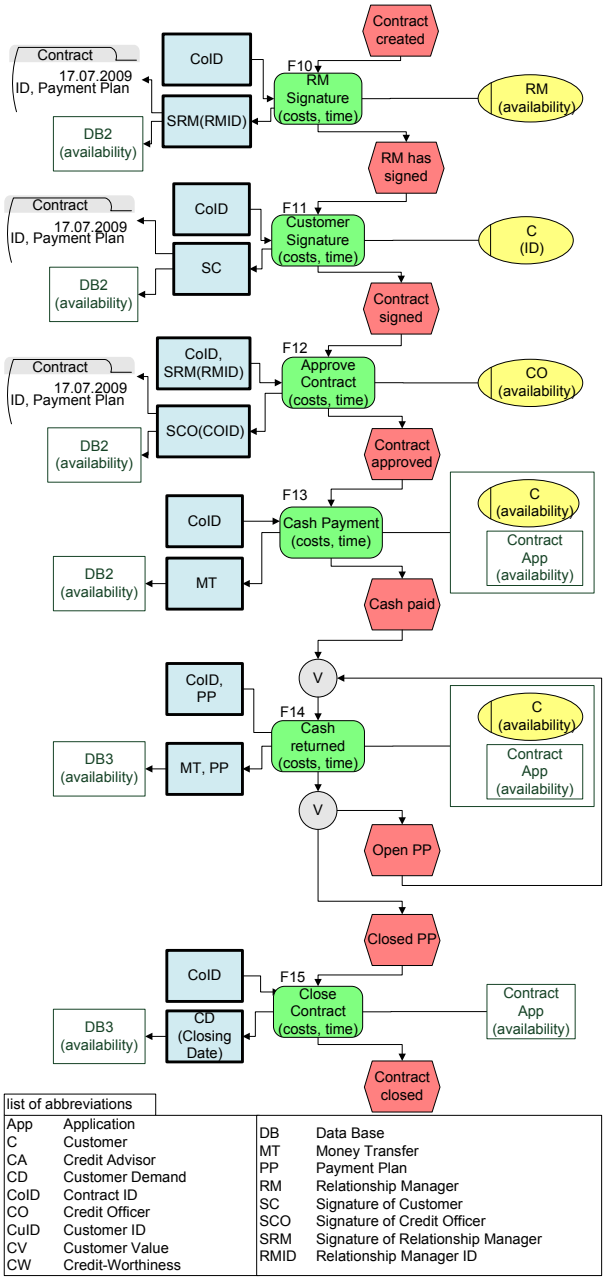


Figure 4: Workflow Part 2 of *LG*

## 6.2 Graph Grammar for a WDEPC

Business process models given by EPCs often consist of chains of functions. The intermediate events imply virtual dependencies of consecutive functions, even if these dependencies do not exist in the real process. 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. Furthermore, if a WDEPC is not executable because data elements are needed by functions but these data elements were not created before, we can detect this inconsistency by checking whether the sequence of graph rules according to the WDEPC leads to a graph transformation in the derived grammar.

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. 3 and its corresponding rule in Fig. 1). 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.

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 [9, 10].

The following figures show the rules of the graph grammar $GG_{LG}$ for the critical business process modeled by the WDEPC $LG$ as shown before. Each rule corresponds to an equally named function in $LG$.

