# BUSSINESS PROCESSES MODELING SOLUTION FOR CROSS-ORGANIZATIONAL COLLABORATION

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea Politehnica Timişoara
în domeniul Calculatoare si Tehnologia Informatiei
de către

**Ing. Maria Laura Șebu**

Conducător ştiinţific: prof.univ.dr.ing. Horia Ciocârlie
Referenţi ştiinţifici: prof.univ.dr. Dana Petcu
prof.univ.dr.ing. Dorian Gorgan
prof.univ.dr.ing. Vladimir-Ioan Creţu

Ziua susţinerii tezei: 2 Noiembrie 2016.

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Ingineria Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică şi Telecomunicaţii
8. Inginerie Industrială
9. Inginerie Mecanică
10. Ştiinţa Calculatoarelor
11. Ştiinţa şi Ingineria Materialelor
12. Ingineria sistemelor
13. Inginerie energetică
14. Calculatoare şi tehnologia informaţiei
15. Ingineria materialelor
16. Inginerie şi Management

Universitatea Politehnica Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul Şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.



România, 300159 Timişoara, Bd. Republicii 9,
Tel./fax 0256 403823
e-mail: editura@edipol.upt.ro

# Acknowledgements

First and foremost, I would like to express my gratitude and my appreciation to my principle supervisor **Prof. Dr. Eng. Horia Ciocârlie** for accepting my research proposal, for believing in me, in my ideas and in my work, helping me with precise advice and continuous support all the way through, for patience and professionalism in supervising and guiding me in the right direction. His attitude influenced me not only in this PhD research but in my entire carrier.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Dr. Eng Ștefan Holban, Prof. Dr. Eng. Ionel Jian and Assoc. Prof. Dr. Eng. Dan Pescaru for their comments and observations, for all the questions and suggestions which made me extend my research to consider new dimensions.

Special thanks go to Prof. Dr. Eng. Horațiu Moldovan for challenging me in open-minded discussions about organizational management, for supporting me in switching my carrier from technical to project management, professional step which created the context for the current research.

I would also like to thank Politehnica University of Timisoara for the financial and academic support of my attendance to various conferences allowing me to gain visibility for my work and to create connections for possible future collaborations.

My sincere thanks go to my current employer Infobest for understanding my passion to materialize my ideas, for the conditions created, flexibility in my work schedule.

Appreciation and thanks go to all the research fellows, PhD colleagues and all the great professionals I met during the conferences and events, who influenced and helped me with advice and ideas.

Special thanks go to all my wonderful friends who believed in my work and surrounded me with confidence and care when I needed the most.

Last but not the least I would like to thank my mother, my biggest fan, for her continuous and unconditional love and support in all my goals, for never leaving my side through tough times, and to my father who is watching over me from up above and hopefully, is feeling very proud of all my achievements.

Timişoara, Septembrie, 2016 Ing. Maria Laura Șebu

Șebu, Maria Laura

**Business Processes Modeling Solution for Cross-Organizational Collaboration**

Teze de doctorat ale UPT, Seria X, Nr. YY, Editura Politehnica, 200Z, 122 pagini, 53 figuri, 13 tabele.

ISSN:

ISBN:

Cuvinte cheie: business process management; process mining; collaborative environment

Rezumat,
The chosen research addresses the problem of multi organizational collaboration in a dynamic context in which organizations share common business goals and the time available for the setup is limited. The thesis proposes a new direction in the area of collaborative BPM and offers technical methods for supporting the collaboration. The current research considers new directions in each of the components included: context dependent business process similarity, merge of business processes, functional validation of the collaborative process model, rule based monitoring of business process execution, business process improvement with input extracted from usability patterns.

**CUPRINS**

**Lista de Figuri**

**Lista de tabele**

# 1. INTRODUCTION

## 1.1 Organizations

Business is organized nowadays on the concept of organizations. An organization is a formal social structure using resources, identified through capital and labor as input factors and products and services as outputs. An organization is a legal and formal entity with internal rules, procedures and social structures as represented in Fig. 1.1. Surely organizations differ radically on the market but in general any entity which follows common structures, methods and processes, which has a common goal and strategy and acts in a legal context, could be considered an organization.

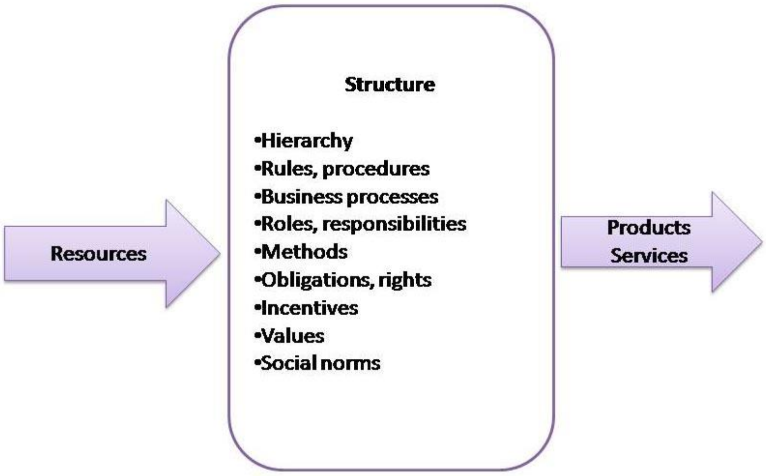


Fig. 1.1 Behavioral view of an organization

Information Systems (IS) nowadays are the basis for the development of business inside organizations. Almost every sector benefits of this trend followed in all enterprises. IS use business processes for the representation of the flow of work executed inside an organization. Business processes are managed by Information Systems to execute business cases for achieving business objectives. Enterprise Information Systems (EIS) are designed, implemented and executed to support an organization. Workflow Management Systems (WMS) and Enterprise Resource Planning Systems (ERP) are two possible EIS which offer capabilities oriented on processes and are executed in many organizations.

Business is organized considering the notion of competitive advantage. A competitive advantage is a concept describing the actions performed by organizations to achieve better results than competitors. Such actions include acquisition of skilled resources, geographic location, trainings, customer analysis, and economical factors. An IS system could be called strategical if it implements a strategical direction inside an organization to ensure the competitive advantage. Possible strategic directions are to deliver products or services of better quality at lower costs, deliver unique products or services, focusing on specific market segments. Sustainable **competitive advantage** over time is a difficult achievement for an organization.

One of the possible strategic solutions for a sustainable competitive advantage is the initiation of collaborations between organizations.

Besides the competitive advantage as a direction, organizations operate nowadays in an **agile business environment**. A characteristic of the current market environment is the high level of dynamism. Performance in business is a mandatory condition for keeping customers satisfied and a continuous development of the business. However, the development of technologies and information is a major challenge for organizations which plan to offer services and products of the highest quality. As such challenges exceed the scope of single organizations, a possible strategy to respond to the high level of dynamism is the setup of inter organizational collaborations.

Many other business strategies nowadays have as possible directions for resolution the setup of inter organizational collaborations. Collaboration is initiated by organizations for achieving common pre-defined objectives in a mutual developed structure. Collaboration could be defined as a formal relationship with clearly defined roles and responsibilities for each of the contributor. A collaboration set up could be more or less formal, different aspects of inter-organizational collaboration are analyzed further in the next chapters.

Because of the wide variety of organizations in several business contexts, the complexity of setting up and supporting collaboration could be easily identified. Any project realized in collaboration instead of internally inside a single organization is considerably more complex because of the many factors which contribute to the success of collaboration. We limit the current research to the operational perspective of collaboration and we target proposing technical methods to ensure collaboration could be easily setup between two or more organizations and to monitor it for ensuring it is on the right track. An extension to include a social dynamics perspective when a joint structure is constructed will be considered as a follow up of the current research.

Collaborations as a clear business direction created the perspective for Collaborative Information Systems (CIS) able to integrate and coordinate information from different sources. CIS are the base for implementing collaborative environments. An effective collaboration is constructed by all the members working together for achieving a common goal, a successful project. Such a cooperative environment constructed on a CIS allows a horizontal integration of the resources for a partnership initiated to bring value to all participant organizations.

*In the current thesis we propose a solution for supporting a CIS and achieving a horizontal integration of processes and resources contributing to business implementation.*

A side effect of an inter-organizational collaboration is the high level of complexity due to the needs of integration and coordination of business processes. We considered this difficulty as a challenge and in the current thesis we offer technical support for setting up collaboration with shared business objectives between organizations.

When different organizations decide to collaborate in their business, a need to integrate their processes for a common IS solution is mandatory, followed by support in methods and technologies used to manage such solutions. In Fig. 1.2

such a collaboration started by organizations and the common organizational process model followed is represented.

In this research we propose technical solutions for constructing automatically a collaborative business process model able support a multi organizational collaboration with clearly pre-defined targets. Once the collaborative model is composed, we offer support for verifying and validating it with the purpose of ensuring it can achieve the business objectives pre-defined. Once the execution is scheduled, we offer technical support to monitor the execution and to detect deviations in time so that response actions could be set up for the prevention and the correction of undesired results. For the composed collaborative solution we offer technical means to support the evolution of the process model, for the good performance of the organization and achieving better business results. Processes might be difficult to follow in specific areas and should not be enforced but instead, business process models should be adaptive and be able to evolve in an agile dynamic context. The current business process based solution able to support technically the collaboration between organizations is optimized by using the monitoring information extracted during the continuous execution.

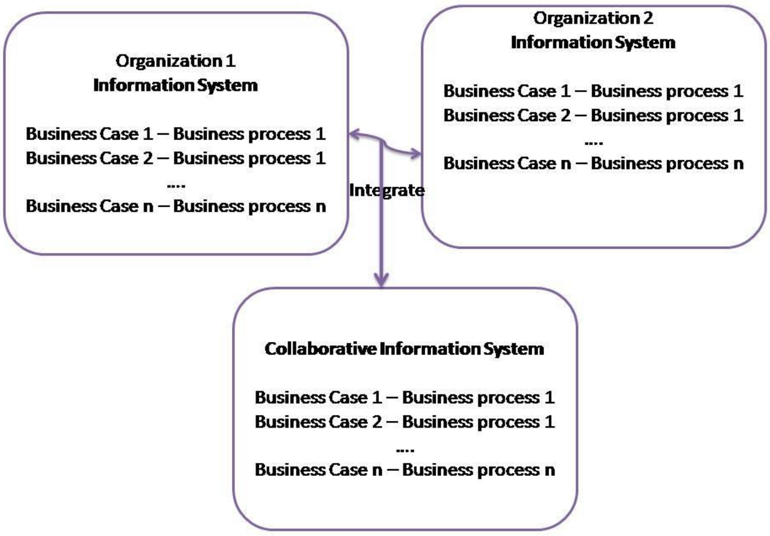


Fig. 2.1 Collaborative information systems

IS applied in different business contexts have different languages and different levels of abstractions. Information heterogeneity leads to another objective identified for the current research, semantic interoperability; achieving a mutual understanding of the data used inside organizations as such data will be the base for the collaborative business process based solution. The solution proposed is able to support inter organizational collaboration between organization belonging to different business environments because we added capabilities for achieving semantic interoperability.

When an inter-organization collaboration could be set without a high effort, the participant organizations are considered compatible. Managing communication, defining working modes and business processes, synchronizing objectives are more facile when organizations have a good compatibility from business process, operational perspective. In the current thesis we propose means to compute the similarity between organizations from process perspectives. The computed value is able to quantify how similar organizations organize their internal processes and logistics for achieving business objectives. The collaborative business process based solution is started out of a high similarity between business processes of the participating organizations.

# 2. OVERVIEW

## 2.1. Market Analysis

Companies involved in strategic alliances and collaborations have better results on the market: return of investment, business results, higher return on equity, than companies involved in merges or acquisitions as directions for developing organizations [1]. This strategy and the possibilities for achieving successful results have opened up the perspective for the research captured in this thesis. Alliances in the form of collaborations are easier achievable than mergers or acquisitions.

An acquisition is the process of taking over, purchasing a company. A merger is the process of agreement between two companies to continue further the business proposed and operate as a single new entity. By merging or by acquiring a new organization, the companies hope to benefit from staff reduction, economies, new technologies, market, and visibility. Due to the high level of dynamism of the market in which projects cover many business areas, mergers or acquisitions are not suitable to respond to this direction. Alliances and collaborations are more convenient for accomplishing projects which exceed the scope of single organizations, for long term commitments with a full reconsideration of the strategy, organizational hierarchy and model, market audience. Alliances need a more detailed procedure for support and a clear plan for setup and control.

In the current thesis we don't target to offer solutions for supporting an acquisition between two organizations. But instead to respond to collaborations initiated on specific business cases in a dynamic environment on short or midterm basis. Such collaborations initiated need an easy setup, a clear working model which could be put in execution as quickly as possible without considerable effort. In the current thesis we propose an automated solution able to check if a collaboration dynamically initiated is feasible from operational perspective and to support the execution and achieve business objectives.

As detailed described in [1], there is a high variety of inter-organizational relations. The most important ones are listed:

- **Hierarchical Relation**: Put in place by a full takeover of one company by another though a merger or an acquisition
- **Joint Venture**: Companies create a jointly owned organization serving limited purposed for participant organizations
- **Strategic Agreements**: collaborations over key strategic decisions and shared responsibilities for outcomes
- **Cooperatives**: small enterprises that manage jointly their resources

The base for the current research is the collaborations between organizations on isolated business cases. The reason is the high adaptability needed for success and the short time available for setup. If mergers and acquisitions are performed after a business agreement, and time and resources are ensured for the setup, for isolated projects realized in collaborations difficulties arise due to the need for adaptation in short time.

Companies enter in strategic collaborations for several reasons [1]:

- market opportunities
- access to new technologies and converging technologies
- need for collective skills
- diversify into new business

- improving performance
- access to resources
- cost sharing
- risk reduction
- competitive advantages
- specialization

The motives for entering in alliances could be enacted at three levels [1]: organizational

- economical
- strategic
- political

Choosing the right partner for the collaboration is critical independent of the reason for entering collaboration. Evaluation must be simple and straightforward and the result must be easily interpretable. We propose in the current thesis such an evaluation of the possible business partners for identifying the correct one by considering the business context.

In [2] four levels of collaborative maturity models are identified.

- **Interconnecting**: communication is ensured by asynchronous messages, communication interfaces are available in both collaborating partners
- **Open**: collaboration is based on sharing features and services, communication is synchronous
- **Federated**: ability to work in a collective environment composed of processes use to integrate and to compose a collaborative process
- **Interoperable**: highest level in which integration is achieved

The collaborative solution presented in this thesis is designed for a federated collaborative environment. The collaboration is business process based and solutions to support lower collaborative forms are proposed and put in practice. When business processes are not formalized inside organizations, process mining techniques are used to extract business processes and to attach them to business cases. Directions for achieving interoperability are also considered, by proposing business process improvements techniques able to ensure the evolution of the collaborative solution. Another direction for achieving interoperability as target is by using the ontologies which allows collaborations between organizations of different business areas.

Semantic interoperability solves conflicts determined by the differences in the terminology of representing the business process data. In [2] two possible directions to achieve semantic interoperability are presented. Standardization is one direction in which a common terminology is used for data representation. This allows the exchange of data according to a standardization model. The collaborative solution presented in the current thesis is constructed considering the availability of standardization in business process representation and organizations collaborating in the same domain area. For this purpose a syntactic approach is proposed for identifying the correct business partner organization and constructing a collaborative process model supporting the business case realized in collaboration. The syntactic approach is fast and low resource allocation.

If standardization cannot be achieved, semantic interoperability is ensured through the usage of ontologies for data representation by merging and mapping of ontologies. Mapping of ontologies coresponding to input business processes was introduced to permit the interrogation of the knowledge base by using a common interface. Merging the ontologies of the input business processes was considered to create a new ontology able to represent the information of the original ontologies for all participating business processes. Ontological dimension was considered to limit the semantic space to the vocabularies of the input business process models. The full semantic space is also presented as an alternative to the lack of standardization in business process representation with the associated limitations: high resource consumption.

By ensuring the collaborative solution proposed in this thesis is feasible to support companies with low collaborative potential maturity and by achieving interoperability, even if the solution is designed for companies with federated maturity profiles, we offer and propose techniques to support collaborations between organizations independent of the maturity level. If organizations don't have a Business Process Repository constructed formally and used as a reference, discovery algorithms are use for identifying the business processes used inside organizations. If organizations use for representation of process models different granularity, we propose techniques to cope with this disadvantage.

## 2.2. Context Analysis

The benefits of succesful collaboration are undoubtable. Market opportunities, in time response to business challenges, access to resources, access to knowledge, new opportunities. However collaborations are usually started by a top management decision without defining the means to achieve business objectives. Unrealistic targets and lack of means to engage the collaboration partner cause disruptions in the execution phase and deviations from business targets identified. One of the main reasons for failures in the area of collaborative business is the incorrectly set objectives. As outsourcing is started by a strategical board decision, one of the most common reasons is reducing costs. Due to this direction instead of choosing the most suitable partner, the cheapest one becomes the straightforward choice. We propose in the collaborative business process based solution described in this thesis identifying the most compatible collaboration partner. The ability to make the right choice from the beginning opens up the perspectives for succesful collaboration from the start. The compatibility factor could be considered in a context and it should be correlated with the business circumstances: costs, political and economical factors, cultural impact.

Any collaboration between two or more organizations includes changes in the workplace culture. Such changes jeopardize the stability and endanger the accomplishment of the business targets set. If similar working modes are used in both organizations participating to the collaboration, changes in the workplace culture are limited and collaboration could be finalized with objectives met. The compatibility factor is able to evaluate from process perspectives the similarity between organizations. The collaborative business process based solution is composed of input from all participating organizations. The impact of performing a project in collaboration instead of in-house is reduced with the technique proposed in the current research.

Another reason identified for failures in multi organizations collaborations could be located in the area of expectations set. Unsynchronized objectives and undefined targets could lead to delays, cost overruns, quality problems. The solution we propose for supporting the inter-organizational collaboration is based on business targets defined and modeled as input data sets. The resulted process model is validated with the test data. The proposed technique for validation ensures the synchronization with business objectives pre-defined.

One aspect which could impact collaboration between two or more organizations is the problems in communication which could arise from cultural differences, language misunderstandings, limitations in understanding industry terms. Such communication issues could be improved with a clear business process model able to support the shared work.

The lack of a clear model for executing the project in collaboration is another cause for failure of collaborations. Usually at top management level a Service Level Agreement (SLA) is defined and mechanisms for arbitrating problems: escalation, change management are put in place. But a clear working model flexible and collaborative is rarely identified. Lower hierarchical levels in organizations have as a target performing the collaborative work and accomplishing business objectives. Due to the lack of information and support good results are delayed and time is spent in the setup of the collaboration instead of performing the work for achieving objectives. The proposed solution in the current thesis has as a major objective overcoming this common scenario on the market.

Surely many other failure reasons could be identified. Working as a Project Manager in outsourcing for many years challenged me in finding solutions for supporting projects in collaborations with different customers coming from different business areas. Collaborative solutions found are more theoretical and need a lot of input data and effort for setup. Current solution uses process log data already available inside participant organizations; it computes the compatibility level between organizations and ensures a correct partner selection from the start. Also a business process model is proposed as the base for the collaboration which could be enhanced with custom activities if case. The methodology is synchronized with the business objectives defined and modeled for validation. Operational support for execution is offered and the field for further process improvements is prepared. The system is able to evolve and adapt influenced by external factors: resources executing activities.

The collaborative solution business process based proposed and the set of tools is attempting to become an alternative to management trends put in practice nowadays in companies with the scope of aligning business strategic objectives with operational: Six Sigma, Lean. Such business directions are considered great directions to follow but difficult targets for companies which already have a clear operational setup. Most of the techniques for process improvements are based on changing the culture of an organization, a very challenging plan to start and achieve. We are proposing a set of techniques and tools which would help organizations to improve their business and reach new markets by setting up collaborations based on business process. Our solution is constructed on the assets of each organization and attempts to offer a solution that avoids a major change in the way each organization performs individually. But instead, to gradually integrate the common processes followed in business cases which must be realized in collaborations.

## 2.3. Technical Analysis

Collaborative solutions existent on the market are either theoretical without technical support to achieve business results on short term, or are based on a complete change of the culture for an organization. We will detail in the next chapter the alternatives and the drawbacks for each of the existent solutions on the market.

Besides the lacks in the area of collaborative solutions able to identify a correct collaboration partner and to offer support for accomplishing projects and business targets in a collaboration instead of individually, another motivation which triggered the current research was the lack of specialized algorithms in each of the components part of the proposed business process based solution.

The collaborative solution proposed in the current research has the following important components:

- **Compatibility Factor Component**: a component able to compute a similarity value for two or more organizations from operational perspective.

For this component, we used adapted graph comparison algorithms able to consider two or more business processes and calculate the compatibility between them. The similarity should quantify how similar two or more organizations perform their actions from operational perspective. After performing a literature review, we concluded that the area of business process comparison has lacks. Algorithms use a syntactic business process representation and could not be used without a clear standardization step performed in advance, which implies manual actions and a lot of effort. We targeted our solution to be fully automated. We extended the business process comparison algorithms with semantics. We planned to achieve semantic interoperability of heterogeneous business environments. We've extended business process comparison algorithms to model one-to-many and many-to-many relationships permitting an evaluation for business process representation with different granularity and sub-process consideration in business process representation. We've considered ontology-based semantics to achieve the interoperability.

- **Collaborative Composition Component**: a component able to propose a collaborative business process followed by organizations to achieve common business results and to perform individual projects in collaboration.

One objective we've taken into account for the current research was that our collaborative solution should be independent of the business process representation languages. The existent business process merge algorithms are specialized for business process representation languages. We proposed in the current thesis an algorithm able to identify a collaborative solution out of a set of input business processes which could be described in any representation language. The solution could be constructed if there is a good compatibility, organizations have similar operational profiles.

- **Rule-based Business Activity Monitor Component**: a component able to track business process execution and to ensure the operational support, trigger alerts when the execution is deviated from the expected behavior.

When business processes are not formalized, we used process mining techniques to extract and attach business processes to business cases and store them in the business process repository. Process mining operates with a standardized format eXtensible Event Stream (XES) for execution logs representation. After analyzing existing Business Activity Monitoring (BAM) solutions, we noticed the inexistence of a BAM solution operating with execution logs represented in XES and allowing the end user to define the expected behavior. We proposed for the current research a BAM solution operating with process mining data representation which ensures the operational support and information in real time while the execution could still be corrected. The rule definition language is custom, it covers the needs for business process representation, it can be easily applied as it uses logical operators and it allows the definition of rules at trace and event level.

- **Business Goal Validate Component**: a component able to verify the collaborative business process based solution for inconsistencies and to validate that it could be used to achieve the predefined business objectives.

Assessing the quality of the resulted business process model is a crucial step in achieving good business results and accomplishing collaborative business targets. Once the collaborative solution is constructed, we've considered validating it. If the quality of the process model could be analyzed by using conformance analysis techniques and computing several metrics, we've noticed the inexistence of solutions to ensure a business process is synchronized with the business objectives defined. We've used functional testing paradigms which propose checking if a system conforms to a set of requirements for validating the automatically composed process model. In our case in which we model collaboration, the requirements are represented as the business targets for the collaboration.
Validation of process models increases the confidence in the automatically obtained collaborative solution as the organizations have the guarantee that the business processes reflects the expectations.

- **Improve Component**: a component able to detect difficult areas in business process execution and use operational log information for improvement

There are several directions in the current research proposed for improving business processes. Most of them have a clear purpose: improve the quality by eliminating bottlenecks and problems in business process representation.