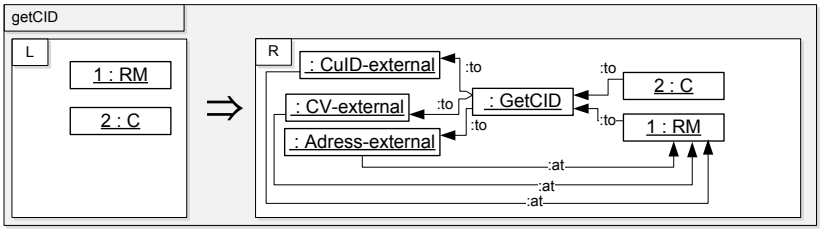


Figure 5: Rules of graph grammar $GG_{LG}$ - Part 1

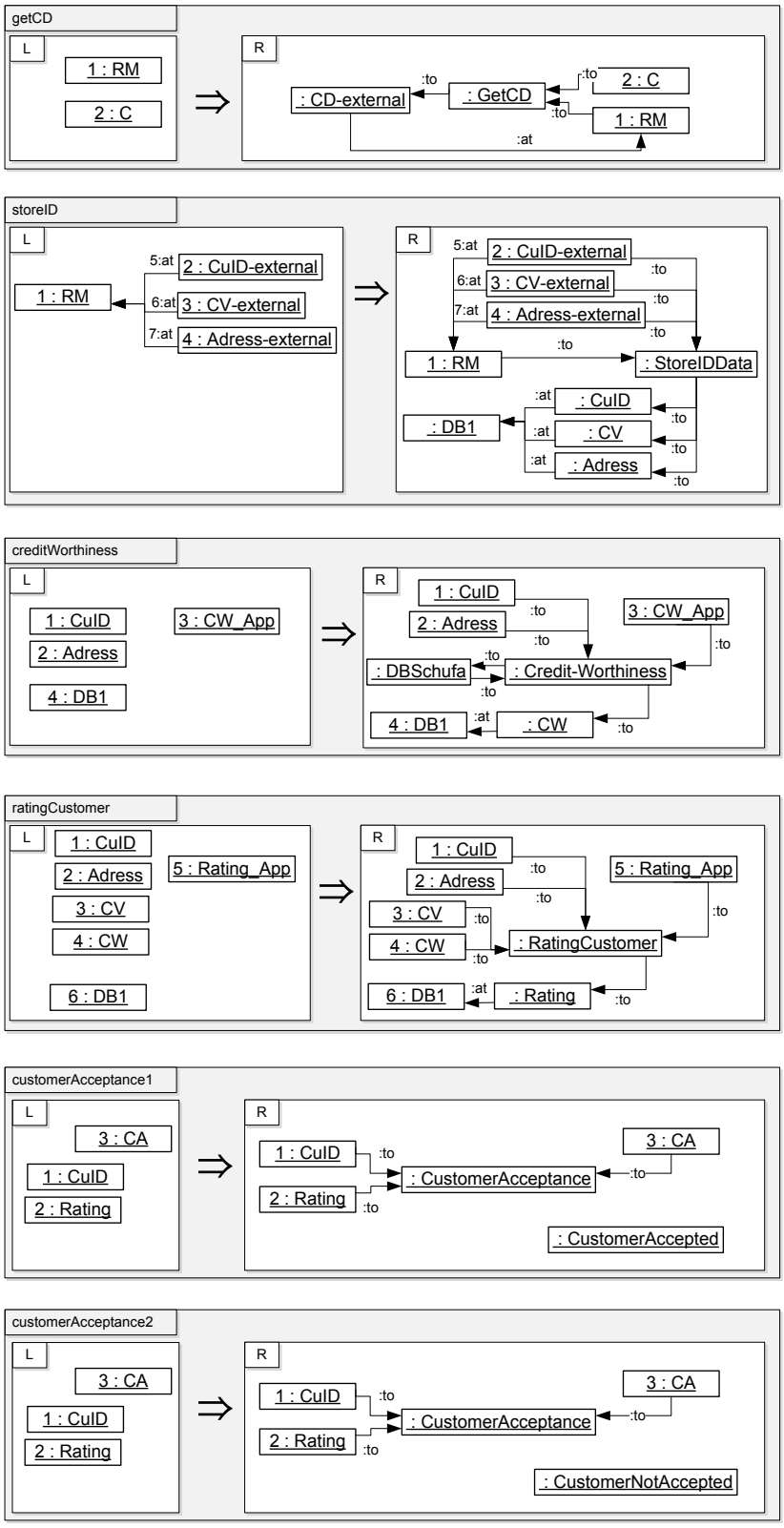


Figure 6: Rules of graph grammar $GG_{LG}$ - Part 2

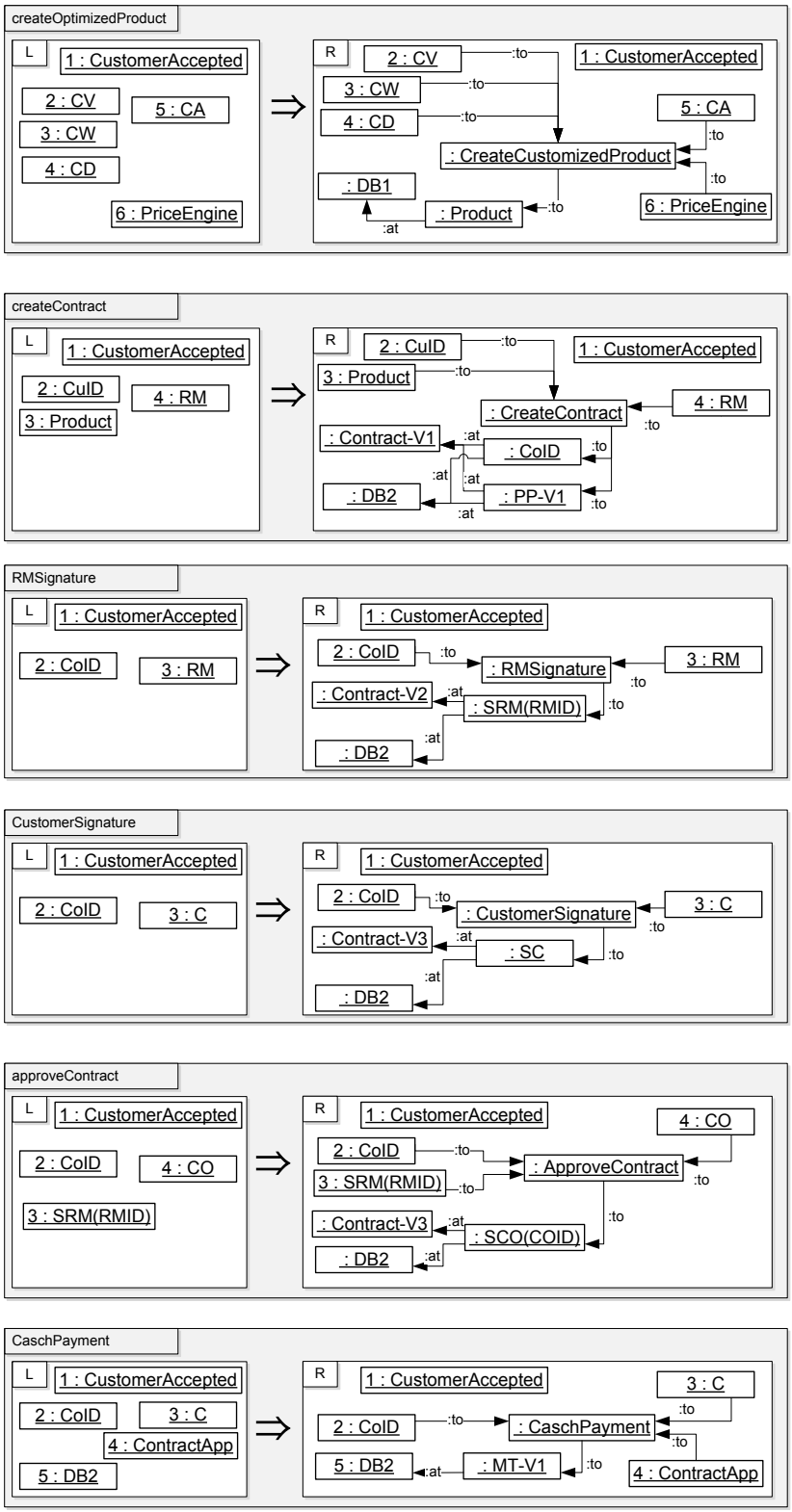


Figure 7: Rules of graph grammar $GG_{LG}$ - Part 3

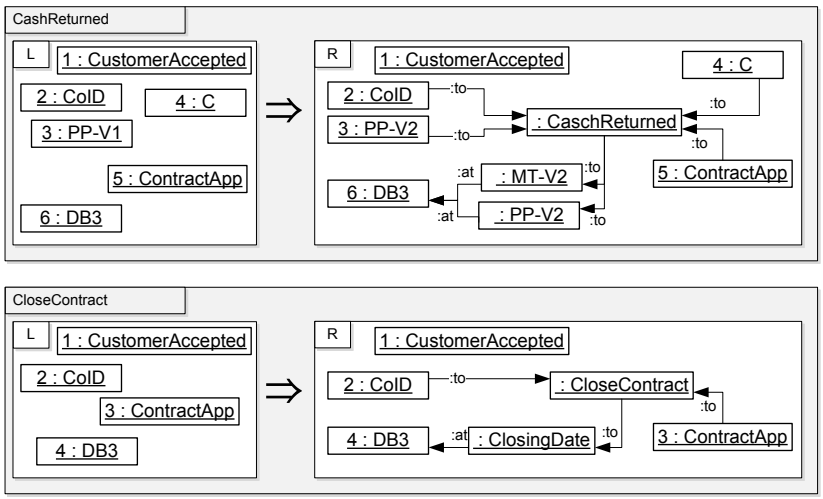


Figure 8: Rules of graph grammar $GG_{LG}$ - Part 4

## 6.3 Computation of Dependencies

In the following we explain the analysis of dependencies for the process $LG$, which is described in Sec. 6.1. Consider the first four functions "Get Customer ID", "Get Customer Demand", "Store ID Data" and "Store CD Data". The corresponding rule of "Store CD Data" is shown in Fig. 1 and we have for the corresponding rules $p_1 = getID$, $p_2 = getD$, $p_3 = storeID$ and $p_4 = storeD$ in $GG_{LG}$ only the following dependencies: $p_1 <_{rc} p_3$ and $p_2 <_{rc} p_4$, where "rc" 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 and thereafter, the necessary identification information is collected and stored. 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.

## 6.4 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:

**Composition of Process Parts** Consider that we have process parts $P1$ and $P2$. They are composable, if first of all the start event of $P2$ occurs in $P1$ - this is the gluing point of the composition and we denote the new part by $Q = P1; P2$. Furthermore - in order to have that

$P1; P2$ can be executed - each left hand side of a rule $p_x$ of the corresponding grammar $GG_Q$ has to be included in the start graph joined with the right hand sides of the rules that correspond to preceding steps. This condition is sufficient, because the constructed rules are non-deleting. If $P1$ is already executable then the check can be reduced to the rules of $GG_{P2}$.

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. 9 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.

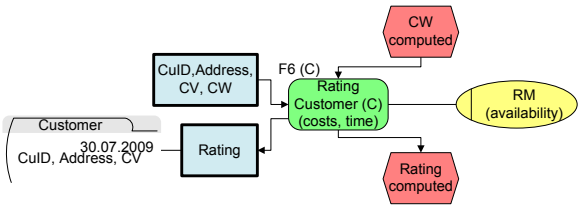


Figure 9: Alternative Function "Rating Customer (E)"

"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 $p_1 = getCID, p_5 = creditWorthiness, p_6 = ratingCustomerC$ as well. We have the following pairs of the relation "read causality": $p_1 <_{rc} p6$) and $p_5 <_{rc} p6$), i.e. $p6$ needs some elements that are produced by $p_1$ and $p_5$.

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:

1. created by "Rating Customer (C)" as well or
2. 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. 9 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. 10 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.

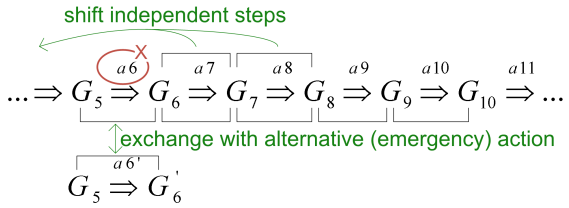


Figure 10: Automatic generation of alternatives

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. 11, 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.

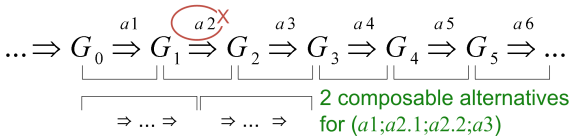


Figure 11: Complex alternatives

We now present the further emergency fragments for the process $LG$, such that alternative continuity processes can be generated automatically for combinations of failures. For some business functions there are two emergency fragments, depending on the combination of failures that may occur.

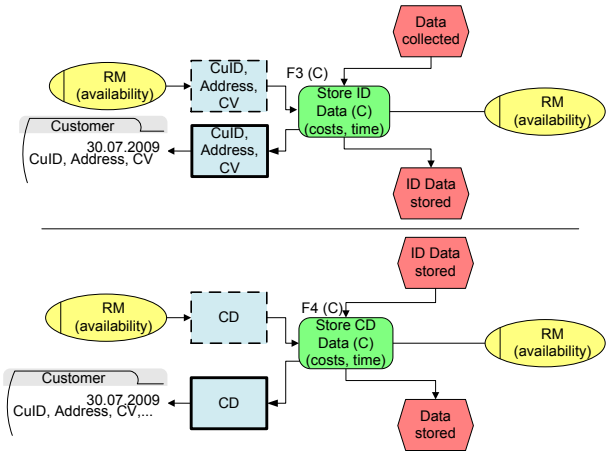


Figure 12: Emergency Fragments - Part 1

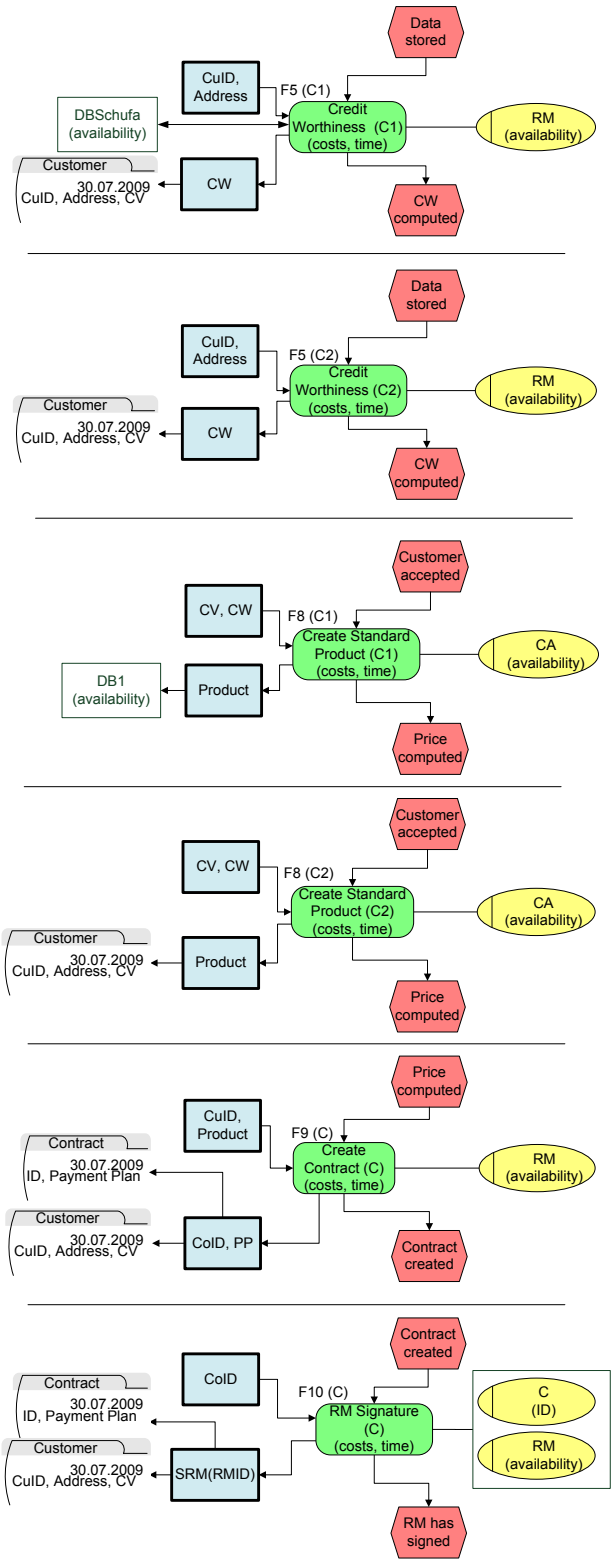


Figure 13: Emergency Fragments - Part 2

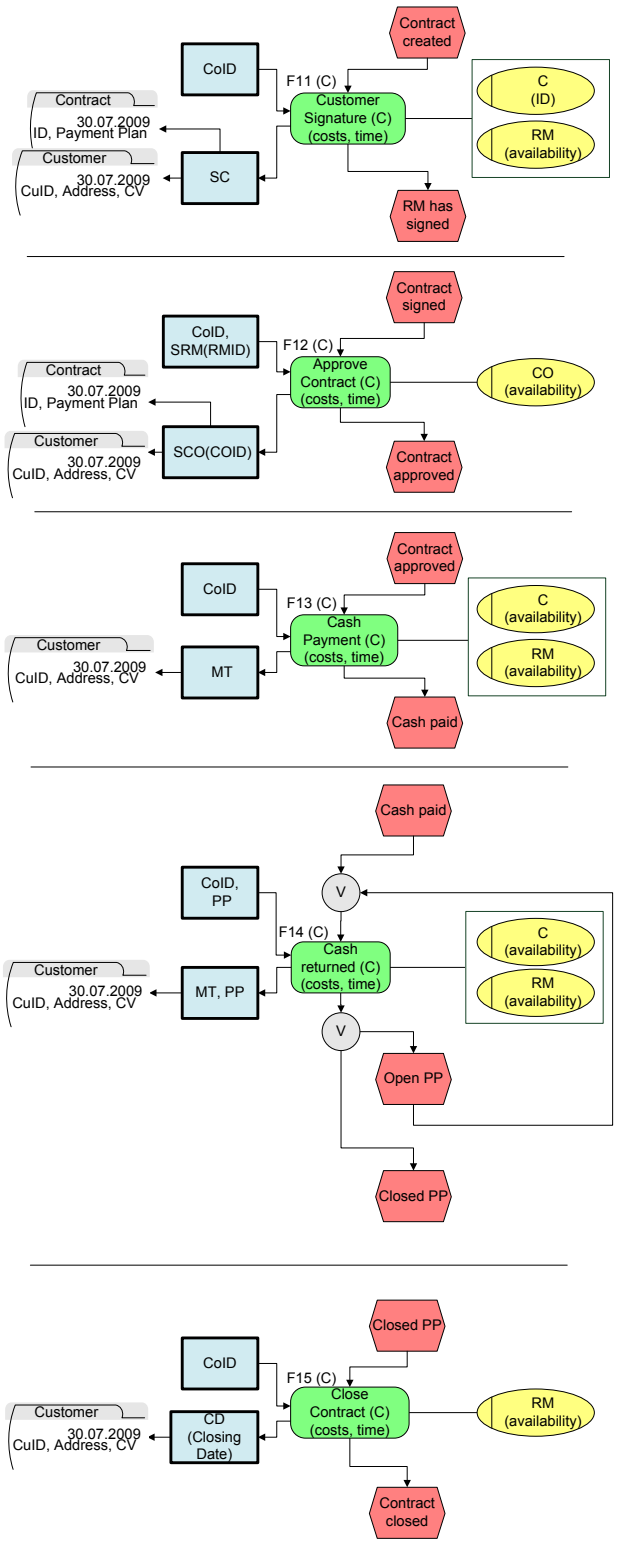


Figure 14: Emergency Fragments - Part 3

## 6.5 Validation of Objectives

In order to validate that the non-functional objectives are fulfilled by the generated alternative processes the requirements for security and compliance 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. Now, 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".