We've considered for the current collaborative solution a different approach. Processes inside organizations are meant to automatize the work. The process models should be suitable, must be understood and accepted from functional perspective by the people executing the activities. We've considered the logs of the custom BAM solution proposed and we've analyzed the rules constantly broken. Such results could show that a business process is difficult to follow in a specific area and must be improved. We've decided to allow the evolution of the business process by responding to business rules constantly broken and including in the model the feasible updates. Practically the business process is not enforced because we targeted rebuilding it by considering the monitoring results and statistical information. Enforcing a business process inside an organization could limit the power of expression for the team members to perform actions and to obtain results.

Practically besides proposing a new direction in the area of collaborative business process management and ensuring the technical support, we've considered new directions for each of the components included. We've contributed in the area of business process comparison, merge, improvement and business activity monitoring.

## 2.4. Objectives

We've started our research by considering the following objectives when designing the collaborative business process based solution.

- **Design an end to end solution able to support a multi-organizational collaboration**

Certainly there are lacks in all areas of collaborative business process management. Our goal was to offer a complete collaborative solution able to support an alliance between two or more organizations, business process based. The start point is the identification of the most compatible collaboration partner; an easier setup is ensured with this action. As organizations use similar operational components, a common business process could be constructed with input from all participant organizations. The common solution is monitored, validated and improved for achieving good business results and accomplishing pre-defined business targets.

- **Embrace diversity**

As a general trend in the business environment, a direction also taken into account was the specific in the execution of activities. Each organization and each assigned user has a personalized mode of accomplishing a task. Such particularities could bring benefits to the realizations of business targets. The collaborative process model is not enforced but instead the input resulted from execution logs are used for optimization and improvement.

- **Identify from the beginning the chances for success**

Most of the collaborations are doomed to failure from the beginning as there is a high incompatibility between organizations. If cultural and social differences are difficult to model, we propose qualifying the compatibility from operational perspective. Organizations which use similar processes have a higher success rate in realizing projects in collaborations.

- **Abstract from business process representation**

Organizations could represent their operational layer in different business process representation languages. However we believe that this should not stop them if they want to engage in collaboration. We've accomplished this objective by reducing the business process representation to directed attributed graph format.

- **Abstract from domain areas**

Information Systems as the one we propose address functions on different level of abstractions and operate with heterogeneous information. We targeted to achieve semantic interoperability, a mutual understanding of the data. Business processes could be created in different modeling languages using a specific vocabulary.

- **Ensure operational support at execution**

One paradigm which became a trend in business nowadays is prevention over inspection. We've taken into account exploiting the information in the event log data during business process execution to ensure the correct handling of a case. The major goal is that corrective measures should be taken while the results are not endangered.

- **Tailor solution considering the profile of the organization**

Business processes which support the collaboration should not be enforced but instead the information about how the users perform the tasks is used further for customizing the collaborative solution considering the profile of the users in the organization. The customization is performed on a consistent basis and under supervision of a higher authority.

- **Increase confidence in the collaborative solution though validation**

Validation techniques we proposed are meant to check the quality of the collaborative process model and to ensure the business targets established for the alliance could be accomplished if the execution follows the predefined process model.

- **Support collaboration between organizations without formal business process representation**

In case organizations don't have a Business Process Repository and formalized business processes associated to business cases, we propose using process mining tools and algorithms to extract them. The resulted business processes are used further for identifying the most suitable collaboration partner and computing the collaborative solution.

- **Solution is fully automated**

Even if improvements are performed upon manual actions, the end-to-end solution is fully automated and supports a collaboration business process based between organizations.

## 2.4. Proposed Solution

The solution we propose is presented in Fig 2.4.1.

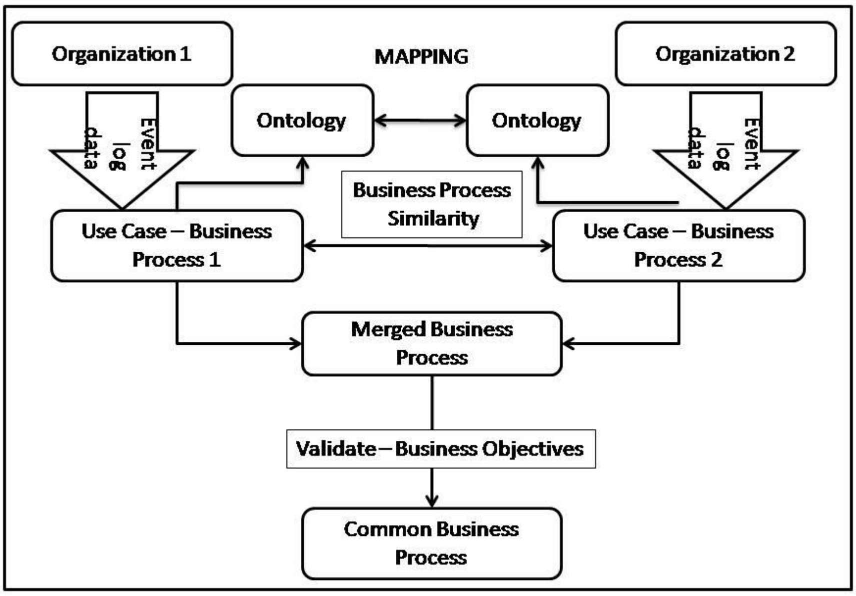


Fig. 2.4.1 Collaborative business process solution

Two or more organizations decide to collaborate on an isolated business case for which a business process is not formalized inside the contributing organizations. But instead there is an availability of event log data able to track the operational activities executed to achieve results. Once the log data is pre-processed for eliminating noise and converting to a standard format recognized by process mining algorithms, the business processes could be discovered and a Business Process Repository (BPR) could be created as a common storage entry point.

The business processes formalized are the base for further processing. The similarity is calculated for qualifying if the organizations have a good chance to synchronize their operational activities to achieve the expected business results. A high similarity leads to an easier and automatic setup phase.

Once a high similarity is calculated, a collaborative business process based solution could be identified able to support a business case executed in collaboration. The result is validated with event log input data and with shared business objectives represented as test datasets.

The merged business process is improved further if any malfunction is detected and will be monitored according to the targets. By monitoring the execution, the operational support of the business process execution is ensured.

## 2.5. Achievements

- **Context dependent business process similarity**

We've started from the assumption that collaborations should be started project based with different collaboration partners, specific to the business case.

Such a dynamic collaborative environment should rely on an easy and quick solution with a straightforward setup. To identify the most suitable collaborative partner, we proposed for the current research performing a comparison of the business processes used internally inside organizations.

One achievement is the abstractization of business process representation. The business process representation language should not be a barrier in an automatic collaborative solution.

Another achievement is the computation of how common are the business processes used inside organizations for achieving common business targets, the possibility to calculate the similarity dependent on the context. The business process similarity considers the topology of the process graph and the labels of the activities composing the business process. If collaborating companies belong to the same domain area, a syntactic string similarity could be used for label match. To allow business process comparison and similarity calculation independent of the representation of the text describing the business activities, we proposed and evaluated using semantic label match the similarity between business processes.

Another direction was the possibility to calculate how similar business process are if the representation has a different granularity. One business process could be represented in more details than another business process. We've model a one-to-many relation between process models with this approach.

To limit the search space for a semantic label match, we've used an ontological space instead of a full semantic space. The advantages are relevant concerning resources used and the faulty interpretations of business process activity names.

For modeling a many-to-many relation between business processes, we've proposed extending the business process comparison algorithm to consider subprocess composition of the business processes. With this improvement proposed, we've synchronized the solution proposed in the current research to the trend in business process modeling: modularity in design.

- **Merge of business processes and identify a collaborative solution**

Starting from the premise that a collaborative solution business process based should be composed of input from all participant organizations, we've proposed a merge algorithms able to take two or more input business process and to propose a single one. The resulted process model captures the common behavior which should support the collaborative business case. The algorithm is fully automated.

- **Functionally validate a business process**

The resulted business process model is validated from structural and functional perspective. Structural validation uses conformance checking techniques which were already available in the literature. Functional validation of a business process model is an enhancement proposed in the current research. Functional validation checks if the resulted business process is aligned with the business objectives. Functional validation is performed by generating an event log data compliant with business objectives and applying conformance checking techniques on the process model with generated test dataset as input.

- **Rule based monitoring of business process execution**

We've proposed a custom Business Activity Monitoring (BAM) solution for ensuring operational support. Deviations are detected in real time and owners of the activities are notified in time allowing them to take action. The rule-based monitor system is important as it operates with event log data format recognized by process mining algorithms. As none of the rule expression languages were suitable for business process monitoring, we've proposed a custom rule definition language, simple, which allows definition of rules at event level and at trace level. Rules at event levels check the properties of the events in correlation. Rules at trace level check the order of the events inside a trace. The order considers the properties of the events interconnected in a trace.

- **Enhance business process with input extracted from usability patterns**

We've proposed an optimization of the resulted collaborative business process model. The novelty is consisted by the source of the improvements. We think that business processes should not be dictated but instead these should be commonly understood by all users executing activities. Because of this, the patterns of incorrect execution are analyzed and input could be extracted for further process enhancements. Each accepted pattern resulted from a broken rule at runtime is transformed in a set of graph operations: insertion and deletion of nodes and edges, and the business process is upgraded considering these changes.

# 3. STATE OF THE ART

## 3.1. Process Systems

### 3.1.1 Process Mining Techniques

Extracting meaningful information from the log data written by Information Systems (IS) became nowadays an important target for many organizations. Operational log could represent a solution for organizations which don't have standardized business processes or any other formal representation of the process models used in specific business cases. Process Mining techniques are introduced with the purpose of automatically discovering a process model by analyzing the event log data recorded by enterprise systems. Process Mining is composed of a set of disciplines, techniques, tools, methods to support process improvements. Process Mining could be considered an important technique of Business Process Management (BPM); the discipline that manages the operational composed of process models and combines information technology and management sciences for supporting the business cases of organizations process based.

Process Aware Information Systems (PAIS) are systems actively involved in the management and direction of the processes executed using a process model as reference.

As stated in [3], in BPM different phases of managing a business process are identified. The first phase is the design phase in which a business process is constructed and transformed in a running system afterwards in the implementation phase. Once a business process is started and executed, the next phase is the monitoring in which the processes are supervised for any needed changes. Changes in business process models are handled in the adjustment phase and in the diagnose phase the process model is evaluated.

Process Mining as discipline includes a set of data mining and machine learning techniques for discovering, monitoring and improving the real processes [3]. Process Mining was introduced as a discipline due to the characteristic of Information Systems which operate with unstructured data.

The first part of process mining is discovery, extraction of a process model out of an event log data. Once business processes are formalized inside organizations, conformance process mining techniques are used to check the feasibility of the extracted process model by using as a reference the event log data.

Conformance techniques are used to detect deviations and to analyze the effects of such deviations. Once a business process is valid, after applying conformance checking techniques and retrieving the expected results, the next type of process mining area is enhancement.

A process model could be extended and improved further by using the recorded log information. The improvement could be triggered by the quality of the process model, the model could be repaired to reflect the expected behavior, or the model could be extended further for including additional behavior. The process mining phases are reflected in Fig. 3.1.1.

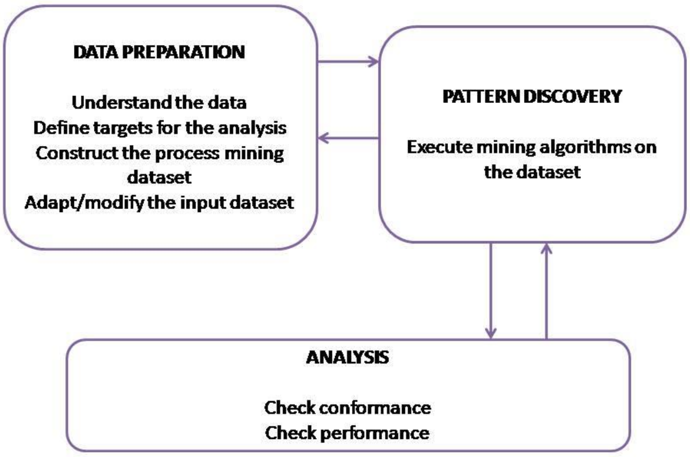


Fig. 3.1.1 Process mining phases

Process Mining allows an analysis over the existing processes inside organizations from a process oriented perspective [3]. Process Mining operates with event log data which could be extracted from different kinds of sources: well structured or less structured. When analyzing the log data some pre-conditions must be considered: the event log refers to a single process, the event recorded can be associated to activities and the events are ordered. An event is characterized by attributes. A set of events compose a case and a process instance is composed of cases.

As event process logs could be extracted from different data sources, a common standard format is needed for log representation. This format is XES – eXtensible Event Stream. XES uses the concepts of event and trace as a sequence of events. Events and traces are characterized by a set of attributes, mandatory or optional. The events are ordered by timestamp when a trace is composed.

According to [4], the process discovery problem could be formalized: if L is the event log specified by the XES standard, a process discovery algorithm is a function that maps L onto a process model such that the model is representative for the behavior seen in the event log. As the discovery of the model should be representative for the event log data, a measurement is defined to quantify attribute representative. This is described in detail in [3] and uses four metrics:

- **Fitness**: the discovered model is a projection of the behavior in the event log.
- **Precision**: the discovered model should not support a behavior different than the one in the event log.
- **Generalization**: the discovered model should be a generic representation of the behavior in the event log.
- **Simplicity**: the discovered model should be as simple as possible.

The simplest algorithm for process discovery that we analyzed to assess the feasibility for our needs to extract process models from real life event log data was the α-algorithm presented in details in [4]. The **α-algorithm** is one of the first process mining algorithms able to deal with concurrency. Unfortunately α-algorithm does not handle noise very well, which is the case of real life data. Even if the log is complete, it has several other limitations. As it is based only on the event log ordering relation, many process models can be different but still produce the same traces. Also the algorithm has difficulties in the area of loops as frequencies are not taken into account. α-algorithm is very sensitive to noise and incompleteness. As it is able to discover only Petri nets and we are abstracting from business process representation in our study, we needed a process discovery algorithm able to support any business process representation.

Heuristic mining algorithms described in [5] and [6] take frequencies of events into account and sequences when a business process is extracted. Less frequent paths are ignored, not considered for the model. Heuristic Miner only uses the order of the events within a case [6]. A dependency graph is constructed in an initial phase, each dependency relation is quantified by using a metric frequency based indicating the certainty in a dependency relation between the two events. Metrics are presented in [7].

Several other advanced process discovery algorithms are analyzed for our study such as evolutionary genetic algorithms, but for our objectives of formalizing business process models inside organizations, we consider that the Heuristic Miner algorithm is suitable.

Once the result of the process mining discovery algorithms is available, in the last phase an analysis of the results could be performed which calculates and interprets the performance indicators, checks the conformance of the resulted business process by identifying bottlenecks and cycles. Conformance checking is an important method which could be used when the process definition and the event log are given as input. Conformance checking associates the events in the log definition to activities in the process model. The goal is to find the differences between the model and the execution. Conformance checking is relevant for business alignment and auditing [3].

We used conformance checking techniques for business process validation. The resulted process model which could be used in a collaborative scenario is validated from business objectives perspective. Conformance checking techniques could be used to fix models not aligned with reality.

The interpretation and actions performed of non-conformance depends on the purpose of the model. In the case of a descriptive model like the collaborative business process based solution is, any difference with the event log data means that the model is not able to capture the desired behavior and it should be improved. We are using such information for updating the business process and ensure it is accepted by the people involved in the execution. On the other hand, in the case of a normative model which is enforced, the differences could indicate deviations which should be handled and controlled accordingly.

Performance problems could be identified by checking the timestamps and the frequencies of activities.

Process Mining allows control flow perspective mining extensions:

- organizational perspective
- case perspective
- time perspective

Organizational mining can be used to get an overview of the organizational structures and social networks. Organizational mining allows the extraction of the organizational perspective by using the resource attribute of the event log input data. Organizational mining offers information about organizational structure, work distribution and work patterns [3]. Social network analysis techniques are used for this purpose. Social Network Analysis (SNA) is a collection of methods, techniques and tools for the analysis of social networks. In [8] the authors apply the results of SNA to event logs in enterprise information systems for extracting social networks information from this type of data. SNA is performed from two different perspectives, sociocentric by using the interactions in a defined group and considering the group as a single entity; and egocentric direction which is focused on the individual and the relations of the individual within a group. A social network has a graph representation in which the nodes are the people involved in the activities and the edges are the relations between people.

A number of metrics are defined to analyze social networks and to characterize the role of nodes in a network [8]. Causality metrics verify for individual cases how work is handed over among performers. A work is transferred from one individual to another if there are consecutive activities where the first is accomplished by the first individual and the second is completed by the second individual. Indirect succession could also be considered using a causality factor β representing the number of activities between one activity is completed by the first individual and one activity is completed by the second individual. Another related metric considered is subcontracting, counting the number of times the second individual executed an activity between two activities executed by the first individual [8]. Metrics based on joint cases count how frequently two individuals are performing activities for the same case. Individuals working together on cases are stronger connected than individuals rarely working together [8]. Metrics based on joint activities focus on activities individuals execute. People executing similar activities have stronger relations than people doing different activities. If the activities executed by an individual compose a profile, the distance between profiles could be computed with this metric [8]. Metrics based on special event types use the type of events for evaluation.

Besides metrics, other SNA concepts are used for analyzing the organizational structure. Centrality is one important factor to qualify a node in a SNA as it analyzes the position of one organizational entity in the whole social network. The node is very central and could be a source for bottlenecks in the execution of activities if all other nodes are in short distance to the given node. Cliques are groups of nodes strongly connected to each other.

Process mining is used to process event log data of already executed scenarios to extract process models and to permit further analysis. For ensuring operational support the running cases are analyzed. Partial traces composed of events are analyzed in real time to detect, predict behavior or recommend corrective actions [3]. When a partial trace is compared with a normative model, an operational support activity is in progress. Such checks could identify violations in process execution. Predict is the operational support activity that makes assumptions about the following events. Recommend is the operational support activity that guides the user in executing the actions. All operational support activities are based on extracted process information and running cases. Operational support includes an immediate response when any deviation is detected. If a prediction activity is also performed, the result could be a performance indicator such as the remaining flow execution time, or a categorical response time. In case of recommendations, a partial trace is analyzed and the response about next steps is prepared. Recommendations are given respecting specific goals such minimizing execution time, minimizing resource usage, minimizing total costs.

Several new perspectives on the possibilities of process mining algorithms have been recently considered. One is the behavior analysis of real life process definitions. The method proposed in [9] is a reverse engineering technique able to cope with real-life event log data from distributed systems extracted from multiple components of different systems. The proposed technique allows obtaining precise business process models as it is based on an integrated view across distributed systems. The method proposed is an enhancement to the existing process discovery algorithms which we used for extracting business process models of participating organizations to a collaborative scenario.

Several process mining implementations are available on the market, academic and commercial. For our research we used ProM 6.3 Toolkit, an open-source framework.

Initially for a hands-on experience with process mining algorithms and for testing the capabilities and identifying the most suitable combination for our research, we applied them on a real log data in the software industry, the event process logs of a software development process used in the IT department of an automotive industry company. Our results are captured in [73]. The process followed for handling a new release is not enforced initially, it is a declarative process model used as a reference. The purpose is to improve it further by analyzing with process mining algorithms and use the upgraded process model in further releases. We used process mining algorithms to identify deviations and hidden information, difficult to observe without mining techniques due to the complexity of data.

The development is outsourced to an external company and the process followed for a new product release was defined by the external customer as a project requirement. The provider company was allowed to customize the process for overcoming the internal organizational constraints. The updates to the process model could cause deviations; these deviations are analyzed with process mining techniques. We also wanted to extract hidden information about the process, difficult to identify due to the wideness and complexity of the input data.

As any process mining analysis, we started ours with clear targets:

- Identify the real process model and compose the list of differences with the descriptive process model used as a reference
- Identify an enhancement of the process model to avoid the rework of items due to conflicting requirements, unclear specifications, and missing specifications. Such problems found after a release is performed induce risks and additional costs and must be avoided. The process model supporting the project must offer the means for a good performance and good results.
- Calculate statistics about the quality of the issues (items rejected and reasons for reject, items included in a release, number of problems reported)

For applying process mining algorithms, we needed as input the event log dataset which was exported in csv format from an internal activity tracking tool as described in details in [73]. The event log contains information about the states of the items and the users executing activities to update the states. As process mining operates with XES format and our exported log data is in csv format, we performed the translation by using OpenXES. Once the input data is prepared, we targeted obtaining the process model out of the event log data and we applied for this purpose the Heuristic Miner algorithm. The result obtained is represented in Fig. 3.1.2, the business process used by the engineering, support and management teams for achieving the business objectives.

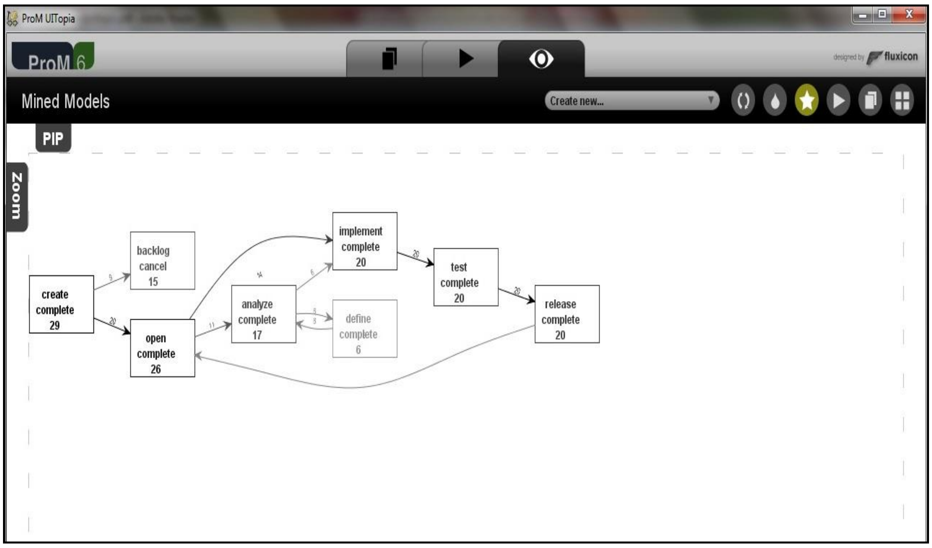


Fig. 3.1.2 Change management business process

Once the real process model was available, we performed an analysis for identifying the differences between the model obtained after pattern discovery phase execution and the theoretical model defined when the project was started. We reached to some interesting conclusions as important differences between the expected process and the real process are noticed.

Once the process model is extracted out of the event log data, we've scheduled an analysis of the statistical results for quantifying and improving the performance. Improvements are scheduled for future project releases. Following the analysis, we performed an evaluation of the performance of the real process model by analyzing the statistical results. Low process execution performance is analyzed and possible improvements are extracted and scheduled to upgrade the model for future project releases.

Our conclusion was that process mining techniques are able to give valuable information about the processes followed inside organizations and such information could be used further for improving the business results and enhance the performance. The disadvantage is that applying process mining on a dataset is not a straightforward activity as it requires a deep understanding of the data and a lot of expertise in interpreting the results.

We concluded that we could use and adapt process mining techniques for our purpose to identify methods to support and improve collaborations between organizations.

One major direction for process mining algorithms is the scalability of process discovery algorithms. Even though several techniques are presented, few of them combine scalability with quality expectations which implies operating with large data and producing correct models, without deadlocks or anomalies. In [10] the authors propose a framework for process discovery in large data context by applying a divide and conquer strategy. Before this approach process mining was used mostly for small academic datasets, limitation which represented an incompatibility with the real-life context. With this direction of scalability to big data the process mining algorithms are feasible for usage on real datasets. Another limitation of the current process mining algorithms is that the full processed data log must be present in the memory for analysis. A big event log could not be processed under these restrictions. The proposed algorithm has as a major target traversing only once the event log. This property would eliminate the constraint of having the complete event log data in the memory and it should produce process models of the same quality.

Process Mining algorithms offer support for the verification of process models by computing metrics: fitness, precision, generality, simplicity. These metrics are used for evaluating the quality of process models if datasets are split before analysis.

We consider process mining techniques in a cross-organizational context. As collaborations could be initiated with organizations with a less formal operational context, process models extracted with process mining techniques represent the start point for initiating collaboration with a specific partner. Once business processes are formalized, the similarity factor could be calculated and a collaborative solution could be constructed. The start point for the proposed technique is represented by the business processes associated to business cases inside organizations. Though there are many challenges in the area of process mining such as coping with event log dataset with inconsistencies or process log data in a distributed environment, operating with large datasets, dealing with concept drift and custom situations, dealing with attributes of event logs, our research is not oriented on improving process mining algorithms results in different contexts but instead to use process mining process models as an input for identifying if a collaboration could be easily set up in between two or more organizations and to compute a business process based solution able to support the collaboration.

### 3.1.2 Process Modeling Languages

Petri Nets are one form of representing a business process, with a major disadvantage of allowing the representation of concurrency. As stated in [3] a Petri net is a bipartite graph consisting of places and transitions.

A Petri net is a triplet $N = (P,T,F)$ where $P$ is a finite set of places, $T$ is a finite set of transitions such that $P \cap T = \emptyset$, and $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs, called the flow relation.

A marked Petri net is a pair $(N,M)$, where $N = (P,T,F)$ is a Petri net and where $M \in B(P)$ is a multi-set over $P$ denoting the marking of the net [3]. One sample of a Petri Net is represented in Fig.2.1.2.1 [3].

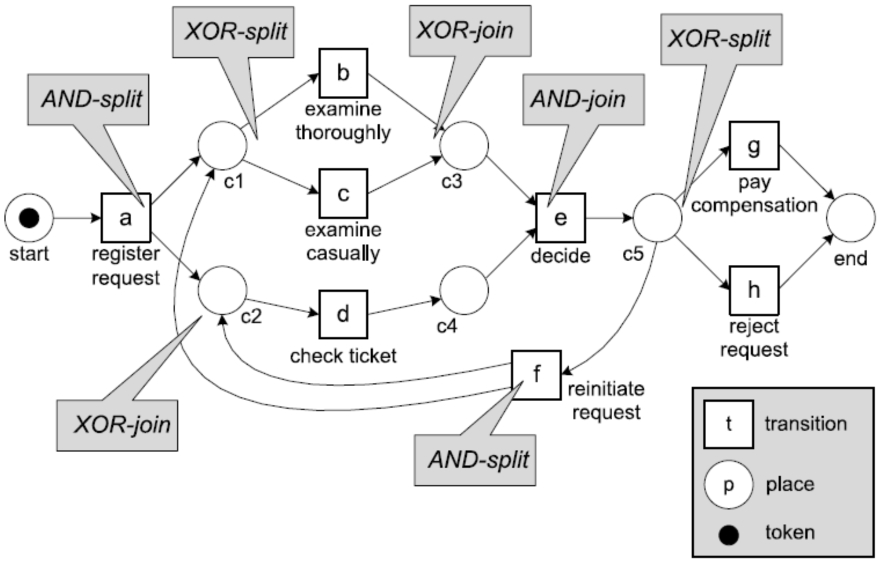


Fig. 2.1.2.1 Sample Petri Net

A labeled Petri net is a tuple N = (P,T,F,A, l) where (P,T,F) is a Petri net, A ⊆ A is a set of activity labels, and l ∊ T →A is a labeling function.

One major advantage of Petri Nets as business process representation languages is the wide range of techniques and tools for analysis existent on the market. Petri Nets are able to cope with timestamps and ordering of tokens.

Workflow Nets are another possibility for representing business processes. Workflow Nets are subsets of Petri Nets with a clear source place where a process is started and a sink place where a process is completed [3]. Workflow Nets as business process representations are suitable as modeling languages because the process models are instantiated for each case and there is a start point and an end point. In between start and end the activities are executed according to a clear procedure. Workflow Nets are not the safest representation form of business processes.

YAWL is another workflow modeling language available, constructed on patterns. YAWL offers direct support for patterns and could be seen as a reference implementation for most of the workflow patterns. YAWL operates with tasks as activities and conditions for interconnecting tasks. Tasks could have a split or join semantics and could be either atomic or composite, which allows a hierarchical structuring of the model. In Fig 2.1.2.2 a YAWL extracted from [3] is displayed .

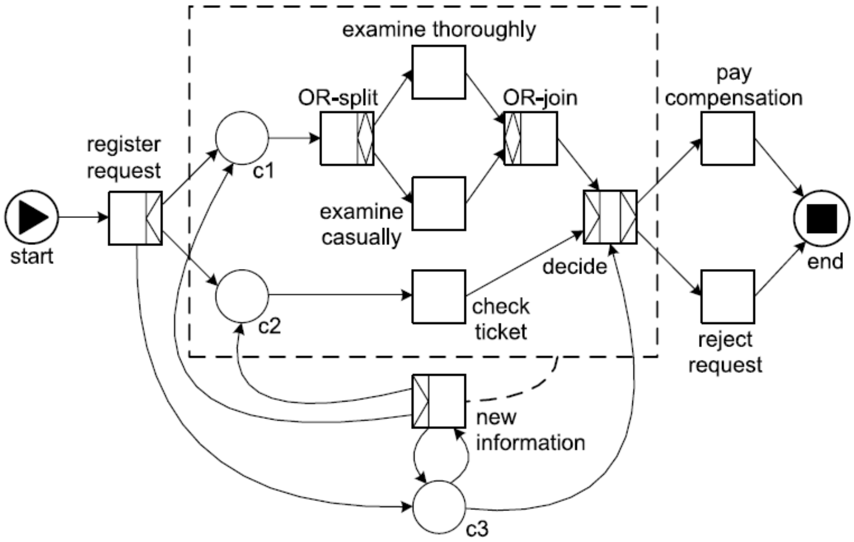


Fig. 2.1.2.2 Sample YAWL

One of the most known processes modeling language representation is Business Process Modeling Language Notation (BPMN) which is standardized. BPMN works with a big number of modeling elements [1]. Tasks are units of work, atomic activities within a process flow. Non-atomic complex work that can be split into smaller components is modeled as a subprocess. Events either internal or external could modify the business process execution. Gateways control the business process flow. Exclusive gateways ensure that only one flow is traversed, inclusive gateways create parallel paths which may or may not result in executing multiple flows. Event based gateways permit modeling paths based on events. Sequence flows connect flow elements and message flows ensure the communication between pools by sending messages. An example of a BPMN representation of a process model can be found in Fig. 2.1.2.3 [1].

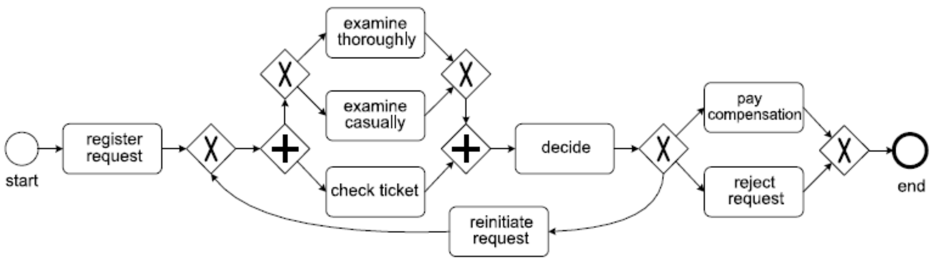


Fig. 2.1.2.3 Sample BPMN

Event driven process chain (EPC) represents another possible notation for process model representation. Activities are modeled as functions with input arcs and output arcs and splitting and joining are modeled as connectors. Events (start, intermediate, end) are also represented in EPC. An EPC representation of a business process is displayed in Fig. 2.1.2.4.

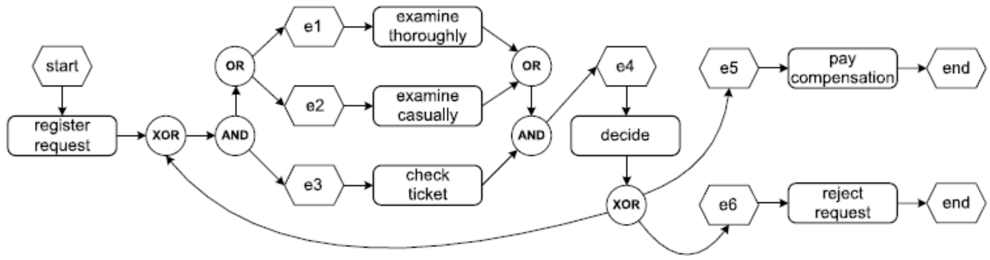


Fig. 2.1.2.4 Sample EPC

Several other modeling languages are available for business process representation. One of the most important targets for our collaborative solution was to abstract from any business process modeling notation. The algorithms for calculating the similarity between organizations and identifying a business process model able to support collaboration for a specific business case abstract from business process modeling languages and operate with directed attributed graphs as input.

## 3.2. Collaborative BPM

Since BPM started gaining more and more attention as a field in operations management and optimization, an inter-organizational collaborative framework supported by methodology became a subject for research. Different forms of middleware were introduced to allow the integration of BPM across organizations.

One early direction is represented by message systems and publish/subscribe mechanisms which ensure the process automation by allowing the application to post and react to events. Process coordination is achieved by using message queues. One major advantage considered when designing such a collaborative solution was the communication across heterogeneous networks and systems temporary offline. Such a BPM solution is presented in [10].

CrossFlow was one of the projects with a practical implementation aiming to offer a collaborative BPM solution in dynamic virtual enterprises. CrossFlow is designed by considering outsourcing as a business collaboration form. The system is able to take the decision to outsource during the execution of the process and the provider is chosen dynamically. The interaction is based on contracts. Service providers promote their services in templates filled by consumers. The contract defines all the relevant details needed for execution. CrossFlow approach is focused on integrating consumers and providers.

In [12] a process framework is described in which the process model involves multiple parties, each one having a specific role in the process. The collaborative process is constructed following a set of rules. The process definition is based on an integration protocol and the process execution is split between several workflow engines.

In [13] a more theoretical approach is presented and several forms of interoperability are identified:

- **capacity sharing** in which a centralized control is enforced

- **chained execution** with a common business process constructed from a number of disjoint processes executed by organizations sequentially
- **subcontracting** in which the control is kept by one organization and sub-processes composing the common business process are contracted to organizations
- **case transfer** in which each organization uses the same process and the process specification is replicated. Cases can be transferred from one organization to another.
- **exploiting commonality** in which the process is cut into parts which could be executed concurrently. The definition of a sub-process is local and the protocol used for communication is public.

Also in [13] two forms of cross-organizational processes are identified, horizontal and vertical partitioning, by considering two dimensions: case and process. One direction is represented by the distribution of cases across organization. This is called vertical partitioning. Another direction possible is to split a process into components and assign each organization to implement a component. This direction is called horizontal partitioning.

Several process mining projects consider process mining in inter-organizational context. Three of them are EDImine, CoSeLOG and ACSI. EDImine project [14] extends the current process mining research to apply process mining algorithms to inter-organizational business processes. CoSeLoG project [15] is oriented on a particular application domain: dutch municipalities for analyzing the differences and commonalities. ACSI project [16] uses artifact-centric modeling to support service collaborations in open business networks. All the projects are referenced in [13].

Another project targeting inter-organizational collaboration, a software platform for B2B in e-commerce focused on small and medium sized enterprises is WISE as described in [69] and [70]. WISE is constructed on a workflow engine that administers and controls processes in inter-organizational scenarios. Such a process is composed of a set of services, each one is assigned to a participating organization. WISE is able to orchestrate workflows of different collaborating organizations in which the ownership of components is assigned to organizations. Components could be sub-processes or external services. WISE is able to orchestrate shared business processes in collaborative scenarios but it lacks in the area of flexibility and visibility. WISE targets a separation of concerns between participating organizations, different than the purpose of our collaborative solution in which we target sharing the same business process, resources of all participating organizations execute the same activities for business results.

The new trends for collaborative BPM solutions involve applying semantic technologies to BPM. Cross organization collaboration is identified as a requirement for Business 2 Business Integration and the limitation of current BPM standards: Business Process Modeling Notation (BPMN), XML Process Definition Language (XPDL) and Business Process Execution Language for Web Services (BPEL) are that none is satisfying this requirement [17]. Because of the failures in defining a collaborative BPM solution, the ontology based direction is introduced.

Semantics Utilized for Process Management within and between Enterprises (SUPER) project described in [18] is one of the most popular SBPM solutions which switch the business process control into the business area instead of IT area by ensuring a semantic is accessible to business experts for business process definition. SUPER enriches BPM standards with semantics by using BPMO – Business Process Modeling Ontology, and sBPEL [19].

A solution for inter-organizational business process management based on XML Nets in the area of e-commerce could be found in [20]. The authors propose an extension of Petri Nets as a modeling notation for business processes called XML Nets, a graphical modeling language for the flow of XML. XML nets bring an improved process-oriented organizational structure. One major accomplishment of the solution presented in [20] is the possibility of fragmentation of an XML Net and execution in a distributed environment. Globalization and the trend for flexibility as a business direction require adaptable solutions and the solution proposed in [20] is based on splitting the centralized workflows.

The literature has few solutions for inter-organizational BPM, a lot of them are oriented on Web Services. Process Mining algorithms are applied either to the message logs of Oracle BPEL as in [21] or in the context of IBM WebSphere [22].

Most of the inter-organizational research in the area of BPM is oriented on process management in a distributed environment. Such an approach could be found in [23] and [24] in which techniques are developed for event correlation and process discovery from web service interaction logs. Events are correlated in a process view and a collection of process views is called a process space. The authors describe the correlation rule patterns and propose a formalization of such rules and an heuristic to derive the rules. The authors propose the key-value correlation.

Event based correlation is also considered in [25] for Electronic Data Interchange (EDI) systems used in Business-to-Business e-commerce. The authors analyze on a real case study the inter-organizational business processes of an automotive supplier company by analyzing the messages received from partners. Observed messages are transformed in an event log which performs the correlation of individual messages into process instances. This is followed by an application of process mining techniques to reveal the inter-organizational business structure.

A process collaboration framework based on semantic agents is proposed in [72] in which the process collaboration information is transferred between different processes by semantic agents. Agents are able to access different business process management systems and extract information expressed in process ontology. An agent is a software component which offers support for human-machine collaboration. Shared process ontologies solve the problem of the heterogeneity of knowledge between agents. The framework proposed is just theoretical; plans exist for developing it in details.

We are defining the collaborative solution as based on a new business process identified after the analysis of the business processes used in partner organizations and by handling the input from all participating organizations. Once the common process model is identified, the resources and activities are shared and the control of the process is performed centralized. The partitioning is not horizontal or vertical, each organizations is responsible of specific areas of the business process and participates with resources for realizing the common business goals. We also added semantic capabilities to our solution. Collaborations could be started in different domain areas and cultural environments; a syntactic solution cannot cover custom business process representation and cannot ensure the interoperability.

## 3.3. Sourcing

Sourcing is a wide concept used in inter-organizational collaboration. Patterns for sourcing are proposed in [71]. Sourcing is based on the concept of inter-organizational business process management in which the dynamic process model is composed automatically from sub-processes of the participating organizations. Dynamic in sourcing means that collaborating organizations are found on the market, this is possible by exposing means for providers to offer services and for consumers to access the services and integrate their processes. Sourcing includes definition and identification of processes, possibilities for the participating organizations to find each other, and means for the setup of the collaboration. Practically our end to end solution could also be considered a sourcing framework as the collaborative process model is automatically and dynamically composed, organizations could qualify their compatibility from process perspective and could decide if collaboration is feasible.

As defined in [71], sourcing is a framework for harmonizing the inter organizational business processes in a service consumers and providers environment, in a business to business supply chain collaboration. Sourcing as concept extends previous collaborative solutions proposals as it includes mechanisms for monitoring and validation if the expected business output is achieved. Monitoring and validation of the business objectives are also directions that we considered for the collaborative solution proposed.

Clear implementations for Sourcing concepts are missing in the literature but instead patterns for standard sourcing solutions to recurring problems are identified. Sourcing is more a concept, it's not related to any technological aspect or to any real solution existent on the market. Sourcing manages the business process perspective in an intra and inter organizational scenario from three different perspectives: conceptual, business and technological. We focused our research on the technological side, to offer tools and techniques for supporting easily collaboration when the time available for the setup is limited.

## 3.4. Process Compare

In [26] the authors present a solution for ranking business processes stored in a Business Process Repository by comparing them with a given process model. The authors propose a graph matching algorithm for calculating the similarity between two business processes and offer a heuristic alternative to the search problem as the graph matching technique is NP-complete. The authors demonstrate their algorithm on a Business Process Repository (BPR) composed of 100 process models. As a business case, the similarity computation allows detection of overlaps or duplication when a new process model is inserted in the BPR. The similarity model proposed covers two directions: text similarity: comparison of the labels that define the activities of the process models and structural similarity considering the topology of the process models. A future direction is also identified, the behavioral similarity which is based on the execution of process models.

The algorithm proposed in [26] is using the graph edit distance, a graph matching problem NP complete. Therefore, four heuristic directions are proposed to cope with this major limitation.

We used the algorithm presented in [26] and we extended it in several directions. One is the usage of semantics for text similarity instead of syntactic algorithms, in [26] the authors use syntactic similarity for identifying if two labels represent the same activity. Semantics introduced enlarge the domain for calculating the compatibility between two or more business processes. The business processes could be compared and the compatibility between organizations could be evaluated if the companies don't use a standardized language for business process representation. As the semantics space is wide, another extension proposed was to consider domain ontology for the representation of the vocabulary and rules for the business process. We used the algorithm described in [26] to compute business process similarity by considering a one-to-many relation between business processes and also a many-to-many relation.

Graph-edit distance as presented in [26] is the minimal cost of transforming one graph into another. Each conversion is represented by a set of operations, each operation has a cost given by a cost function. The operation with the minimal cost represents the value of the graph edit distance. Operations considered are node replacement when a node from one graph is replaced with a node from the other graph; node insertion and deletion in which a node is added into or removed from a graph; edge insertion and deletion in which an edge is added into or removed from a graph.

In the case of many-to-many graph matching, in the literature this set of operations is completed by merge and split operations. We proposed a different approach for the many-to-many graph modeling relation. The estimation of the optimal sequence is a hard combinatorial problem. Because of this major disadvantage approximate method such as beam search [27] or best-first, breadth-first and depth-first searches are used. Graph edit distance as a method to compute the similarity between graphs is also proposed in [28] and [29].

For calculating the graph edit similarity of two process graphs, the mapping that induces the maximal similarity must be found. For this purpose a possible direction is to compute all possible mappings and return the one with the maximal similarity. This approach is unfeasible for large graphs as business process abstractions are, and it has a factorial complexity.

One heuristic which is also the one we used in our research is presented in [27], the greedy algorithm. The mapping between pairs of process graphs is incrementally composed. The algorithms identify all possible pairs of nodes from graphs and mark them as open pairs. In each iteration the pair that increases the similarity the most is chosen. Nodes of the pair are removed from the list of open pairs and the algorithm continues by selecting the next open pair of nodes. When no open pair is left which could be added to the mapping, the algorithm is finished. As it's a greedy approach in each step the pair that increases the similarity the most is chosen, less optimal directions could be followed. Open pairs that might increase the similarity in future steps could be disconsidered. Another heuristic proposed in [27] is the exhaustive algorithm with pruning which recursivly explores all possible mappings.

The third heuristic proposed in [27] is the process heuristic algorithm, a variation of the exhaustive heuristic in which a recursion tree of possible mappings is also built. The algorithm is constructed on an improved pruning performed; nodes are mapped accoring to their position in the process model.

The fourth algorithm as a heuristic presented in [27] is the A-star algorithm. In each step the algorithm selects the existing partial mapping with the maximal graph edit similarity. An optimization is added, mapping of nodes with different labels is avoided by using the string edit distance value is used, if it is less than a cut-off value, such nodes cannot be mapped.

In [27] the heuristic algorithms are evaluated for optimal parameter value identification. We used for our similarity algorith the Greedy heuristic proposed as it produces good results and it's faster than any other. Graph edit distance algorithms applied in specific combinations are also presented in [28], [29] and [30].

Besides the local search algorithms, another direction possible is the spectral approach. Such a method is presented in [31] in which the clusters of one process are matched to the clusters of the partner business process. Graph matching as business process abstraction aligns two graphs by matching vertices in the optimum way of keeping most of the edges. The authors present the many-to-many graph matching as an optimization problem by extending the one-to-one matching. As the many-to-many graph matching problem is a hard discrete, the authors propose in [31] an approximation method to solve it, an algorithm based on continous relaxation.

In [31] Caelli and Kosinov also use spectral matching for many-to-many graph matching, instead of a one-to-one corespondence, they search a many-to-many mapping by running clustering algorithms. Another spectral approach is given in [32] where the adjancency matrix is replaced by a matrix of shortest path distance and spectral decomposition.

## 3.5. Semantic Web

Semantic Web allows the representation of the information of Web in a meaningful manner. Content and services are described in a machine readable form which permitts annotating, discovering and publishing of services automatically. At the base for Semantic Web is the concept of ontology.

Ontology is an explicit representation of a conceptualization [33]. A formal ontology defines the terms and their relationships composing the vocabulary of an application domain and the set of axioms for representing the relationships among the terms of the vocabulary. A formal ontology is composed of specifications of vocabularies which include definition of classes, relations, functions and other objects.

Ontologies are classified based on the level of generality as shown in in [34]:

- **Top-level ontologies (upper level ontologies)**: describe general concepts like space, time, object, event, action
- **Domain ontologies**: describe the vocabulary related to a specific domain
- **Task ontologies**: describe generic tasks or activities
- **Application ontologies**: mix of domain and task ontologies

Several standards for Semantics are defined in the literature. OWL-S is the most familiar ontology representation language for semantics used mostly for Web Service description. This has been further developed into Web Service Modeling Ontology (WSMO). Web Service Modeling Language (WSML) is a group of ontology language representations and Web Service Execution Framework (WSMX) is the most relevant implementation of WSMO.

Web Ontology Language (OWL) extends Resource Description Framework (RDF) and RDF Schema. OWL defines a terminology used in RDF documents composed of classes and properties. A class represents a set of resources. OWL defines two set of properties: object and data types. Object properties identify an attribute capturing the relation between a resource and a data type value. Properties could be defined by using sub elements. OWL supports the relationships between instances.

An ontology is a pair O = (S, A), where S is the vocabulary and A is a set of axioms operating and describing vocabulary elements. Ontology mapping was introduced mostly because a single ontology was not able to support a wide distributed environment of Semantic Web. Mapping allows the existence of multiple ontologies and provides a common layer for access. An ontology mapping process is able to associate vocabulary elements of two ontologies. The mapping considers the validity of axioms. In [35] a total ontology mapping is defined. Having two ontologies O1=(S1, A1) and O2=(S2, A2), a total ontology mapping is a morphism f: S1→S2 of ontological signatures in such a way that all interpretations that satisfy O2 axioms also satisfy O1 translated axioms. A partial ontology mapping from O1=(S1, A1) to O2=(S2, A2) could be identified if there exists a sub-ontology O'1=(S'1, A'1) in which S'1⊆ S1 and A'1⊆ A1 for which there is a total mapping from O'1 to O2.