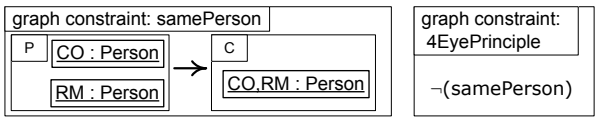


Figure 15: Graph Constraint: 4 Eye Principle

Fig. 15 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. 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 [7]. 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 $p : P \to G$) we have that there is also a compatible occurrence of $C$ (given by an injective morphism $q : C \to G$, such that $p = q \circ c$ for the constraint morphism $c : P \to C$). 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 constraints 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 [7, 10].

Summing up, once a business process is modeled its corresponding graph grammar can be derived automatically, and graph constraint checks can be performed on the abstract syntax of such a model 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.

After having validated these objectives, we are able to make a statement about the effectiveness and efficiency of a business continuity management system, as well as if it is economically sound in respect to security, risk and compliance requirements.

# 7 Related Work

TODO: integrate the new parts

In [11] 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 [12] 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 [13] an organizational solution to address information security management problems is presented. But this solution cannot be automated as requested by CS. In [14] and [15] 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 [16] 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 [17] 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 [18] 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 [19] change patterns are proposed as a means to handle modifications of a workflow model and in [20] important correctness problems regarding general modifications are discussed in a comparative survey. In the present scenario already well-formed sub-processes 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 [21] and [22] 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.

# 8 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. The business functions of the derived process models are ensured to get correct and available in- and output data and furthermore, the organizational entities are ensured to be able to retrieve the data, because they are required to have access to them. Finally, the graph model specifies which actor executes which business function and which data occurs on which storage devices. This enables automatic checks of security requirements using graph constraints as

well as simulations that can be evaluated usind the annotated costs.

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.

# References


[1] Knight, Pretty: The impact of catastrophes on shareholder value. In: The oxford executive research briefings, University of Oxford, Oxford, England, Templeton College (1996)

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

[3] Brandt, C., Engel, T., Hermann, F.: 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, Berlin/Heidelberg, Springer (2009) 339–352

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

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

[6] Boehmer, W.: Survivability and business continuity management system according to bs 25999. In: Proc. Int. Conf. on Emerging Security Information, Systems and Technologies (SECURWARE 2009), Athens/Vouliagmeni, Greece, IEEE Computer Society (June 2009)

[7] Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. EATCS Monographs in Theoretical Computer Science. Springer (2006)

[8] Scheer, A.W.: ARIS-Modellierungs-Methoden, Metamodelle, Anwendungen. Springer, Berlin/Heidelberg (2001)

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

[10] Hermann, F.: Permutation Equivalence of DPO Derivations with Negative Application Conditions based on Subobject Transformation Systems. In: Proc. Int. Conf. on Graph Transformation-Doctorial Symposium (ICGT-DS'08), Electronic Communications of the EASST (2009) (to appear).

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

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

[13] Eloff, J.H.P., Eloff, M.: 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), Republic of South Africa, South African Institute for Computer Scientists and Information Technologists (2003) 130–136

[14] Quirchmayr, G.: 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), Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2004) 3–6

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

[16] Asnar, Y., Giorgini, P.: Analyzing business continuity through a multi-layers model. In: Business Process Management, Lecture Notes in Computer Science. Volume 5240., Berlin/Heidelberg, Springer (2008) 212–227

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

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

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

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

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

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