Semantic interoperability is a target in many areas. It could be defined as the ability of two or more computer systems to exchange information by having the ensurance that the information is accurately interpreted and transformed by a system. Semantic interoperability is difficult to achieve due to the semantic heterogenity. Standardization is an alternative to semantic interoperability.

Semantic heterogenity could be identified considering different types of conflicts [36]:

- **Semantic conflict**: modelers do not operate with the same set of real world objects but they visualize ovelapping sets
- **Descriptive conflicts**: naming conflicts, differences in terms, meaning of terms, homonyms and synonyms, attribute domain, scale
- **Structural conflicts**: even if the same data is used, different constructs could be chosen to represent common real-world objects

Semantic annotation was introduced for allowing computer systems to query and find specific information. Semantic annotation allows linking the ontology information to the original information sources. An annotation is additional information associated with a piece of information. In case of semantic annotation the extra information is represented by the meaning of concepts used. Annotations are ussually represented in the form of meta-data. In the area of Web Service semantic annotation improved software reusage and discovery, allowed composition and enabled integrating legacy applications as part of business process integration.

Our concern for the current theisis is achieving the inter-operability in the area of process models as we are composing a collaborative solution business process based. Annotating semantically business processes is a prerequisite of Semantics Business Process Management.

In [36] the requirements for Semantic Business Process Management (SBPM) are formalized. In the first phase of Semantic Business Process Modeling the process models are semantically annotated. Each element in the process flow has a semantic description associated by referencing an ontology concept. In [36] this step is performed by the business analyst annotating the business process components with ontology entities. The base for the business process annotation is represented by the process ontology composed separately for each industry/domain. With this enhancement, several functionalities become available as business process discovery and domain search. The major benefit noted is the possibility of reusing the process models and the discovery of process fragments able to support a business case.

To achieve interoperability two possible directions could be followed. The first one is to enable all components to use a common terminology in the communication. The semantics of the shared terminology is specified by a meta standard accepted by all components.

If an application uses a different terminology, then a transformation mapping must be established for a correct inter-connecting. Another direction possible and more flexible is that components use different terminologies for which semantics are specified using ontology. Relationships between terminologies are indirect, elements of one ontology could be mapped to elements of the second ontology. Mappings could be computed automatically and could be classified into point-to-point mapping and global mapping. Point-to-point mapping is constructed between two participating systems and it can preserve the semantics of each system. The major disadvantage is the large number of connections which could be performed. Global mapping constructs mappings between participating systems and a global domain.

The ontology mapping is a widely researched problem and in spite of the good results, it is still not fully functional. For improving the results specific techniques of preprocessing are used in advance but such manual actions are uncompliant with a fully automated solution as the one we designed.

Ontologies are formal representations of concepts within a domain and describe instances, classes, attributes, relationships between instances and events. In [61] two major directions are considered for mapping ontologies, either by using a shared ontology or by machine learning techniques. Shared ontologies are meta ontologies used for a domain which could be adapted manually to a particular domain. Mapping is performed by connecting elements of the ontologies with the high level ontology defined. The second approach based on machine learning is used in situations in which a common ontology is not available and machine learning techniques and heuristics are used to find the commonalities between the two input ontologies. The second approach is based on concepts like ontology structure, concepts, instances of classes.

In BPM ontology mapping by referencing a high level ontology is a feasible direction for process models integrated in the same domain area as the business scenario we are targeting for the current research: supporting dynamic collaborations on the same business cases between organizations when the time available for the setup process is limited. The collaborating organizations ussually belong to the same domain area so a common ontology used for mapping the individual ontologies for participating organizations is a feasible approach. However for extension reasons organizations could collaborate even if the domain area is different. For this purpose a common ontology as a reference for mapping is not a suitable idea and the second direction of using machine learning techniques must be considered.

In [62] the authors approach the topic of unifying the schema definitions. The method proposed is based on metadata for automating high level schemas definition unification and it involves three steps: the analysis of the elements of schemas that have a high affinity with each other, clustering schema elements for the elements found in the first step based on the level of affinity found and by using hierarchical clustering procedures, and in the last phase, construction of the global views composed out of the semantic clusters identified. The globals views permitt querying the information. The authors developed ARTEMIS (Analysis and Reconciliation Tool Environment for Multiple Information Sources), a tool facilitating the construction of global views.

In [63] a different approach for ontology mapping is presented. The authors present COMA, a system able to perform an ontology mapping based on source schemas. Their method is based on different data types and does not use a single matching direction but a combination of different matching approaches: hybrid and composite. Hybrid direction uses a combination of data types to match within the execution of an algorithm. Composite executes separate matching algorithms and analyzes the results of different executions. The base for COMA is schemas, either database schemas or XML files. Schemas are represented as directed acyclic graphs where schema elements are represented as nodes and links between schema elements as links. Match operation gives a value in between 0 – no match and 1 – strong similarity between elements. Initially the matchers are chosen considering context factors (input data). Matchers could be simple matchers, hybrid matchers and reuse-oriented matchers, dependent on their particularities: names, data types, structural properties, semantic information (synonym tables). The results are stored in a similarity cube. Finally the data in the similarity cube is analyzed to determine which pairs could be set for the input schemas.

In [64] the authors present Anchor-PROMPT, an ontology mapper which finds similarity terms between two ontologies. The start point is an identified pair between the ontologies called anchor. The anchors could be detected manually or the system automatically generates them. Starting with the identified anchor a new set of pairs is produced of semantically closed terms. Anchor-PROMPT follows the paths to find similar terms. The proposed solution has good results and is fully automated.

In [65] the authors present another direction for ontology mapping QOM (Quick Ontology Mapping) which focuses on speed, retrieval of the ontology mapping results in a good execution time. Accuracy is not a major goal for the proposed solution. For this purpose heuristcs are used to lower the number of candidate mappings and to reduce the complexity and improve the performance.

S-Match is another ontology mapping method based on semantic integration of schemas which are constructed independently. The algorithm is presented in [66]: the input schema is converted into a tree structure in which a note is associated with a unique number and a label. The system supports connecting other match making components and algorithms suitable in a particular situation.

Currently we are modelling the business processes used inside organizations for specific business cases and represented in several business process modelling languages by abstracting the process model in a directed attributed graph and by using the semantics associated with each organizations for the representation of the ontologies supporting the business case. Another possible direction for business process representation with less formalism involved is represented by the conceptual graphs. In [68] some directions for defining processes as conceptual graphs are presented. Conceptual graphs could also be used for finding similarities between ontologies We have disregarded this direction due to the lacks in the area of formalism and real business process implementation and execution engines. Our solution targets real business scenarios with organizations that follow processes represented in standard business process representation languages.

### 3.6. Business Process Merge

A business process merge approach is presented in [37]. The authors also use a directed graph abstraction of the business process. The similarity of nodes is computed using the string edit distance syntactic similarity. One major difference with our approach is that we are proposing a semantic string similarity comparison which interprets node labels in a context and is not based on the effective content of the string. An enhancement of the proposed solution in [38] is that the merged process ensures backward traceability, not relevant in the context of our solution. Another difference is that in our approach only nodes of the same type are compared. We consider that it’s risky to perform comparisons followed by operations with control nodes. In [37] a context similarity is used for control nodes computed based on the transitive presets and transitive postsets of nodes. Instead of context similarity, we are using for control nodes the same mechanism as for activities. The merge algorithm in [37] is based on the maximum common regions as the one proposed by us. But the rest of the nodes and edges are reconnected by adding control elements, an operation we consider risky. If a source node in first graph and its matched node have exactly one predecessor each, then a configurable XOR-join is inserted in the common graph to reconnect the two predecessors to the source [37]. Also the reduction rules applied are different than in our merge algorithm. In [37] the redundant transitive edges are deleted, consecutive split and joins are merged into a single split (join) and trivial connectors are removed. Instead of these, we are performing a transitive reduction of the common graph. The algorithm in [37] is exemplified on EPC as business process representation.

In [38] the authors present a simplified version in which the connectors are removed as they are attached to the edges. The authors in [39] construct a process model in which the sum of change distance between each process and the generic process is minimal. In [40] the authors map tasks and apply merge patterns (sequential, parallel, iterative and conditional). The research in [41] presents an approach not fully automatically.

In [42] the authors propose a merge algorithm for integrating a set of process variants into a single configurable process model. The integrated model sums the behavior of the originating models, ensures a trace back of the origin for each element. In [42] the authors propose merging a set of process models at once. As a preprocessing step, the business process models are verified and conflicts of start and end node are resolved. The merge process is simple; it is a union of all input graphs.

The algorithm presented in the current paper could also easily be extended to multiple graphs.

The approach we propose is more state machine based. All elements are reduced in the end to simple graph nodes. The similarity of nodes is calculated by using a semantic algorithm instead of a syntactic computation which brings more flexibility. Our approach is also defined over the maximum common regions and allows replacement of nodes if the similarity is high. We are applying a transitive reduction as a reduction rule once the common graph is available.

## 3.7. Business Monitor

Business Activity Monitoring (BAM) focuses on using processing events with a minimum latency, near real time [43] for calculating the performance information about business process execution. Before BAM was introduced as a direction for analyzing real-time events, BAM was limited to a passive monitoring without any support for taking action on a business process execution for improving the results. Currently BAM targets transforming the process related events into performance indications with the purpose of identifying trends, opportunities and threats for which responsibles need to take proactive or corrective actions.

In [44] the general architecture of a BAM solution is presented. A BAM solution is composed of four components:

- processing module
- monitoring module
- visualization module
- process definiton module

The BAM system we propose for monitoring the collaborative business process based solution respects the component based architecture of any BAM solution.

BAM could go even further than that, beyond computing and interpreting performance indicators, to offer functionality of filtering, transformation and aggregation of events [45].

Another possible direction for a BAM system is the predictive analytics, simulation performed for improving business process execution and predicting results. Several complex techniques are proposed for this purpose which helps assesing the quality of process models in different execution cases: possible bottlenecks or loops.

Besides the quality perspective in certain execution paths, another direction for a complex BAM system includes predicting the execution results based on evaluating the history of a business process. Such a system is called an operational business intelligent system [45]. An operational business intelligent system is able to trigger actions based on events. A BAM system is able to react in real time to an event or to a combination of events and it is able to corelate events into metrics interpreted in a context over time [43].

The base for a BAM system is Complex Event Processing (CEP), a set of tools and techniques for giving a meaning to a set of events created by a system [43]. An event is a record of an activity which happened in a system. In BPM an event represents the state change of an object within the context of a business process [46] such as process started, activity executed, activity completed. Complex events could be composed by aggregating simple events.

Another direction for complex events is the concept of causality, corelation of low level events in dependence to each other. An event is caused by another if due to the presence of the first event, the second one occures also [46].

In a BPMS system the business process is executed and produces process instances. The logging capabilities of the process engine allow creating log data on the transitions occuring in the process and all the relevant information: resources involved and activities. Logging systems could operate with all kinds of formats therefore an external system able to corelate logs and transform them into a specific standard format is needed. One of the standardized formats is eXtensible Event Stream (XES). The custom BAM solution we proposed is operating with XES format event log data. Once an event is identified, the event processor component must be notified of the occurence. A direct communication could be initiated or a message oriented approach could be used instead. As we were interested in a scalable communication, we followed the second approach.

The BAM Solution proposed in this thesis has simple capabilities of detecting events relevant to the scope of the monitoring and integrate events in contextual business information for executing business rules according to performance indicators and trigger corrective actions. No capabilities of predictive analytics are currently supported.

Several directions in the area of real time analytics are considered for BAM systems for shortening the time between the appearance of an event which requires an action and the time the action is executed. In [47] the authors proove that the business value of an action decreases as the time passes. Action time has three components:

- **Data latency**: time between the occurance of an event until the information is ready for analysis

- **Analysis latency**: time between the information is available for analysis and the tiem the analysis provides valuable output
- **Decision latency**: time between the delivery of information until actions are taken

For reducing the action time, latency must be reduced in all three components [47].

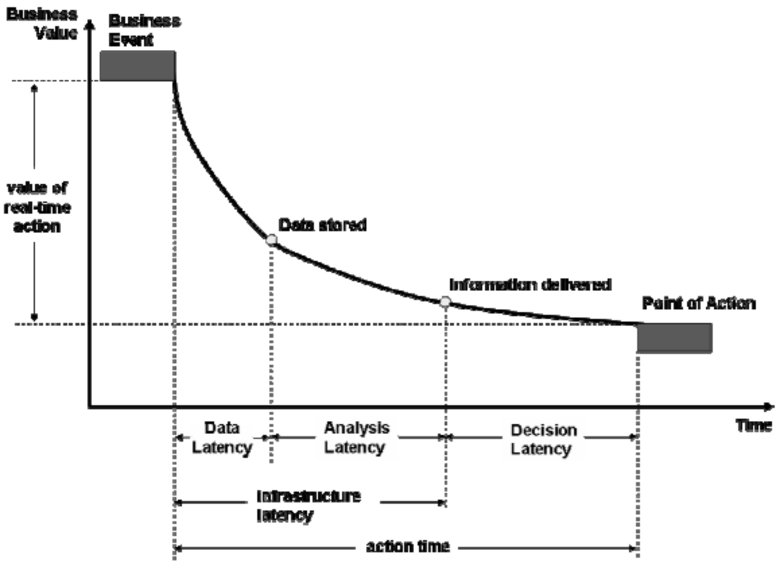


Fig. 2.6.1 Value of reducing action time

In [48] the main steps followed to prepare a BAM system are summarized. A BAM solution must be designed based on a good understanding of the business benefits. It must be integrated with existing Business Intelligence solutions by establishing a data model, building real time event stream data and providing processed information about correlation of specific events.

## 3.8. Process Improve

Business Process Improvement (BPI) is a direction followed in BPM for supporting process enhancements using approaches such as benchmarking, process redesign and process reengineering. The major difference with Business Process Reengineering (BPR) is that BPR is associated with radical changes in a business process.

We plan to enhance our collaborative solution business process based with input coming from end users executing activities in a collaborative business scenario. As we don't target a major refactoring of the collaborative solution, BPR is not in the scope of the current thesis.

In [49] a general methodology for BPI is proposed called Model-based and Integrated Process Improvement (MIPI). MIPI is a seven step iterative procedural approach that guides the actions for process re-engineering and improvements. In the first step an understanding of the business needs must be performed, step which includes establishing targets, developing objectives, developing an organizational model, evaluating current practices and benchmarking the process. Once the business needs for further impovements are identified, the business process must be analyzed and understood. The process performance is measured and the business process is analyzed. Once the analysis is available, the redesign process can be started through technologycal development, making the new process operational, reviewing the change management plan, conducting process deployment.

Several BPI methodologies have been adopted in manufacturing and production environments. One of the most famous is Six Sigma which proposes identifying and eliminating causes of errors for improving the performance, productivity and customer satisfaction. Six Sigma measures and analyzes processes to determine why occure defects and identifies techniques to avoid them as a perspective. Total Quality Management (TQM) is a process improvement technique more focused on satisfying customers through a common effort: teamwork, people, and leadership. Several other theoretical improvement techniques are synthetised and available.

We plan a different approach for the collaborative solution proposed. Organizations are in a deep need of more flexibility and the benefits are noticed in all the areas of an organization. Organizations should be able to adapt easily to continous changes in the business environment or to the internal conditions: resources, market context. Due to this we consider that business processes should not be enforced inside organizations as this would limit the possibilities and the benefits of a custom implementation. Business processes as major resources of the organizations must be approved and accepted by all the people executing activities and contributing to the business development. If the execution is monitored, any deviation from the business process model should be analyzed and the information could be used further for enhancing the business process and ensuring the possible evolution if needed. Statistica information about the repeatedly broken rules could be used for extensions of the process model and adaptation to social factors. As we are designing the business process modelling solution automatically by merging the input process models of the collaborating organizations, enforcement of the process model is a restrictive direction uncompliand with the agile trend.

The literature identifies several other directions for enhancing the business processes for achieving more flexibility. The area is called Adaptive Business Process Management. Such systems are called Adaptive Process Management Systems (APMS) designed on evolving the business processes used inside organizations by using the execution input. APMS support changes at any level in business process definition. Changes recorded at instance level in the execution flow are used further for enhancing areas of process models. Practically APMS permitts learning from the change log information.

Changes in process instances should be handled in a context. An isolated change is a particular situation which could be corrected further and an action could be put in place. A change occuring in several process instances could suggest that the process is difficult to follow in specific areas and must be refactored. A sum of changes in process instances should be followed by a change in the process schema. Once the business process definition is upgraded, the new process instances should be migrated to the new schema.

Several solutions existent in the literature propose using the information from process instances to update process schema definitions. Such a system is ADEPT2 [50] in which process instances are able to trigger updates of the process schema. The updates are composed of insertion of activities, removal of activities, insertion and removal of connections between the activities and replacement of activities. Practically we are using the same set of operations for the evolution of the proposed collaborative solution; the difference is noticed in the source of the update.

ADEPT2 is a complex system with several other capabilities, in which a high level semantics abstract is used. Our goal is to design the collaborative solution in a well known business process representation such as BPMN for which business process engines are available and the process could be easily put in practice. The solution is automatically constructed, all participating organizations are contributing. Due to one major constraint for the current research, a fully automated solution, the business process might be difficut to get accepted inside the organizations participating to the collaboration. Because of this we included learning capabilities in our collaborative solution. ADEPT2 is an abstract solution which needs effort for the setup, with no support for execution and monitoring.

# 4. TECHNICAL OVERVIEW

## 4.1 Architecture

The high level architecture of the service oriented enterprise application able to support the methodology and to compute the collaborative business process based solution is presented in Fig. 4.1. Participant organizations contribute with event log data tracking the activities performed internally to achieve business results in custom business cases. As the context created exceeds the scope of single organizations, same business case is targeted for realization in a multi organizational context. With process mining algorithms the business processes are formalized and stored in Business Process Repositories. Once such a repository is constructed, a check is performed for computing the similarity factor and calculating if the two organizations are compatible from business process perspective. If a good compatibility is calculated, the organizations could easily start collaboration and a collaborative business process based solution is automatically constructed, validated and monitored to ensure the finalization with expected results.

The collaborative business process could be considered a tool for supporting business case development.

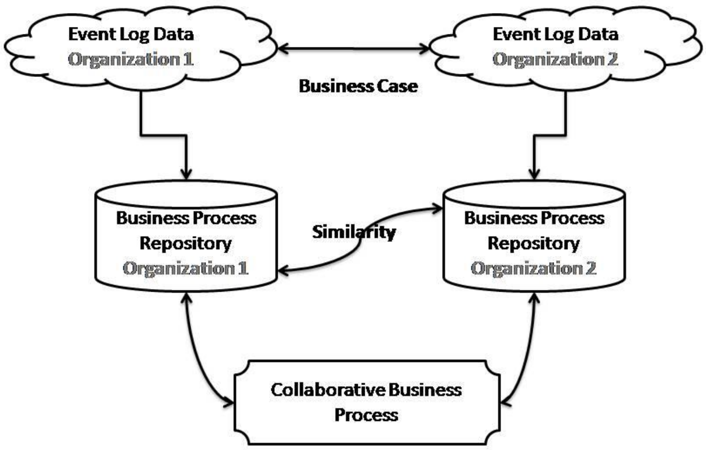


Fig. 4.1 High level architecture

A detailed representation of the components and services composing the enterprise collaborative support application is presented in Fig. 4.2.

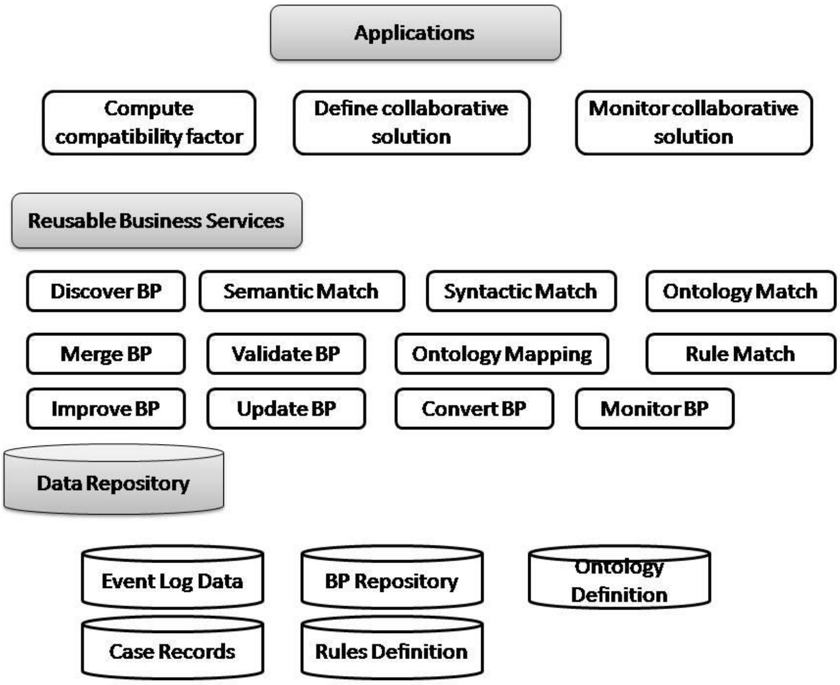


Fig. 4.2 Component based perspective

The Data Repository is composed of:

- **Event Log Data**:
  - repository of events created by organizations when resources are executing the activities which are composing business cases
  - repository of the event log data created by the collaborative business process based solution
- **Business Process Repository**:
  - respository of business processes formalized by applying process discovery algorithms on the event log data created in each organization
  - repository of collaborative business processes able to support business results in a collaboration
- **Ontology Definition**: repository of ontologies extracted for each participant organizations
- **Case Records**: repository of the running cases used for ensuring the operational support and tracking and responding to deviations in real time
- **Rules Definition**: repository of the rules used for ensuring the operational support. The running cases are checked for match with the rules defined.

The Reusable Business Services used for supporting the methodology, composing, monitoring and improving the collaborative solution are the following:

- **Discover BP**
  - Works with the Event Log Data Repository for accesing the traces of the activities executed in implementing a business case

- Operates with process mining algorithms for the extraction of process models supporting a business case
- Once discovered, the business processes are stored in the Business Process Repository

- **Merge BP**
  - Works with the Business Process Repository for accesing the input process models of the participating organizations
  - Performs a merge of the input business process for composing a collaborative process model
  - Once merged, the resulted business process is stored in the Business Process Repository
- **Validate BP**
  - Works with the Business Process Repository for accessing the collaborative process model
  - Works with Event Log Data Repository for accessing the test datasets created for checking if the collaborative business process based solution is compliant with the business objectives defined for the collaboration
  - Operates with process mining algorithms for conformance checking of the resulted process model with test datasets
  - Operates with process mining statistical results for analyzing the behaviour of the resulted collaborative process model
  - Once validated, the resulted business process is stored in the Business Process Repository
- **Update BP**
  - Works with the Business Process Repository for accessing the collaborative process model
  - Updates the business process by inserting and removing nodes and edges
  - Once modified, the resulted business process is stored in the Business Process Repository
- **Convert BP**
  - Works with the Business Process Repository for accesing the input process models of the participating organizations
  - Transforms the business process in an abstract representation – directed attributed graph format following the rules defined for each business process representation
  - Converts the collaborative process model from graph format in a business process representation
  - Works with the Business Process Repository to store the collaborative process model once converted
- **Monitor BP**
  - Works with the Case Record Repository for accesing the traces of the running cases for a business process
  - Composes the events into traces and associates them to cases
  - Works with Rule Match service for verifying the composed traces with the rules defined
- **Syntactic Match**
  - Calculates a syntactic string similarity value for the labels of the activities represented as phrases
  - Several algorithms are implemented: Levenshtein Distance (edit distance), Jaro-Winkler, Jaccard and Sorensen

- **Semantic Match**
  - Calculates a semantic string similarity value for the labels of the activities represented as phrases
  - Algorithms are defined over an external knowledge database: Wordnet
  - Several algorithms are used from Semilar API: Wordnet Lesk Tanim – Greedy and Optimum, Wordnet Lin - Greedy and Optimum, Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), BLEU, Meteor, Pointwise Mutual Information (PMI)
- **Ontology Match**
  - The ontologies considered are stored in the Ontology Definition Repository
  - Performs a match of an activity label in an ontology domain specification
- **Ontology Mapping**
  - Works with the Ontology Definition Repository for accessing the ontologies
  - The ontologies extracted for each input business process are mapped; elements of one ontology are matched to the elements of the other ontology
- **Discover Ontology**
  - Works with the Ontology Definition Repository for storing the ontologies discovered
  - Extracts an ontogy domain out of the activities composing a business process
  - Uses several additional tools: VetIS, s2o converter, OLA – OWL Lite Alignment, Aligment API
- **Rule Match**
  - Works with the Rules Definition Repository for accessing the rules defined
  - Works with the Case Record Repository for accessing the running cases
  - Matches the cases agains the rules defined and triggers an associated action if a rule is broken

The Applications operating with reusable business services used to support the methodology and offer a business process based solution able to support dynamically a collaboration are the following:

- **Compute compatibility factor**
  - Works with Discover BP service for identifying the business process models used internally inside organizations for realizing business cases
  - Once formalized, the business processes are compared and a similarity factor is calculated
  - Works with Syntactic Match, Semantic Match and Ontolology Match for identifying the nodes which can be replaced: nodes from one business process which have a corespondent in the other business process
  - Works with Discover Ontology and Ontology Mapping for extracting formal ontology definitions for the input business processes and mapping the input ontologies (automatic approach) for finding the corespondence
- **Define collaborative solution**
  - Works with Discover BP service for identifying the business process models used internally inside organizations for realizing business cases

  - Works with Merge BP for composing the collaborative process model as a result of merging the input business processes
  - Works with Validate BP for validating the process model with predefined business objectives modelled as test datasets
- **Monitor collaborative solution**
  - Works with Monitor BP service for monitoring actively (triggering actions) the common process model in execution
  - Works with Rule Match service for matching the running cases with predefined monitoring rules
- **Improve collaborative solution**
  - Works with Monitor BP for recording broken rules and extracting patterns with business process areas difficult to follow which will be improved by including the behaviour proposed by the users
  - Works with Update BP for including the changes in the business process definition

## 5.2 Technologies Used

The following external libraries are used for the implementation of the collaborative business process based solution. The implementation is done in Java using Oracle SOA for supporting services.

- **ProM 6.3 API**

ProM [51] is an extensible framework supporting process mining algorithms applied of event log datasets. Process Mining allows extraction of knowledge from process execution logs and gaining different perspectives inside organizations: process, organizational structure, performance. ProM has a graphical interface available which we used for evaluating process mining algorithms on real datasets and choosing the most suitable one in this context. But in the application we used the ProM API for implementations of process discovery and conformance algorithms. ProM has a modularized architecture; additional components could be added in the form of plugins.

- **Semilar API**

SEMILAR [52] is a semantic similarity toolkit which is able to asses the similarity of a source text with a targeted text by calculating the similarity of both input text elements with a benchmark text with a known meaning. Semilar offers implementations for several semantic string similarity algorithms which could be executed at different levels of granularoty: word to word, sentence to sentence, paragraph to paragraph, and document to document or combinations of granularities. In the application developed, we used the implementation of semantic similarity algorithms for computing the compatibility factor able to quantify how similar two or more organizations are from process perspective.

- **OWL2 API**

The OWL API [53] is a Java API for creating and operating with OWL ontologies. OWL API offers support for loading ontologies, saving ontologies, searching between the entities of the ontologies, working with different data types, checking properties. OWL API was used for the management of the ontologies for business processes.

- **OWL Lite Aligment API**

The OWL Lite Aligment API [54] was used for identifying a corespondence on between the ontologies supporting the input business processes. This library has several capabilities, the most important one is the automation; the production of the alignment does not require any user intervention. We also considered another advantage of the implementation, only entities of the same categories are compared. The values of the similarity measures are normalized and normalization is enforced during the full alignment process.

- **JGraphT API**

jGraphT [55] is an open source API offering support for graph representation and visualization. jGraph supports different types of graphs: directed and undirected, weighted, unweighted, labeled, subgraphs.

- **OpenXES API**

OpenXES [56] is an open source implementation of the XES standard for storing and managing event log data. OpenXES is also used in process mining implementations and in several other process analysis and process aware systems. We used OpenXES for managing th evenet log data when ensuring operational support.

# 5. ORGANIZATIONAL COMPATIBILITY FACTOR

## 5.1 One-To-One Model

### 5.1.1Syntactic activity label match

Inter-organizational collaboration between two or more companies is a clear business direction followed by organizations to overcome limitations and to achieve projects objectives when these could not be realized in a closed, single organizational environment. The compatibility information is valuable. Knowing if an organization could be a feasible collaboration partner represents key information. An unfeasible collaboration produces a lot of damage to organizations in the area of costs, customers, business results, portofolio.

A good compatibility from operational perspective certainly does not mean the collaboration will be succesful. A lot of discussions could be started by interpreting the representation of success in a collaborative business relation. But an incompatible partner certainly reducess the success rate. More effort is put in the setup of the operational and the lack of a common understanding of the business process could suggest that the startup of the project is troubled by misunderstandings and differences in opinions concerning activities interpretation. Such issues are not foreseen if organizations use similar business processes for implementing similar business cases.

As standardization in vocabularies and business process representation is a target for many organizations, the similarity factor could be computed just by considering the labels of the activities, the syntactic representation. Standardization is not an easy target, especially in the case of large companies. Standardization requires a good understanding of the use case, products, and customers.

The start point is represented by the Business Process Repository (BPR) we propose calculating how similar the business processes used for specific business cases are, use cases which must be acomplished in collaboration. We extended the algorithm presented in [26] in which the authors propose using a syntactic label match for text similarity. In [26] a specific syntactic algorithm Levenshtein Distance is used. We are evaluating several other syntactic string similarity algorithms to identify the most suitable one for computing the business process similarity as a factor.

For comparing business processes and retrieving a similarity factor, we are using the graph edit distance algorithm. As the graph edit distance is NP-complete, this limitation is overcome with heuristic algorithms as possible and viable alternatives for matching graphs in a reasonable execution time.

The first step is to abstract from business process representation and to transform each input business process in a directed attributed graph.

For the current research, as business processes are not formalized inside organizations we used process mining techniques to extract process models by mining the event log data. Several process mining algorithms could be used to discover models. α-algorithm as the most famous process mining algorithm is not feasible for real life data with inconsistencies, different input sources. We've decided not to use this algorithm but instead to use Heuristic Miner algorithm which is deterministic and produces good results. As business processes are available as

Heuristic Nets which are abstracted as directed attributed graphs.

If $\lambda$ is the set of labels and $\Gamma$ a set of types of nodes, a business process graph could be defined as a tuple (N, E, $\tau$, $\lambda$) in which N is the set of nodes. E is the set of edges, $\tau : N \rightarrow \Gamma$ is a function that maps nodes to types, $\lambda : N \rightarrow \Lambda$ is a function that maps nodes to labels. The resulted directed graph is composed of nodes of different types with specific labels and connected through edges.

The Heuristic Nets as the business process representation are the easiest reduced to graph format. A node in the graph represents one activity executed and the label on the node is the name of the activity executed. An edge represents the switch from one activity to another. A transformation of an Heuristic Net into a directed attributed graph is represented in Fig. 5.1.1.1.

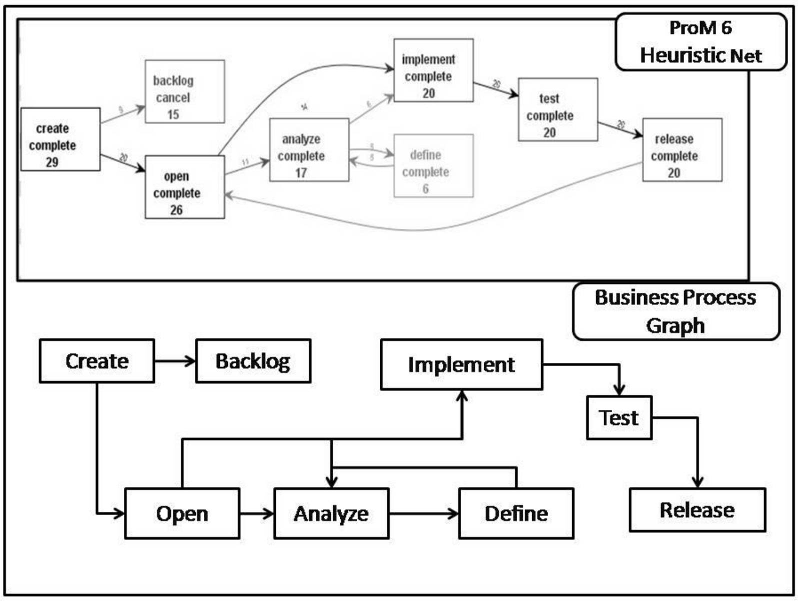


Fig. 5.1.1.1 Transformation of a heuristic net in a directed graph

The technique used to compare business processes is the one presented in [26]. The graph edit distance is the cost of converting one graph to another. A transformation is represented by a set of operations: substitution, node insertion and deletion, edge insertion and deletion. Each operation has a cost associated. We used the default values as costs for each operation: 0.5 for edge insertion and deletion, 0.2 for nodes insertion and deletion and 1 for node substitution. Node substitution is the operation of replacing a node from one graph with a node from the other graph. Node insertion is the operation of adding a node into a graph. Node deletion is the operation of removing a node from a graph. Edge insertion is the operation of adding an edge to a graph. Edge deletion is the operation of removing an edge from a graph.

The similarity of two nodes is represented by the similarity of the node labels. As we are currently using syntactic string similarity, overlaps of strings suggests a similarity between node labels. Syntactic string similarity algorithms are implemented at word level. We used an extension for identifying the similarity of two phrases. The algorithm used is **bipartite graph matching**. The phrase is split into words represented as tokens and a similarity in between the token list of both nodes is computed.

The procedure is also known as the optimal assignment problem. Given a weighted complete bipartite graph, G = (X×Y, E), where an edge [x, y] has a weight w(x,y), a matching from X to Y with maximum value is searched. X is the set of the first list of tokens and Y is the set of the second list of tokens. A matching is composed out of the elements of X and Y so that the total weight is maximum. The edge connecting X nodes to Y nodes is represented by the similarity between the nodes.

For the current research, we evaluated several syntactic string similarity algorithms: Levenshtein Distance (edit distance), Jaro-Winkler, Jaccard, Sorensen. Our results are captured in [74].

Levenshtein Distance (edit distance) is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other. Zero means the strings are equal and the length of the maximum string is the largest value possible. The greater the edit distance is, the more different the input strings are.

We are considering extending the edit distance with a weighting system for quantifying the position of the letters composing the string. Also beggining of strings should have a higher weight as the end of the strings.

Once the label edit distance is computed for all pairs of possible labels, the similarity of nodes is calculated with the method presented in [26]. If we have the following graphs G1 = (N1, E1, τ1, λ1) and G2 = (N2, E2, τ2, λ2) where and n1 € N1 and n2 € N2 are two nodes. The similarity of n1 and n2 is (5.1.1):

$$\text{Sim}(n1, n2) = \left\{1 - \frac{LevensteinDis\tan ce(\lambda 1(n1), \lambda 2(n2))}{\max(\lambda 1(n1), \lambda 2(n2))}\right\} \qquad (5.1.1)$$

Another algorithm considered for syntactic string match is Jaro-Winkler based on the number of characters and transpositions. The similarity of given strings s1 and s2 is calculated with the following formula (5.1.2):

$$\text{Sim}(s1, s2) = \frac{1}{3}\left(\frac{m}{s1} + \frac{m}{s2} + \frac{m-t}{m}\right) \qquad (5.1.2)$$

In the formula m is the number of matching characters and t is half the number of transpositions.

Jaccard syntactic similarity algorithm is also evaluated. In this algorithm strings are interpreted as set of characters and the similarity index is the intersection between characters split with the total number of characters. S1 is the set of characters of the first string and s2 is the set of characters of the second string. The formula is presented in (5.1.3).

$$\text{Sim}(s1,s2) = \frac{s1 \cap s2}{s1 \cup s2} \qquad (5.1.3)$$

Sorensen algorithm is evaluated as well (5.1.4). It is also based on the set of characters representation of the strings. This coefficient is not very different from the Jaccard index.

$$\text{Sim(s1,s2)} = \frac{2|s1 \cap s2|}{|s1| + |s2|} \qquad (5.1.4)$$

We evaluated the syntactic string similarity algorithms on four business processes representing the same business case – change management [74]. We concluded that Jaccard and Sorensen syntactic similarity algorithms are not feasible to be used in this business process match context as they don't take into consideration the order of the letters. For comparing business processes of the same domain area, WordLevent algorithm gives good results. In a standardized environment we will use the WordLevent algorithm for comparing the labels of the business processes and finding equivalences.

The following business processes are evaluated Fig. 5.1.1.2, Fig. 5.1.1.3, Fig. 5.1.1.4, Fig. 5.1.1.5:

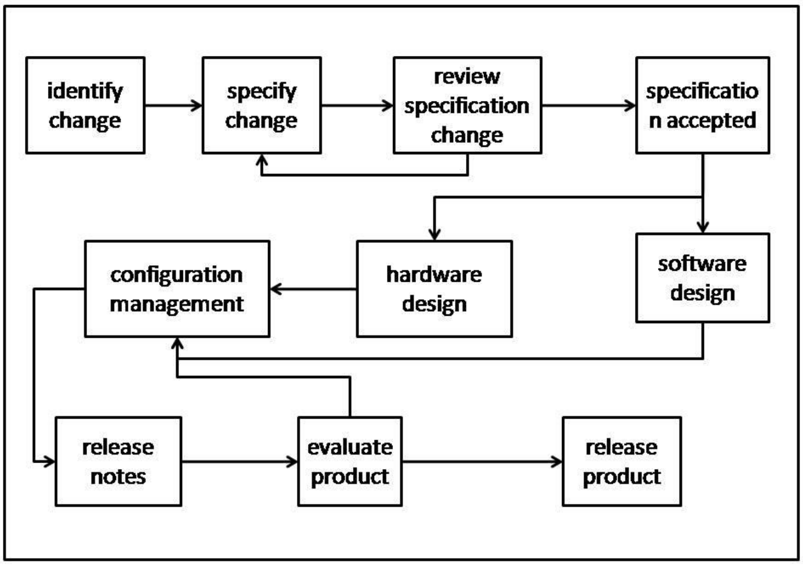


Fig. 5.1.1.2 Business process 1

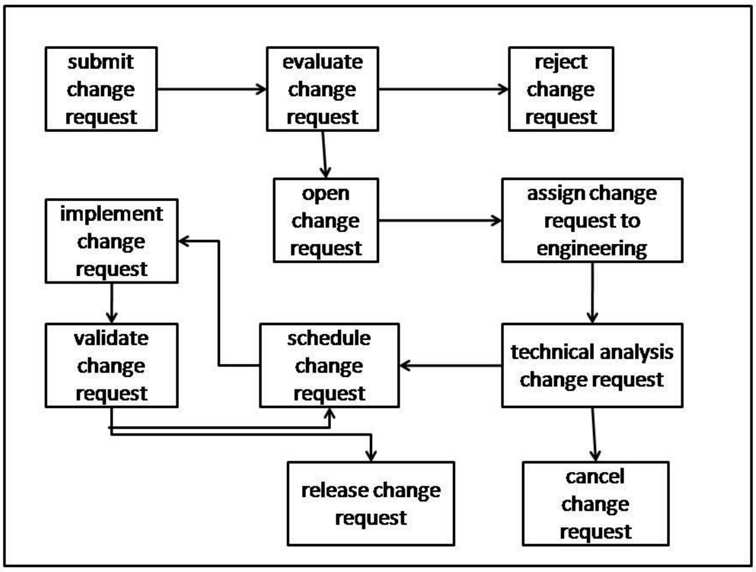


Fig. 5.1.1.3 Business process 2

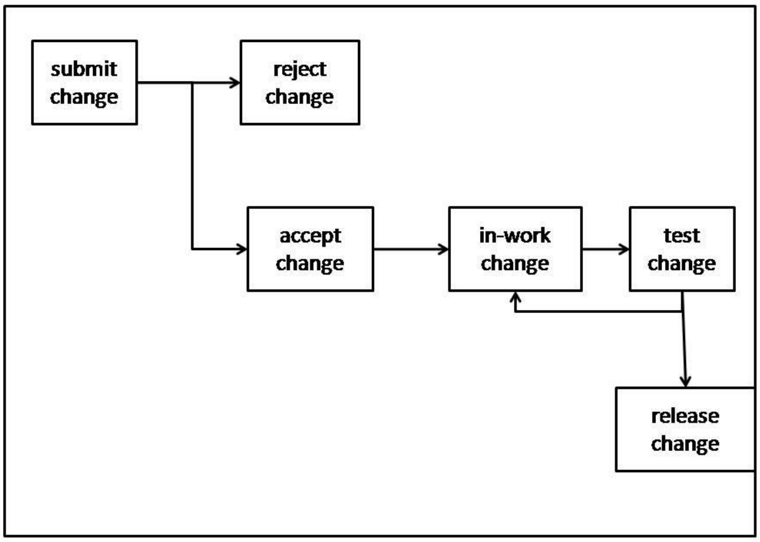


Fig. 5.1.1.4 Business process 3

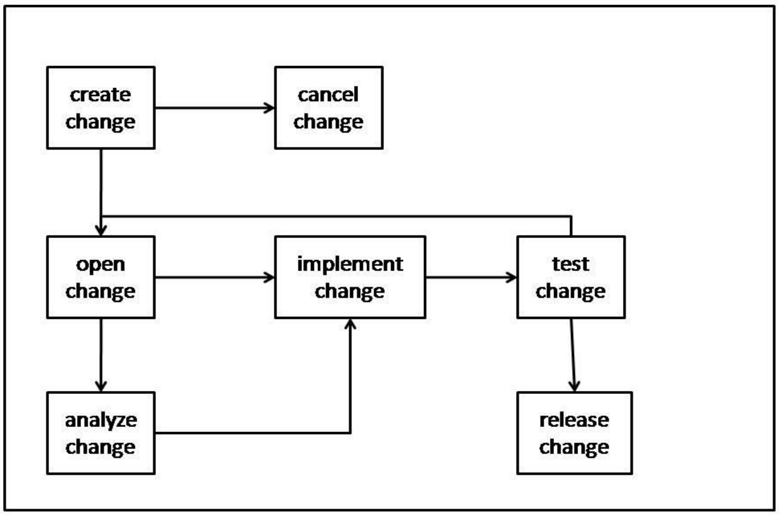


Fig. 5.1.1.5 Business process 4

In Table 5.1.1 the results of the syntactic similarity algorithms used for label match could be found:

| LABEL MATCH ALGORITHM | BUSINESS PROCESSES SIMILARITY | | | | | |
|---|---|---|---|---|---|---|
| | [1, 2] | [1, 3] | [1, 4] | [2, 3] | [2, 4] | [3,4] |
| **WordLevent** | -0.27 | -0.14 | -0.05 | 0.01 | 0.08 | **0.59** |
| **Jaro-Winkler** | -0.81 | -0.46 | -0.47 | -0.68 | 0.25 | -0.01 |
| **Jaccard** | -0.91 | -0.66 | -0.41 | -0.62 | **0.47** | -0.02 |
| **Sorensen** | -0.76 | -0.64 | -0.49 | -0.59 | -0.43 | 0.03 |

Table 5.1.1 Business process similarity with syntactic activity label match

The similarity values are interpreted. The interpretation results are synchronized with manual analysis results. An interpretation could be performed by a business analyst due to the small size of the business processes:

- **Business Process 1 (BP1)** uses a different approach than all business processes. It includes a step in advance before submitting, the identification of the change. Also the technical side of implementing a change is missing. This process model considers configuration management, different than the rest of business processes.
- **Business Process 2 (BP2)** has a low similarity with Business Process 3 (BP3) and Business Process 4 (BP4) because of the granularity level in business process representation. BP2 has a lower granularity representation; the business process is described in more details than BP3 and BP4.
- The highest similarity is found between **Business Process 3 (BP3)** and **Business Process 4 (BP4),** the technical implementation of a change is described in more details in BP4 than in BP3.

As BP1, BP2, BP3 and BP4 represent business processes for the same business case in the same business area a syntactic string similarity is a feasible approach for considering the labels in the comparison of the business processes and calculating the compatibility between organizations from an operational perspective.

### *5.1.2 Semantic activity label match*

Semantic similarity extends the functionality offered by syntactic similarity as it evaluates two strings by considering their meaning and not syntactical representation. All semantic string similarity algorithms operate with a knowledge database by using a benchmark text with known meaning.

We extended the activity label match from syntactical interpretation to a semantic context with the purpose of allowing business process comparison in different domain areas.

For evaluating semantic string similarity algorithms, we used SEMILAR, a java API offering diverse implementation for the semantic similarity algorithms available. All SEMILAR API is based on Wordnet database and the following algorithms are available: Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), BLEU, Meteor, Pointwise Mutual Information (PMI), Dependency based methods, optimized methods based on Quadratic Assignment etc. All similarity algorithms could be used with different granularities: word to word, sentence to sentence.

For this thesis the following algorithms are evaluated: Wordnet Lesk Tanim - Greedy and Optimum, Wordnet Lin - Greedy and Optimum, LSA, LDA.

Wordnet Lesk Tanim is a semantic similarity algorithm based on the Wordnet database. The algorithm states that the relatedness of two words is proportional to the extent of overlaps of their dictionary definitions. In advance, each phrase is prepared by performing normalization, a stemming of the words. The greedy version of algorithm is constructed by ignoring the history. Each word is checked with Wordnet database. The optimal algorithm takes into account the current meaning of the phrase. Due to the high complexity such an algorithm cannot give results in a feasible execution time. Because of this disadvantage, the algorithm is combined with heuristic techniques to eliminate the not possible definitions match.

Another algorithm using the Wordnet database evaluated for the current research is Wordnet Lin. Wordnet Lin algorithm computes the semantic relatedness of word meanings using the information content of the concepts in Wordnet and the similarity theorem: the more differences between the input words are noticed, the less similar the words are. Similarity between words is measured by the ratio between the amount of information needed to state the commonality of the words and the information needed to fully describe what words represent.

Besides the algorithms based on Wordnet database, some other types of semantic algorithms are evaluated. Latent Semantic Analysis (LSA) is a technique used in natural language processing. The algorithm is based on extracting and representing the meaning of words by statistical computations applied to a large corpus of text. The method compares the meanings or concepts behind the words. LSA maps phrases in a concept space and performs the comparison in this space.

Latent Dirichlet Allocation (LDA) extracts similarity of phrases based on topics the sentences contain. The database contains a mixture of topics using words with specific probabilities. The similarity is given by the proportion of words belonging to a topic.

| LABEL MATCH ALGORITHM | BUSINESS PROCESSES SIMILARITY | | | | | |
|---|---|---|---|---|---|---|
| | [1, 2] | [1, 3] | [1, 4] | [2, 3] | [2, 4] | [3,4] |
| **Wordnet Lesk Tanim Greedy** | -0.08 | **0.20** | 0.12 | 0.07 | 0.17 | 0.50 |
| **Wordnet Lesk Tanim Optimum** | -0.26 | -0.06 | 0.18 | 0.06 | 0.19 | 0.62 |
| **Wordnet Lin Greedy** | -0.13 | -0.02 | 0.03 | 0.23 | 0.30 | 0.56 |
| **Wordnet Lin Optimum** | -0.29 | -0.15 | 0.01 | 0.18 | 0.25 | **0.64** |
| **LSA** | N/A | N/A | N/A | N/A | N/A | N/A |
| **LDA** | N/A | N/A | N/A | N/A | N/A | N/A |

Table 5.1.2 Business process similarity with semantic activity label match

Semantic similarity is the only viable solution when business processes don't use the same vocabulary for activity labels representation. Even if semantic similarity brings flexibility, the major disadvantage is in the area of resources and execution time. The execution time of semantic similarity algorithms is ~450 times larger than the execution time of WordLevent syntactic. The differences are more relevant in the area of memory consumption. LDS, LSA need more than 6G memory, the algorithm execution results were not available due to the high resource consumption.

The greedy version available of the algorithms reduces the search space. The execution time is higher for Optimum version but the quality of the results is improved. Greedy version of algorithms could lead to incorrect results in exceptional scenarios in which specific semantic search directions are not considered.

Concerning the business processes presented in Fig. 5.1.1.2, Fig. 5.1.1.3, Fig. 5.1.1.4 and Fig. 5.1.1.5, the Greedy version of the Wordnet Lesk algorithm detects a similarity between BP1 and BP3. Semantic algorithms are able to detect similarity between BP1 and BP4 as activities are placed and compared in a context. None of the syntactic algorithms is able to achieve that.

### *5.1.3 Ontology mapping activity label match*

Due to the high resource consumption for considering the semantics in comparing business processes, we've performed a limitation of the semantic search space to the ontologies behind the process models.

Ontologies are popular nowadays in many domain areas exceeding the scope of artificial intelligence, including them allows the separation of domain knowledge from operational knowledge. Domain ontology permits the specification of the vocabulary used to represent a system and it could be constructed on a top ontology. We proposed comparing business processes each one defined in an own ontology by considering the relation between the elements of the ontologies.

Similarity is calculated after extracting business process definitions and associating them to business cases. For each input business process, the process ontology is identified listing the semantics of the area. Ontologies are formalized and a mapping is performed to associate elements with the same meaning.

Ontology mapping follows several directions possible. One, the most common is that a similarity of labels means that entities could be associated and a mapping is valid. This approach is not improving the current semantic direction used for comparing the labels of the business processes. In our solution for mapping the ontologies supporting the process models we are using semantic nets which assumes that if the properties of the two concepts are equal, the concepts are also equal. An extension could be considered further with description logics. Mapping considering description logics is defined over concepts. Concepts are hierarchically organized; a similarity between concepts is calculated if a similarity between super concepts is found. If concepts have similar siblings, the concepts are similar. Several other possible directions for a suitable mapping between ontologies are possible. Our goal was not to extend the techniques for ontology mapping but to use the algorithms for identifying the correspondence between the elements of the business processes.

One important target for the presented solution is automation, due to this some intermediate steps are performed to formalize ontologies and to ensure a correct mapping in between. We are extracting the Semantics of the Business Vocabulary and Business Rules (SBVR) of both business processes subject of the comparison. Once the SBVR representations are available for each business process, the SBVR business vocabularies are represented in Web Ontology Language (OWL2). Each business process will have associated ontology and the comparison is performed in the ontological space. The ontologies extracted are mapped and for each activity of the business process the corresponding activity (if found) is identified in the partner business process, subject of a comparison.

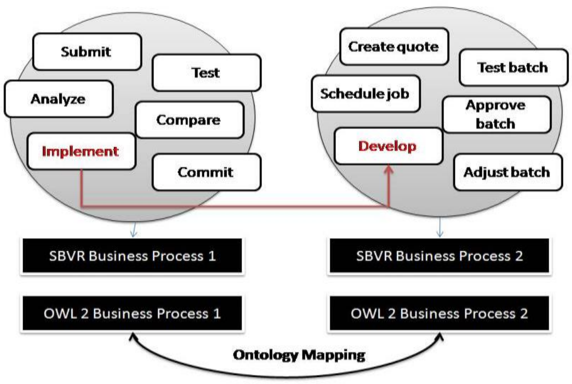


Fig. 5.1.3.1 Compute similarity using ontologies

In Fig. 5.1.3.1 the activity label match is graphically represented by considering the ontological space. A business process is composed of activities and an identification of the activities in the first business process which have a correspondent in the second business process is performed. This information is used in the business process similarity algorithm, instead of using the syntactical representation of the labels or performing the search in a fully semantically space, the search is limited to a domain composed of the mapping of the ontologies. We've presented this direction in [75]. For computing the similarity of nodes to identify node substitution operations, the syntactic string similarity algorithm is replaced with a search in a semantic space represented by the ontologies of the source input business processes.

The first step in comparing business processes by using the ontologies extracted and integrated for each business process is the extraction of business vocabularies from process models.

According to [57] business rules could be considered as part of the Business Rules Management (BRM) discipline. As stated in [57] a business rule is a logical statement that defines or constraints aspects of the business in the concrete situation. A Business Vocabulary contains a set of terms specialized and concepts that a given organization uses in performing business. In [58] an approach to integrate a business process with a business vocabulary is presented by mapping certain elements of the business process meta-model with elements of the business vocabulary meta model. For example, the BPMN activity could be mapped to SBVR noun concepts, fact type and rules associated with that activity. Certain elements of the business process might not be mapped to business vocabulary elements.

As a simple example, the activity "Submit Change" for the change management business process could be mapped into a noun concept "change" and a fact "support users register a change request". SBVR is able to formalize the definition of a business vocabulary particular to a business process.

We used for this purpose the Magic Draw UML tool extended with BP→BV (business process to business vocabulary) and VeTIS tool [58] which could be used either as a standalone or as a plugin.

The following business processes represented in Fig. 5.1.3.2 and Fig. 5.1.3.3 are used for comparison:

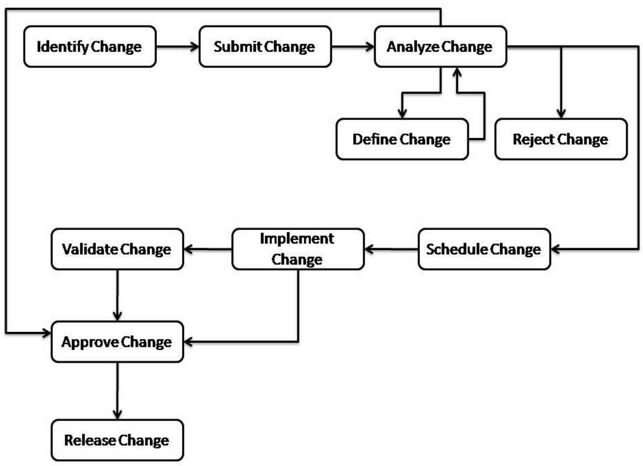


Fig. 5.1.3.2 Business process 1

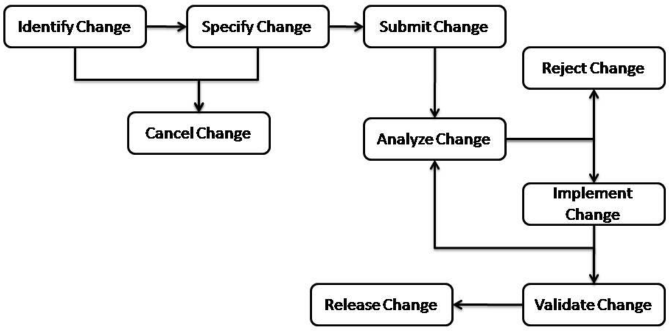


Fig. 5.1.3.3 Business process 2

The vocabulary is extracted for each business process. A fragment of the vocabulary for Business Process 1:

Change_management
General_concept: vocabulary
Language: "en"
Namespace_URI: "http://cs.upt.ro/Change_management"
change

change contains change
Concept_type: partitive_verb_concept
Concept_type: transitive_verb_concept

change is_replaceable_by change
Concept_type: purely_reflexive_verb_concept

replaceable_change
Definition: change that is_replaceable_by change
Concept_type: verb_concept_role
General_concept: change
replacing_change
Definition: change that is_replacing_change_for change
Concept_type: verb_concept_role
General_concept: change

informationrequest_change
General_concept: change
changerequest_change
General_concept: change
featurerequest_change
General_concept: change

problemreport_change
General_concept: change

informationrequest_change has description
description is_detailed
concept 'informationrequest_change' incorporates characte-ristic 'informationrequest_change is_detailed'

…
owner
Definition: owner is the current responsible and ac-countable of the change
engineer
General_concept: owner
businessanalyst
General_concept: owner
projectmanager
General_concept: owner

owner identify change
Synonymous_form: change is_identify owner
change is_identify owner
See: owner identify change
Concept_type: inverse_verb_concept

The business rules for Business Process 1 are identified and a fragment is presented below:

It is necessary that problemreport_change is_released_by exactly 1 projectmanager
It is necessary that problemreport_change is_rejected_by exactly 1 projectmanager
It is necessary that problemreport_change is_approved_by exactly 1 projectmanager
It is necessary that problemreport_change is_validated_by at_least 1 engineer
It is necessary that problemreport_change is_analyzed_by at_least 1 engineer

A fragment of the vocabulary for Business Process 2 is listed below:

Change_management
General_concept: vocabulary
Language: "en"
Namespace_URI: "http://cs.upt.ro/Change_management"
change

featurerequest_change
General_concept: change
problemreport_change
General_concept: change

responsible
Definition: responsible is the person in charge of the change
support_user
General_concept: owner

IT_user
General_concept: owner
PM_user
General_concept: owner

owner identify change
Synonymous_form: change is_identify owner
change is_identify owner
See: owner identify change
Concept_type: inverse_verb_concept

owner analyze change
Synonymous_form: change is_analyzed_by owner
change is_analyzed_by owner
See: owner analyze change
Concept_type: inverse_verb_concept

And a fragment of the rules for Business Process 2 is extracted:

It is necessary that problemreport_change is_released_by exactly 1 PM_user
It is necessary that problemreport_change is_rejected_by exactly 1 PM_user
It is necessary that problemreport_change is_validated_by at_least 1 IT_user
It is necessary that problemreport_change is_analyzed_by at_least 1 IT_user
It is necessary that problemreport_change is_implemented_by at_least 1 IT_user
It is necessary that problemreport_change is_submitted_by exactly 1 support_user

As the vocabulary and the rules are formalized, the next step is to extract the OWL2 representation of the business process ontology. A tool is used for this purpose, the s2o converter. The start point is the SBVR vocabulary and rules and the output is the OWL2 representation of the business process ontology. The technique is presented in [59] and a short fragment of the OWL representation is listed below:

```
Prefix( xsd:=<http://www.w3.org/2001/XMLSchema#> )
Prefix( ns:=<http://cs.upt.ro/> )
Ontology( <http://cs.upt.ro/Change_management>
Declaration( AnnotationProperty( <ns:Change_management#label_sbvr> ) )
Declaration( AnnotationProperty( <ns:Change_management#label_en> ) )
Declaration( Class( <ns:Change_management#role> ) )
AnnotationAssertion(
<ns:Change_management#label_sbvr><ns:Change_management#role> "role"@en
)
AnnotationAssertion( <http://www.w3.org/2000/01/rdf-
schema#label><ns:Change_management#role> "role"@en )
AnnotationAssertion(
<ns:Change_management#label_en><ns:Change_management#role> "role" )
Declaration( ObjectProperty( <ns:Change_management#partitive_object_property>
) )
AnnotationAssertion(
<ns:Change_management#label_sbvr><ns:Change_management#partitive_object
_property> "partitive_object_property"@en )
```

AnnotationAssertion( <http://www.w3.org/2000/01/rdf-schema#label><ns:Change_management#partitive_object_property> "partitive object property"@en )

Prefix( xsd:=<http://www.w3.org/2001/XMLSchema#> )
Prefix( ns:=<http://cs.upt.ro/> )
Ontology( <http://cs.upt.ro/Change_management>
Declaration( AnnotationProperty( <ns:Change_management#label_sbvr> ) )
Declaration( AnnotationProperty( <ns:Change_management#label_en> ) )
Declaration( Class( <ns:Change_management#change> ) )
AnnotationAssertion( <ns:Change_management#label_sbvr><ns:Change_management#change> "change"@en )
AnnotationAssertion( <http://www.w3.org/2000/01/rdf-schema#label><ns:Change_management#change> "change"@en )
AnnotationAssertion( <ns:Change_management#label_en><ns:Change_management#change> "change" )
Declaration( Class( <ns:Change_management#featurerequest_change> ) )
AnnotationAssertion( <ns:Change_management#label_sbvr><ns:Change_management#featurerequest_change> "featurerequest_change"@en )
AnnotationAssertion( <http://www.w3.org/2000/01/rdf-schema#label><ns:Change_management#featurerequest_change> "featurerequest change"@en )
AnnotationAssertion( <ns:Change_management#label_en><ns:Change_management#featurerequest_change> "featurerequest change" )
Declaration( Class( <ns:Change_management#problemreport_change> ) )

The OWL 2 representation of Business process 1 is graphically summarized in Fig. 5.1.3.4:

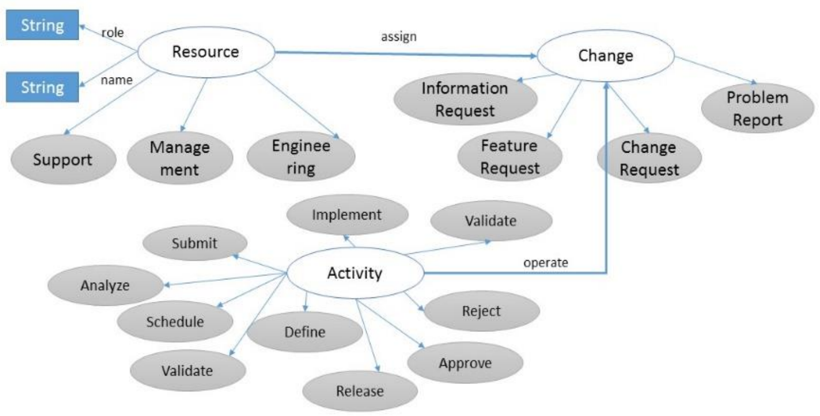


Fig. 5.1.3.4 Ontology domain for business process 1

The OWL 2 representation of business process 2 is in Fig. 5.1.3.4:

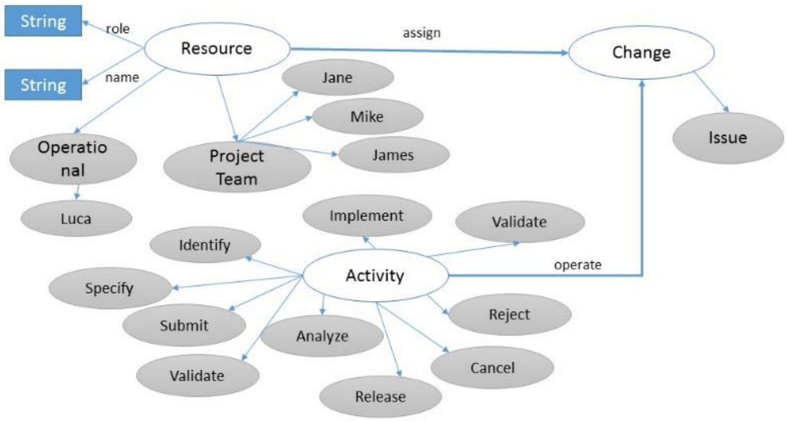


Fig. 5.1.3.5 Ontology domain for business process 2

Once the correspondence between the ontologies is calculated, the next step includes performing a mapping and interconnecting elements of the two ontologies. With this action the search space is reduced significantly, the activities are matched only in this space instead of a full semantic space.

An utility is used for this purpose OLA tool – OWL Lite Alignment described in [60] which ensures the automation. The production of the alignment does not require any user intervention. Another aspect considered was the comparison presumption, only entities of the same categories are compared. The same similarity function is used for comparing. The values of the similarity measures are normalized and normalization is enforced during the full alignment process. Another aspect analyzed when using the OLA tool was the implementation in Java and the availability of an OWL Java API for parsing and merging OWL files. This is a clear advantage as the collaborative business process modelling solution is designed in Java by using jGraph for the representation and visualization of graphs. A fragment of the mapping performed to limit the search space is listed below:

```
<?xml version='1.0' encoding='utf-8' standalone='no'?>
<rdf:RDF
xmlns='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'
xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
xmlns:xsd='http://www.w3.org/2001/XMLSchema#'
xmlns:align='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'>
<Alignment>
<xml>yes</xml>
<level>0</level>
<type>**</type>
<me-thod>ca.uqam.info.latece.sboa.impl.algorithms.OLAlignment</method>
```

```
<time>3669</time>
<onto1>
<Ontology
rdf:about="https://www.dropbox.com/s/e0zj31nboj16axj/ontologyBP1.owl?dl=0">
<location>https://www.dropbox.com/s/e0zj31nboj16axj/ontologyBP1.owl?dl=0</lo
cation>
<formalism>
<Formalism align:name="OWL1.0" align:uri="http://www.w3.org/2002/07/owl#"/>
</formalism>
</Ontology>
</onto1>
<onto2>
<Ontology
rdf:about="https://www.dropbox.com/s/ajwc480afszd48i/ontologyBP2.owl?dl=0">
<loca-
tion>https://www.dropbox.com/s/ajwc480afszd48i/ontologyBP2.owl?dl=0</location
>
<formalism>
<Formalism align:name="OWL1.0" align:uri="http://www.w3.org/2002/07/owl#"/>
</formalism>
</Ontology>
</onto2>
<map>
<Cell>
<entity1
rdf:resource='https://www.dropbox.com/s/e0zj31nboj16axj/ontologyBP1.owl?dl=0
#identify_change'/>
<entity2
rdf:resource='https://www.dropbox.com/s/ajwc480afszd48i/ontologyBP2.owl?dl=0
#identify_change'/>
<relation>=</relation>
<measure
rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.00000000000000000
</measure>
</Cell>
</map>
<map>
<Cell>
<entity1
rdf:resource='https://www.dropbox.com/s/e0zj31nboj16axj/ontologyBP1.owl?dl=0
#identify_change'/>
<entity2
rdf:resource='https://www.dropbox.com/s/ajwc480afszd48i/ontologyBP2.owl?dl=0
#specify_change'/>
<relation>=</relation>
<measure
rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.48334733557983356
</measure>
</Cell>
</map>
<map>
```

<Cell>
<entity1
rdf:resource='https://www.dropbox.com/s/e0zj31nboj16axj/ontologyBP1.owl?dl=0
#reject_change'/>
<entity2
rdf:resource='https://www.dropbox.com/s/ajwc480afszd48i/ontologyBP2.owl?dl=0
#reject_change'/>
<relation>=</relation>
<measure
rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.00000000000000000
</measure>
</Cell>
</map>
<map>
<Cell>
<entity1
rdf:resource='https://www.dropbox.com/s/e0zj31nboj16axj/ontologyBP1.owl?dl=0
#analyze_change'/>
<entity2
rdf:resource='https://www.dropbox.com/s/ajwc480afszd48i/ontologyBP2.owl?dl=0
#analyze_change'/>
<relation>=</relation>
<measure
rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.00000000000000000
</measure>
</Cell>
</map>

And graphically displayed in Fig. 5.1.3.5.

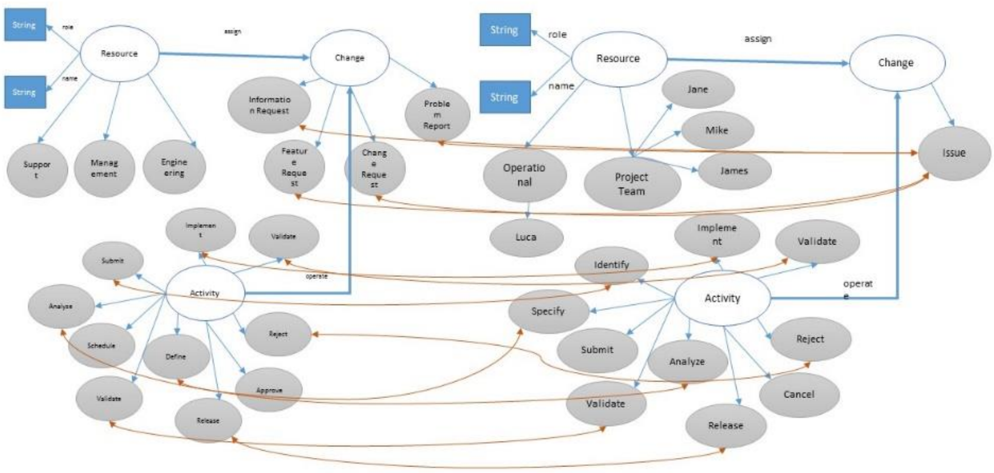


Fig. 5.1.3.6 Ontology mapping between business process 1 and 2

Besides the OWL-Lite approach for ontology reconciliation, we considered other implementations such as the Alignment API 4.0 presented in [67]. Alignment API is a Java API able to manipulate the alignments between ontologies by using predefined matchers. As described in [67] the Alignment API is based on four main components:

- **OntologyNetwork**: a container for handling input ontolgies and alignments
- **Alignment**: composed of a set of Cells and metadata about the alignment
- **Cell**: represents a correspondence, it relates two entities in a Relation
- **Relation**: represents the relation between two entities

Alignment API also offers a processing structure used for manipulating and consuming alignments. It is composed of:

- **AlignmentProcess**: interface for all matchers, uses an Alignment and a Property object as input providing the parameters for the matcher
- **Evaluator**: compares two or more alignments represented as Alignment objects
- **AlignmentVisitor**: interface for displaying alignments in different formats

Once the ontology mapping is available, in the next step the similarity factor can be calculated using the correspondence of the elements in the mapping. In the first step the similarity matrix is constructed. For each of the labels of the first business process, a search in the ontological space is performed, targeting the identification of a correspondent label in the second business process.

Alignment API [67] is widely used as a standalone and portable library for sharing alignments. More than 30 tools existent on the market are using the library as it offers a wide variety of implementations for reconciling ontologies. The API is also embedded in several toolkits and servers and it offers functionality through a variety of means including web services interfaces. Because of the extensive usage and the existence of a Java API, we've also consider it for aligning the ontologies supporting the input business processes for the collaborating organizations.

Below a fragment of ontology alignment is presented:

Alignment metadata

onto1
uri: http://apache/tutorial/myOnto.owl
file: file:///Java/alignapi/html/tutorial/myOnto.owl
type: OWL2.0
onto2
uri: http://apache/tutorial/otherOnto
file: file:///Java/alignapi/html/tutorial/otherOnto
type: OWL2.0
level 0
type **
http://knowledgeweb.semanticweb.org/heterogeneity/alignment : time 322
http://knowledgeweb.semanticweb.org/heterogeneity/alignment : method
fr.inrialpes.exmo.align.impl.method.StringDistAlignment

Correspondences

http://apache/tutorial/myOnto.owl#annote = 0.7142857142857143
http://apache/tutorial/otherOnto#hasNote

http://apache/tutorial/myOnto.owl#publisher = 0.75
 http://apache/tutorial/otherOnto#hasPublisher
http://apache/tutorial/myOnto.owl#PhdThesis = 1.0
 http://apache/tutorial/otherOnto#Phdthesis
http://apache/tutorial/myOnto.owl#Entry = 1.0
 http://apache/tutorial/otherOnto#Entry
http://apache/tutorial/myOnto.owl#mrNumber = 0.7272727272727273
 http://apache/tutorial/otherOnto#hasMrnumber
http://apache/tutorial/myOnto.owl#month = 0.625
 http://apache/tutorial/otherOnto#hasMonth
http://apache/tutorial/myOnto.owl#contents = 0.7272727272727273
 http://apache/tutorial/otherOnto#hasContents
http://apache/tutorial/myOnto.owl#type = 0.5714285714285714
 http://apache/tutorial/otherOnto#hasType
http://apache/tutorial/myOnto.owl#Person = 0.30000000000000004
 http://apache/tutorial/otherOnto#Techreport
http://apache/tutorial/myOnto.owl#key = 0.5
 http://apache/tutorial/otherOnto#hasKey
http://apache/tutorial/myOnto.owl#humanCreator = 1.0
 http://apache/tutorial/otherOnto#humanCreator
http://apache/tutorial/myOnto.owl#chapter = 0.7
 http://apache/tutorial/otherOnto#hasChapter
http://apache/tutorial/myOnto.owl#Title = 0.5714285714285714
 http://apache/tutorial/otherOnto#Article
http://apache/tutorial/myOnto.owl#Unpublished = 1.0
 http://apache/tutorial/otherOnto#Unpublished
http://apache/tutorial/myOnto.owl#Journal = 0.30000000000000004
 http://apache/tutorial/otherOnto#Conference
http://apache/tutorial/myOnto.owl#language = 0.7272727272727273
 http://apache/tutorial/otherOnto#hasLanguage
http://apache/tutorial/myOnto.owl#year = 0.5714285714285714
 http://apache/tutorial/otherOnto#hasYear

```
for(String nodeA : _leftGraphActivities)
  for (String nodeB : _rightGraphActivities)
    similarytyMatrix[nodeA,nodeB] = getHighestMapping(nodeA, nodeB);
```

Method getHighestMapping uses the ontologies extracted for both input business processes and retrieves if case the measure objects. Based on the similarity calculated, the mapping of the nodes from one business process to another is identified.

Mapping [identify change->define change]
Mapping [implement change->implement change]
Mapping [analyze change->analyze change]
Mapping [cancel change->cancel change]
Mapping [specify change->submit change]
Mapping [validate change->release change]
Mapping [reject change->release change]
Mapping [submit change->submit change]
Mapping [release change->release change]

The computed sets of added and deleted nodes and added and deleted edges:

Added nodes [define change]
Deleted nodes []
Added edges [(validate change:analyze change), (identify change:cancel change), (validate change:cancel change), (identify change:specify change), (sub-mit change:cancel change), (validate change:release change), (implement change:validate change), (specify change:submit change), (submit change:analyze change), (analyze change:reject change), (submit change:implement change)]
Deleted edges [(validate change:analyze change), (identify change:cancel change), (validate change:cancel change), (identify change:specify change), (sub-mit change: cancel change), (validate change:release change), (specify change:submit change), (submit change:analyze change), (analyze change:reject change), (submit change:implement change)]

After calculating the fractions of added/deleted nodes and added/deleted edges, the organizational compatibility factor calculated is:

**Sim[BP1, BP2] = 0.60**.

The current proposal for including ontologies in calculating the business process similarity is fully automated. With intermediate steps, the ontologies for all participating organizations are automatically extracted and a mapping is performed in the semantical space represented by the mapping of the ontologies for all business processes. One of the goals we've considered when proposing the business process based collaborative solution is an automatic execution flow. The major limitation for this ontology mapping is represented by custom situations in which the specific relations between elements should be manually identified.

We extended this similarity calculation with a manual action performed; the correspondence between the elements of the ontologies identified. Another extension possible foreseen would be to consider a single ontology as the base for the business process comparison and merge for computing the collaborative solution. Assuming that only a set of ontology elements should be considered, this could be accomplished by manually defining the ontology domain as the base for the collaborative solution and manually annotating business process activities by referencing the elements of the ontology domain predefined. The search uses the annotations instead of the mapping between two ontology domains.

## 5.2 One-To-Many Model

In 5.1 we presented the business process similarity algorithm by considering a one-to-one correspondence in between the nodes of the business processes. We extended the algorithm to multivalent node matching. The goal is to compute the similarity for business processes with different granularity representation. Our results are captured in [76].

One of the nodes in the first graph is matched with a set of nodes composing a sub-process in the second graph. Such a relation between one node of a business process with a sub-process of another business process could be identified either in a pre-processing phase before the similarity algorithm is applied.

One of the goals we've had in mind for the current solution is that it should be fully automated. An automatic match could be also performed if a semantical context is used. A match between one activity of the first graph to a set of activities in the second graph could be possible if a semantical relation between the activity labels, hypernym, is used. Hypernym as a lexical relation is able to match words to contexts, in a broader sense.

The one-to-many similarity is exemplified on the business processes in Fig. 5.2.1. The first activity "Receive Service Request" is the same in both input business processes. For the next activity the second business process is using more detailed information, "Gather Request Information" in Business process 1 is matched with a group of 3 activities "Gather Issue Information", "Analyze Issue Logs" and "Identify Request Reason" in Business process 2. In the same manner activity "Cancel Service Request" of Business process 1 is matched to a group of 2 activities "Add Service Request to Backlog" and "Reject Service Request" in Business process 2.

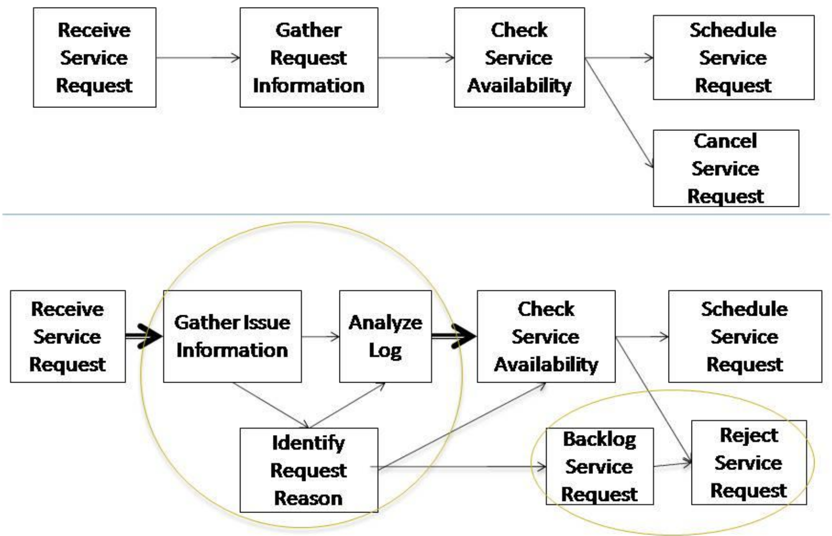


Fig. 5.2.1 Business process multivalent node match

In Fig. 5.2.2 the labels of the graphs are abstracted for explaining the graph edit distance algorithm.

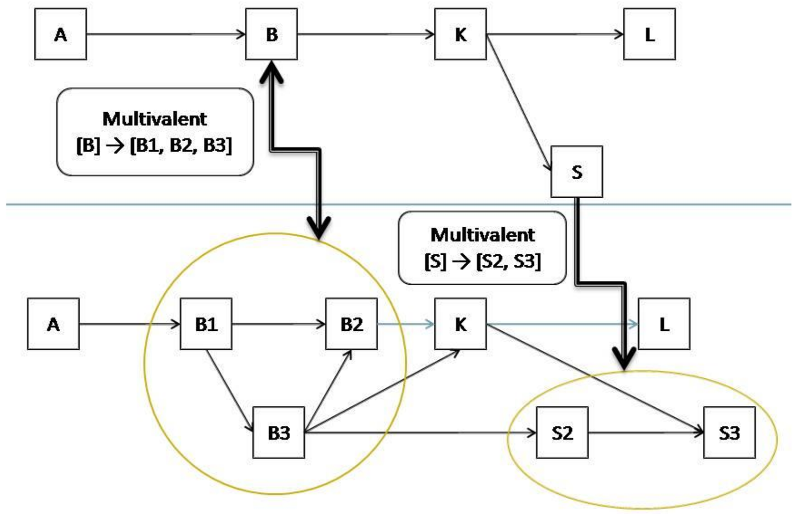


Fig. 5.2.2 Business process abstractization

The graph edit distance algorithm is applied on the business processes represented in Fig. 5.2.2 by using a one-to-one correspondence of the following nodes of input business processes:
{A(G1)={A(G2)=1.0}, B(G1)={B1(G2)=0.5}, L={L=1.0}, K={K=1.0}}

The computed set of deleted and added nodes (nodes which are not substituted) is:
|skipn| = [S3, B3, B2, S2, B1]

The computed set of deleted/added edges is:
|skipe| = [(B : K)] [(A : B), (B : K)]

The similarity distance of the substituted nodes is:
|subn| = 1.0 – 1.0 + 1.0 – 0.5 + 1.0 – 1.0 + 1.0 – 1.0 = 0.5

The fraction of inserted/deleted nodes, inserted/deleted edges and the average distance of substituted nodes is:
fskipn = 0.41666666666666667
fskipe = 0.23076923076923078
fsubn = 0.14285714285714285

And the graph edit similarity considering univalent node match and default weights value wskipn = 0.2, wskipe = 0.5 wsubn = 1.0 is 0.79.

This algorithm is extended to consider multivalent nodes and edges match.

The relation in between one node of a business process and a set of nodes composing a sub-process in the other business process is of substitution. In the current example, the following multivalent relations are detected:

[B] → [B1, B2, B3]

[S] → [S2, S3]

In a multivalent relation, one node of the first business process is matched to a sub-process of the second business process. All the edges of the first business process that are connected to a subprocess in the second business process are matched with a directed path in the second business process. The interconnected edges of the first graph are matched either to an edge or to a directed path in the second graph. We added this constraint to ensure the result of the node and edge multivalent node match is a directed graph.

For the multivalence between B and [B1, B2, B3], as we have the edges [B1,B2], [B1, B3], [B2, B3] in the second graph, edge [A, B] could be matched to [A, B1], [ [A, B1] [B1, B2]], [ [A, B1] [B1, B3] ] and [ [A, B1] [B1, B3] [B3, B2] ]. The constraint is that the last element of the directed path must be the start point for the following directed path used to match interconnected edges. If [A, B] is matched to [A, B1] than B1 must be the start node for the matching edge [B,K]. And so on.

The following combinations are identified:

[A,B]→[A,B1]
    [B,K]→[ [B1,B2] [B2,K] ]
    [B,K]→[ [B1,B3] [B3,B2] [B2,K] ]
[A,B]→[ [A,B1][B1,B2] ]
    [B,K]→[B2,K]
[A,B]→[ [A,B1][B1,B3] ]
    [B,K]→[B3,K]
    [B,K]→[ [B3,B2][B2,K] ]
[A,B]→[ [A,B1][B1,B3][B3,B2] ]
    [B,K]→[B2,K]
[K,L]→[K,L]
[K,S]→[K,S2]
[K,S]→[K,S3]
[K,S]→[ [K,S2][K,S3] ]

The multivalent edge matches combined result in the following graph matches:

[A,B][B,K][K,L][K,S]
→
{
[A,B1][B1,B2][B2,K][K,L][K,S3] (G2-1)
[A,B1][B1,B2][B2,K][K,L][K,S2] (G2-2)
[A,B1][B1,B2][B2,K][K,L][K,S2][S2,S3] (G2-3)
[A,B1][B1,B3][B3,K][K,L][K,S3] (G2-4)
[A,B1][B1,B3][B3,K][K,L][K,S2] (G2-5)
[A,B1][B1,B3][B3,K][K,L][K,S2][S2,S3] (G2-6)
[A,B1][B1,B3][B3,B2][B2,K][K,L][K,S3] (G2-7)
[A,B1][B1,B3][B3,B2][B2,K][K,L][K,S2] (G2-8)
[A,B1][B1,B3][B3,B2][B2,K][K,L][K,S2][S2,S3] (G2-9)
}

Once the multivalence relations are identified, we calculated the graph edit distance between all possible combinations and we computed the graph similarity value. The most compatible combination of business processes is G1 and G2-9.

Similarity(G1,G2-1) = 0.9142857142857143
Similarity(G1,G2-2) = 0.9142857142857143
Similarity(G1,G2-3) = 0.9285714285714286
Similarity(G1,G2-4) = 0.9142857142857143
Similarity(G1,G2-5) = 0.9142857142857143
Similarity(G1,G2-6) = 0.9285714285714286
Similarity(G1,G2-7) = 0.9285714285714286
Similarity(G1,G2-8) = 0.9285714285714286
Similarity(G1,G2-9) = 0.9428571428571428

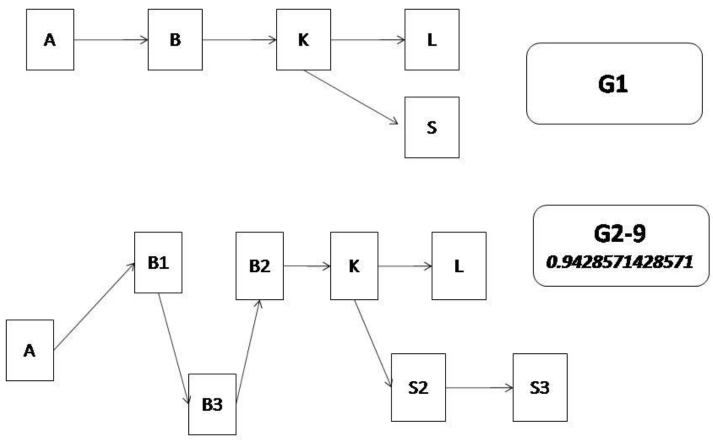


Fig. 5.2.3 Business process match

Some conclusions result after analyzing the values obtained. If one-to-one graph similarity considering univalent node match is less than 0.8, graph similarity considering multivalent node match is 0.94, an almost perfect match is found.

The major disadvantage when computing the graph edit similarity by considering multivalent nodes, in which all the possible mappings could be constructed and the one with the maximal similarity should be returned, is that the algorithm has factorial complexity. Due to this heuristics must be used for obtaining results in a feasible execution time. Such a heuristic is proposed, the greedy version of the algorithm. Greedy algorithm is based on the cost similarity matrix constructed with the similarity between the nodes of the graphs. Similarity of nodes is given by the similarity of node labels. Multivalent node matches are excluded from the computation of the final result. The algorithm selects a pair that increases the similarity the most and adds it to the global mapping. The selected nodes are discarded and the rest of the nodes are processed further until there is no open pair that could be added to the mapping. Once all open pairs are processed, the number of added/deleted nodes and added/deleted edges is calculated and the graph similarity factor is computed.

*Input:*
G1 = (N1, E1), G2 = (N2, E2)

*Initialize:*
notMatchedSet ← N1 × N2
matchedSet ← Ø

*Execute:*
while exists (n, m) € notMatchedSet where similarity (matchedSet ∪ {(n, m)}) > similarity(matchedSet) )
  extract (n,m) with maximum similarity (matchedSet ∪ {(n, m)})
  matched ← matched ∪{(n, m)}
  remove (n, m) from notMatchedSet

*Finalize:*
return similarity(matched)

The algorithm calculating the graph edit distance, greedy version, has an execution time of milliseconds.

## 5.3 Many-To-Many Model

The next extension we propose for the business process similarity algorithm is to consider modularity in business process representation and computation of the similarity. The full description of the algorthm could be found in [77]. Modularity is a design strategy which proposes the split of a complex system into smaller subsystems that could be managed independently and reused further. Each module offers a function and the global behaviour is captured by interconnecting the modules.

In BPM modularity is achieved with sub-processes. In a business process a subprocess is a component with properties: synchrounous or asynchronous, dependent or independent of the parent business process.

Modularity as a design direction could improve the business processes significantly, the process models are easier to understand and modify further, the sub-processes could be reused in other business processes, the modifications are localized inside the sub-process subject of the update. Due to the advantages of including modularity for business process representation, we've also used it for the business process similarity algorithm proposed: comparing business process models in which each module is evaluated with the coresponding module performing the same action, if such a module exists.

Once the individual subprocesses are identifyed in each input business process, the algorithm is associating modules performing the same action in each business process. For this purpose we are using the graph edit distance algorithm for calculating the similarity between all the modules of the input business processes and identifying the corespondence between the modules, detecting which are the ones performing the same action.

For each corespondence detected, module is reduced to a single node in the parent business process and the same algorithm is applied for the business processes composed of nodes as sub-process reduction elements.

For exemplification, we represented two business processes composed of modules as in Fig. 5.3.1 and Fig. 5.3.2. In the first step the coresponding module if exists is identified. Afterwards each module is reduced to a single node and the similarity factor is computed. The graph similarity algorithm has a polynomial execution time and can be applied to real data.

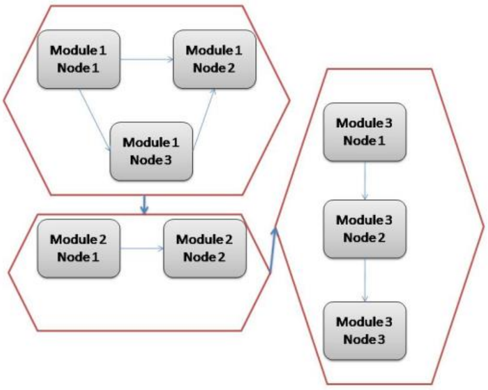


Fig. 5.3.1 Business process 1 in modular composition

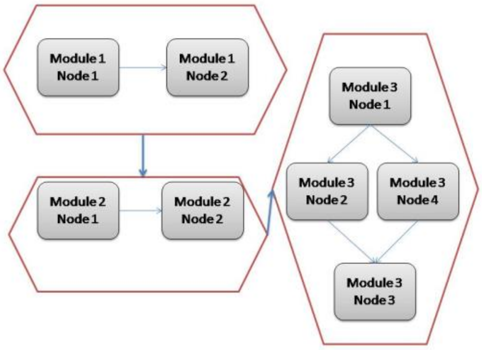


Fig. 5.3.2 Business process 2 in modular composition

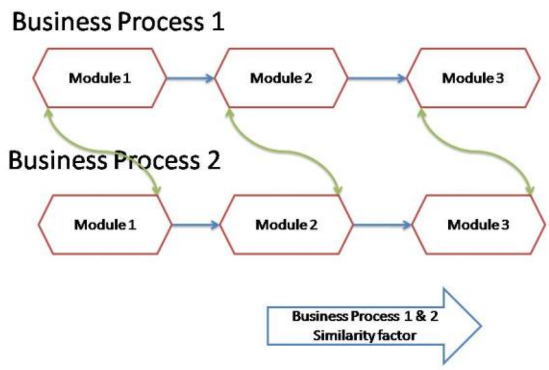


Fig. 5.3.3 Business process after reduction

The algorithm is exemplified on three sample business processes considering the modular design representation.

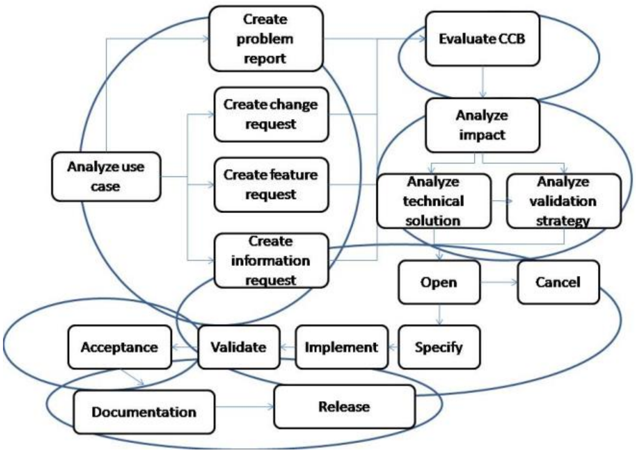


Fig. 5.3.4 Change management business process 1

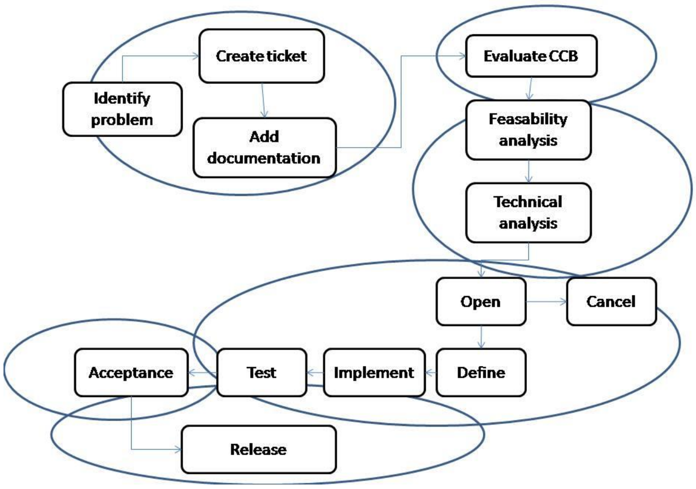


Fig. 5.3.5 Change management business process 2

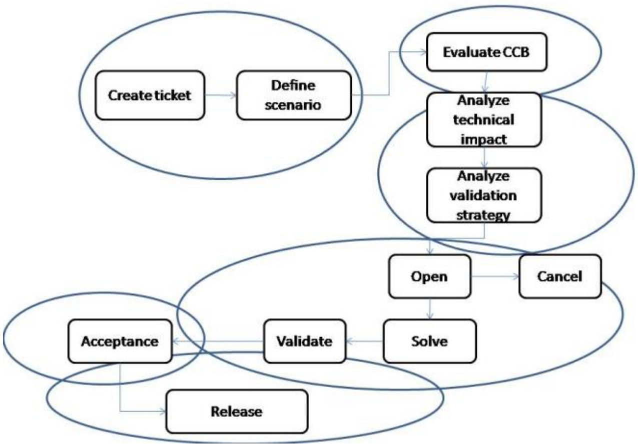


Fig. 5.3.6 Change management business process 3

The similarity between sub-processes is computed by using a semantic string similarity algorithm for label match, Wordnet Lesk Tanim - greedy version.

| SUB-PROCESS | [CMBP1, CMBP2] | [CMBP1, CMBP3] | [CMBP2, CMBP3] |
|---|---|---|---|
| Submit Change | 0.08 | 0.23 | 0.67 |
| Evaluate CCB | 1 | 1 | 1 |
| Analyze | 0.23 | 0.58 | 0.12 |
| Evaluate | 1 | 1 | 1 |

| | | | |
|---|---|---|---|
| CCB | | | |
| Solve | 0.87 | 0.79 | 0.77 |
| Evaluate CCB | 1 | 1 | 1 |
| Finish | 0.97 | 1 | 0.97 |

Table 5.3.1 Business process similarity

We included a convention in the algorithm, all sub-processes with a similarity higher than 50% are candidates for substitution.

The similarity of CMBP1 and CMBP2 in a modular design is calculated following the described algorithm:

Number of substituted sub-processes:
|subs| = 5 (Evaluate CCB, Evaluate CCB, Solve, Evaluate CCB, Finish)
Fraction of substituted sub-processes:
fsubs = 2 * (0+0+0.13+0+0.03) / 5 = 0.064

Number of replaced sub-processes:
|skips| = 2 (Submit Change, Analyze)
Fraction of substituted sub-processes:
fskips = 2/14 = 0.1428

Number of replaced edges:
|skipe| = 3 ([Submit Change, Evaluate CCB], [Evaluate CCB, Analyze], [Analyze, Evaluate CCB])
Fraction of substituted edges:
fskipe = 3/5 = 0.6

If we use the default weights: wskips = 0.1, wskipe = 0.3 and wsubs = 1, the similarity of CMBP1 and CMBP2 in a modular design approach is:
**Sim(CMBP1, CMPB2) = 0.759**

The similarity of CMBP1 and CMBP3 in a modular design is calculated on the same principle.
Number of substituted sub-processes:
|subs| = 6 (Evaluate CCB, Analyze, Evaluate CCB, Solve, Evaluate CCB, Finish)
Fraction of substituted sub-processes:
fsubs = 2 * (0+0.42+0+0.21+0+0) / 6 = 0.21
Number of replaced sub-processes:
|skips| = 1 (Submit Change)
Fraction of substituted sub-processes:
fskips = 1/14 = 0.071
Number of replaced edges:
|skipe| = 1 ([Submit Change, Evaluate CCB)
Fraction of replaced edges:
fskipe = 1/5 = 0.2

If we consider the default weights: wskips = 0.1, wskipe = 0.3 and wsubs = 1, the similarity of CMBP1 and CMBP2 in a modular design approach is:

**Sim(CMBP1, CMPB3) = 0.80**

The similarity of CMBP2 and CMBP3 in a modular design strategy is calculated on the same principle:
Number of substituted sub-processes:
|subs| = 6 (Evaluate CCB, Analyze, Evaluate CCB, Solve, Evaluate CCB, Finish)
Fraction of substituted sub-processes:
fsubs = 2 * (0+0.33+0+0.23+0+0.03) / 6 = 0.21
Number of replaced sub-processes:
|skips| = 1 (Submit Change)
Fraction of substituted sub-processes:
fskips = 1/14 = 0.071
Number of replaced edges:
|skipe| = 1 ([Submit Change, Evaluate CCB)
Fraction of substituted edges:
fskipe = 1/5 = 0.2

The similarity results considering a modular design representation:

| SIMILARITY | [CMBP1, CMBP2] | [CMBP1, CMBP3] | [CMBP2, CMBP3] |
|---|---|---|---|
| | 0.759 | 0.80 | 0.81 |

Table 5.3.2 Business process similarity in a modular design

If a modular design is not considered when computing the business process similarity, the following results are obtained:

| SIMILARITY | [CMBP1, CMBP2] | [CMBP1, CMBP3] | [CMBP2, CMBP3] |
|---|---|---|---|
| | 0.47 | 0.47 | 0.64 |

Table 5.3.2 Business process similarity without a modular design

# 6. COLLABORATIVE SOLUTION

In Ch. 5 we proposed computing the business process similarity factor for quantifying if the organizations have a similar operational perspective for handling the business cases exposed to inter-organizational collaborations. A good compatibility is a good start for an easy setup of a collaboration based on process models. If a high similarity is calculated, the collaborative solution could be constructed by merging the business processes used inside organizations for the same business case.

Collaborations fail nowadays due to several reasons. The most common ones are lack of methodology and resistance to change. By proposing a collaborative solution business process based composed of input from all participating organizations we offer support and a methodology for the collaboration and we ensure the resistance to change will not represent a problem as people will have a good familiarity with the common process model. Such a direction for collaboration has as major advantage an easy acceptance of the collaborating process model. The algorithm proposed merges two or more input business processes compatible, with a good similarity calculated for composing a collaborative process model.

One major goal for the global collaborative solution as subject of this thesis is the independence of business process representation. For this reason, in the first step the business processes are reduced to graph format. Each business process representation has a specific function associated able to transform the business process element in a graph component. The transformation works in the opposite direction as well, a graph element could be transformed backwards in a business process element.

The algorithm which we presented in [78] is based on the common parts of the input business processes abstracted as directed attributed graphs. The common graph is initialized with the set of maximum common regions. Such a region is a maximum connected subgraph composed only of matched nodes and substituted edges.

Once the set of maximum common regions is created, in the next step the added nodes and edges in both graphs are identified and added to the common graph. The weight on the graph edge is the number of graphs which contain the edge. It could have as a maximum value the total number of input graphs and the minimum value 1, the only graph containing the particular edge.

In the next step, we associate to a directed graph $G=(V,E)$ a function $w:E \rightarrow R$, which maps each edge to a number. The result is a weighted directed graph $G=(V,E,w)$.

Once the common directed graph is composed of the maximum common regions and added nodes and edges, a transitive reduction operation is applied for identifying the graph with as few edges as possible that has the same reachability level as the original graph.

In a directed graph, the transitive reduction is a directed graph satisfying the following properties:

- It contains all the nodes of the initial graph
- It contains all the directed paths of the initial graphs
- No graph with less edges satisfies the above conditions

The algorithm for composing the collaborative solution business process based is sketched below:

```
Function MergeBusinessProcess (Graph G1, Graph G2)
	Begin
	Set{Graph} MCR ← MaximumCommonRegion(G1, G2)

	Graph CG ← (NCG = {n € MCR}; ECG = {e € ECR})

	Set{Node} AddedNodes =
		ComputeAddedNodes(G1) ∪ ComputeAddedNodes(G2)

	Set{Node} AddedEdges =
		ComputeAddedEdges(G1) ∪ ComputeAddedEdges(G2)

	Foreach (Node node : AddedNodes)
		If {node} ∩ NCG = Ø
			NCG = NCG ∪ {node}

	Foreach (Edge edge : AddedEdges)
		If {edge} ∩ ECG = Ø
			ECG = ECG ∪ {edge}
		Else
			w(edge)++

	Graph TR ← ApplyTransitiveReduction(MCG)

	Return TR
End
```

As the common merged graph could have cycles, the transitive reduction is not uniquely defined. All edges of the transitive reduction must exist in the merged graph CG.

The minimum size reduction is a problem with NP hard complexity. Due to this major disadvantage, a heuristic must be considered for retrieving results in a good execution time. Such a heuristic is to use each edge successively and delete it if its removal does not change the transitive reduction.

```
Function ApplyTransitiveReduction (Graph MCG)
	Begin
	Foreach (Edge edge : GetEdges(MCG))
		List<DirectedPath> directedPaths =
	getListDirectedPath(edge.getNodeStart(), edge.getNodeEnd());

	Foreach (DirectedPath path:directedPaths)
		If (weight(path)<weight(edge))
			deleteEdge(MCG, edge);
			 break;
	End;
```

The merge of the business processes reduced to graph format is presented on a case study, the business processes reduced to graph format represented in Fig. 6.1.

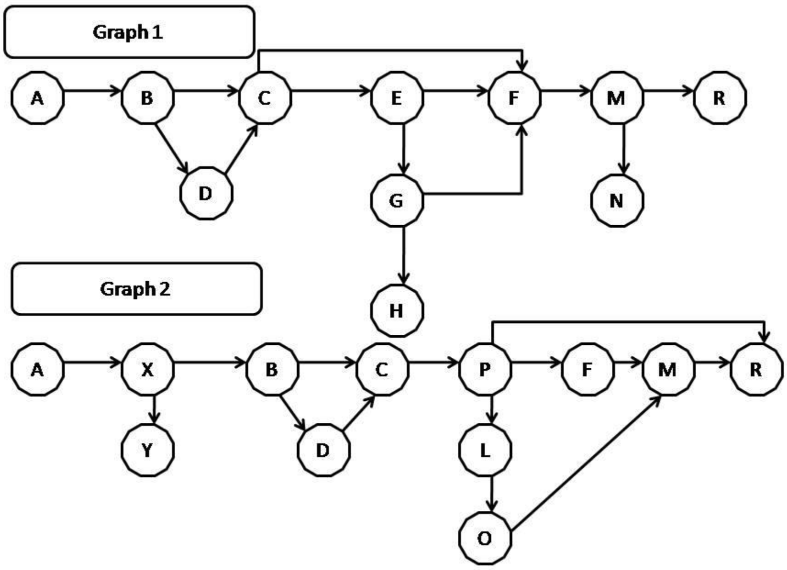


Fig. 6.1 Input business processes

The maximun common region area computed is represented in Fig. 6.2:

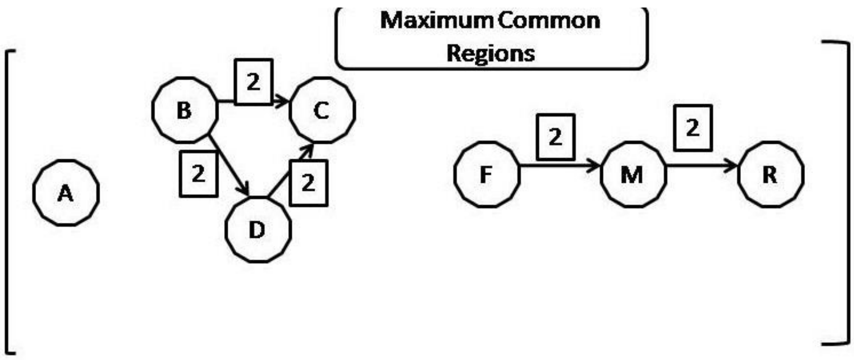


Fig. 6.2 Maximum common regions

The common graph is constructed by considering the added nodes in all input graphs. The resulted merged directed graph contains all nodes present in all input graphs. The edges are added in the resulted graph with weights. A weight represents the number of input graphs the edge is present in. The merge result of the input directed graphs is represented in Fig. 6.3.

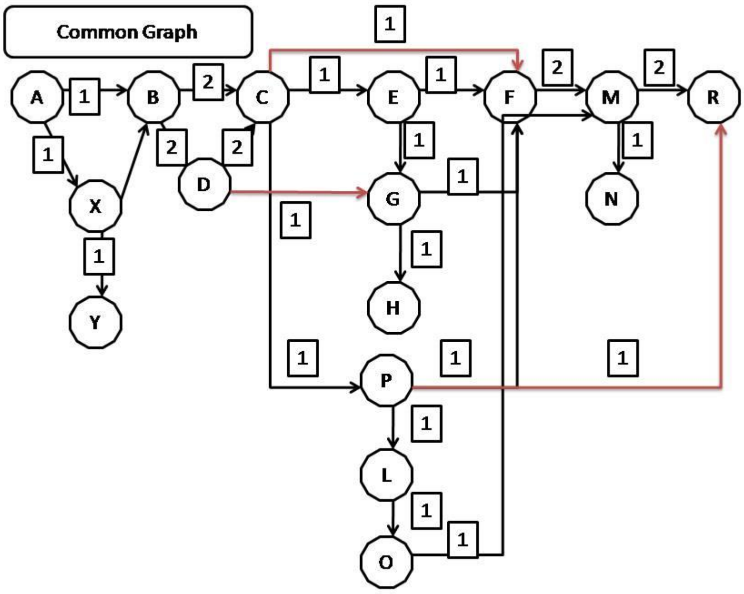


Fig. 6.3 Merged directed graph

Once a common directed path is composed, a reduction operation is applied to remove the unnecessary paths already covered. An edge could be removed if it is covered by a path with a larger weight value. Removed edges are highlighted in Fig. 6.3 and represented in Table 6.1.

| Removed Edge | Path Coverage |
|---|---|
| [C, F] | [ [C, E], [E, F]] and [ [C, P], [P, F] ] |
| [D, G] | [ [D, C], [C, E] [E, G] ] |
| [P, R] | [ [P, F], [F, M], [M, R] ] |

Table 6.1 Removed edges

The weight of edge [C, F] is lower than the weight of directed path [[C, E], [E, F]]. Edge [C, F] is not reduced. Edge [D, G] is also removed as it can be covered by directed path [[D, C], [C, E] [E, G]] and edge [P, R] is removed as it is replaced by [[P, F], [F, M], [M, R]].

The result of the transitive reduction applied is represented in Fig. 6.4.

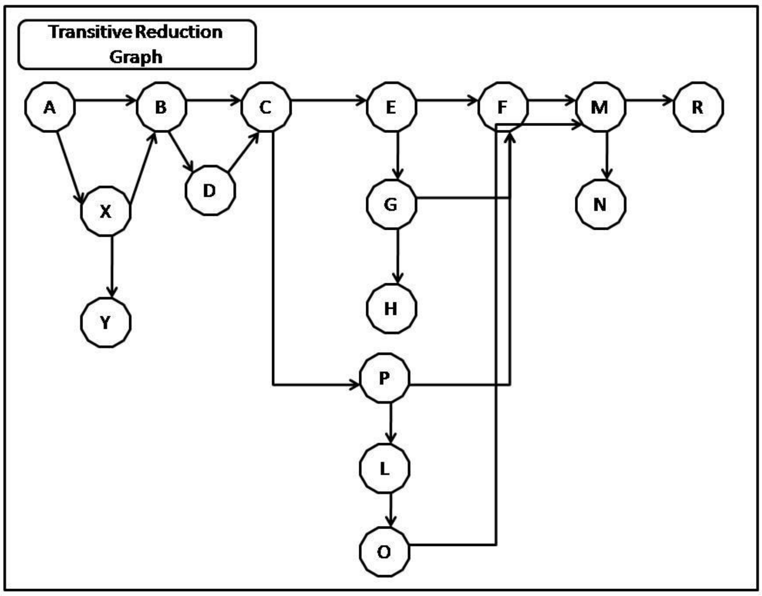


Fig. 6.4 Merged directed graph after transitive reduction

The directed graph merged and reduced is transformed into a business process. Each node is mapped to the corresponding business process element identified by analyzing the content of the label.

# 7. MONITOR AND FUNCTIONAL ASSESMENT

The collaborative business process composed in Ch. 6, as it represents the result of an automatic processing, must be monitored for a correct execution and validated for checking the compliance with predefined collaborative objectives.

## 7.1 Monitoring Execution

We proposed a monitoring application for ensuring the operational support and detecting deviations in business process execution. The business process is not enforced as we planed to ensure it can evolve using the execution information as input. During the execution of a business process, event log data is written continously. Such data could reffer to information about the cases already completed and it is called post mortem data. Post mortem data is valuable as it could be used for extracting statistical information about the results of the business process execution. Such information could be used for evaluating the quality of business process models. Also it could represent an input for predictions of the execution time and results. Pre-mortem data reffers to the cases currently being executed. These logs could be used to improve the current execution and to ensure the operational support.

The monitoring solution proposed is able to verify the pre-mortem data by matching a set of predefined rules capturing the expected behaviour. Any deviation is detected and the set of preconfigured actions is executed. No automatic change of the execution is set up. The custom monitoring solution is able to detect deviations and recommend changes to the people involved in execution with the purpose of improving the current execution. The end user owner of the activities and the owner of the rule are responsible of the changes in the execution.

The monitoring solution proposed is a Business Activity Monitoring (BAM) tool adapted to operate with process mining concepts. Gartner Group defines BAM as a solution able to provide real time information related to significant business events about a system for identifying potential problems and trigerring alerts for corrections. We've designed our custom BAM solution presented in [79] as a rule based system, the event process log is analyzed and predefined rules are matched. In case a match is found, a configured alert mechanism is activated and the corrective actions are executed.

The current BAM solution operates with data format recognized by process mining algorithms XES - eXtensible Event Stream [56]. The event log data in XES format is exported at a configured filesystem location. The alert application monitors the activities executed and triggers a preconfigured alert once a rule is broken.

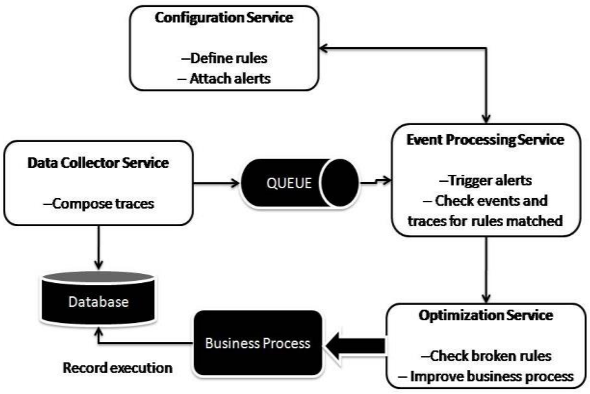


Fig. 7.1.1 Rule based BAM solution

The rule based BAM solution is composed of four services as shown in Fig. 6.1.1: Configuration Service, Event Processing Service, Data Collector Service, Delivery Service.

- **Configuration Service** is the component performing the management of the rules. Each rule has a definition and a configured alert type. Even though predefined alerts could be used, such as email notification, custom alerts could be defined and implemented upon the decision of the monitoring activity owner.
- **Data Collector Service** performs the composition of the traces out of the event log recorded by IMS. Data Collector Service polls the business process execution system and extracts the events in XML format and associates them into traces.
- **Event Processing Service** analyzes the exported event log data and performs a match with predefined rules. Any rule matched is followed by an execution of an alert preconfigured.
- **Delivery Service** is responsible of the execution of the alert. Configuration Service is responsible with definition and management of the rules.

The alert system analyzes the running cases and identifies if a potentially incorrect situation is reached. The attributes of the cases (existence, content) and the order of execution are. The rules matched are tracked and the information is used for further business process improvements. Rules are defined at event level for verifying the existence and values of the attributes, or at trace level for checking the order of events with specific attribute values. The following operators could be used:

| TYPE | OPERATORS |
|---|---|
| Logical | AND<br>OR<br>NOT<br>XOR |
| Relational | EQ – equals<br>NE – not equals<br>LT – less then |

| | |
|---|---|
| | GT – greater then<br>GE – greater or equals then<br>LE – less or equals then |
| Other | . – extract value<br>EMPTY - determine whether an attribute value is missing<br>() - change the precedence of operators<br>EXISTS – check that an attribute is defined for an event and an event with specific attributes is recorded in a trace<br>Arithmetic: +, -, *, /, % |

Table 7.1 Operators

The precedence of operators is as follows:

- .
- ()
- NOT; EMPTY; EXISTS
- / % * + -
- LT; GT; LE; GE; EQ; NE
- AND; OR; XOR

Below you can find some examples of rules defined at event level:

- Checks the incompatibility between the values of two attributes of the same event

If the two attributes ATTRIBUTE_NAME1 and ATTRIBUTE_NAME2 have specific values, the rule is matched and an alert is generated.

```
event.ATTRIBUTE_NAME1 eq VALUE1
        AND
event.ATTRIBUTE_NAME2 eq VALUE2
```

- Checks that mandatory attributes have a value

If an event has ATTRIBUTE_NAME empty, the rule is matched and an alert is generated.

```
NOT EMPTY
        (event.ATTRIBUTE_NAME)
```

- Checks the correctness of attributes for the same event

If attribute ATTRIBUTE_NAME1 is empty or the value of ATTRIBUTE_NAME2 exceeds VALUE2, the rule is matched and an alert is generated.

```
NOT EMPTY
        (event.ATTRIBUTE_NAME1)
        AND
        (event.ATTRIBUTE_NAME2 lt VALUE2)
```

- Checks if at least one of the two attributes of the same event has a value

If both attributes of the same event ATTRIBUTE_NAME1 and ATTRIBUTE_NAME2 are empty, the rule is matched and an alert is generated.

```
NOT EMPTY(event.ATTRIBUTE_NAME1)
        OR
NOT EMPTY(event.ATTRIBUTE_NAME2)
```

- Checks two incompatible values for attributes of the same event

If one of ATTRIBUTE_NAME1 and ATTRIBUTE_NAME2 has a specific value, the rule is matched and an alert is generated.

```
event.ATTRIBUTE_NAME1 eq VALUE
        XOR
event.ATTRIBUTE_NAME2 eq VALUE
```

Besides the rules defined at event level in which the values of the events are checked, rules could be defined at trace level. Rules at trace level check the order of the events in a trace with specific values.

The event with a specific value attribute inside a trace is identified with the following syntax:

```
trace.event.ATTRIBUTE eq VALUE
```

The value of an attribute for this event is accessed with:

```
(trace.event.ATTRIBUTE1 op VALUE). ATTRIBUTE2
```

Some samples of rules defined at trace level for identifying deviations are listed below:

- Check that two events of the same trace have the same value for the same attribute. The events are identified after attribute names.

```
(trace.event.ATTRIBUTE_NAME1 EQ VALUE1).ATTRIBUTE_NAME3
        EQ
(trace.event.ATTRIBUTE_NAME2 EQ VALUE2).ATTRIBUTE_NAME3
```

- Check that a trace is valid if an event identified by an attribute has the value for another attribute constant

```
EXISTS
(trace.event.ATTRIBUTE_NAME1).ATTRIBUTE_NAME2)
        EQ
PREDEFINED_VALUE
```

For exemplification, some sample rules are composed. The change management business process Fig 6.1.2 is used for this purpose. Changes are items of the following types: change requests, feature requests, problem reports and information requests. Submission of a change represents the event that triggers the execution of a process. Once submitted, a change is evaluated by a Change Control Board and the feasibility of the item is assessed (expected behaviour, constraints, impact, importance). Once the analysis is available which includes the technical aspects, testing strategy, risks, costs associated, the Change Control Board is able to approve the start of the implementation. Once the implementation is available and assessed by the Change Control Board, the item could be included in a release.

The target is that an issue reaches an end state: either it is included in a release, or it is cancelled, rejected.

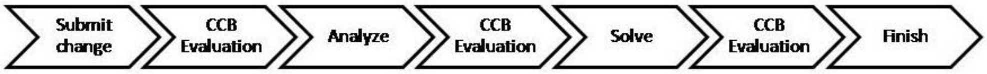


Fig. 7.1.2 Change management business process

The following rules are composed for monitoring the execution of the change management business process.

- Identify when users of type Support perform an analysis of the items

```
event.concept:name eq Analyze
        AND
event.org:group eq Support
```

- Identify when a release version is NOT included when an item is delivered

```
event.concept:name EQ Deliver
        AND
NOT EMPTY(event.release)
```

- Identify if acceptance of the items is not performed by a user of type Support

```
event.concept:name EQ Accept
        AND
NOT (event.org:group EQ Support)
```

- Identify if an item is implemented without analysis

```
EXISTS
(trace.event.concept:name EQ Analyze)
AND NOT
 (
   (trace.event.concept:name EQ Implement ).timestamp
        GT
   (trace.event.concept:name EQ Analyze).timestamp
)
```

- Identify if acceptance is not performed by the same user who created the item

```
NOT
  (
    (trace.event.concept:name EQ Create).org:resource
      EQ
```

```
    (trace.event.concept:name EQ Accept).org:resource)
```

- Identify if specific items identified by name don't have an implementation added

```
NOT (
  (
     trace.event.concept:name eq "Error indication border"
        OR
     trace.event.concept:name eq "Support path names with  forward slashes"
   )
AND
   EXISTS(trace.event.concept:name eq Implement)
)
```

Operational support provides valuable information during business process execution which permits including corrections while the effects are still visible. The proposed rule definition language is custom and it covers the needs for the business processes from real business environments evaluated for the current research. Further extensions could be considered.

## 7.2 Functional Validation

We proposed as a validation direction the functional validation which identifies if the business process obtained is compliant with the business process objectives defined for the collaboration. Functional validation of a business process checks if the process model is able to acomplish the targets of the business case realized in collaboration. The technique is presented in [80].

The start point is to create a test data set capturing the business process objectives. The use cases could be defined from three different perspectives, process, case and organizational. After use case definition, in the next phase the creation of the test dataset is performed. The test dataset must capture the behaviour expressed in the use case definition. In the last phase a validation of the process model with each use case modelled through test dataset is performed by executing conformance checking techniques on the process model with the test datasets. The results show how compliant is the business process model with business objectives modeled in use cases.

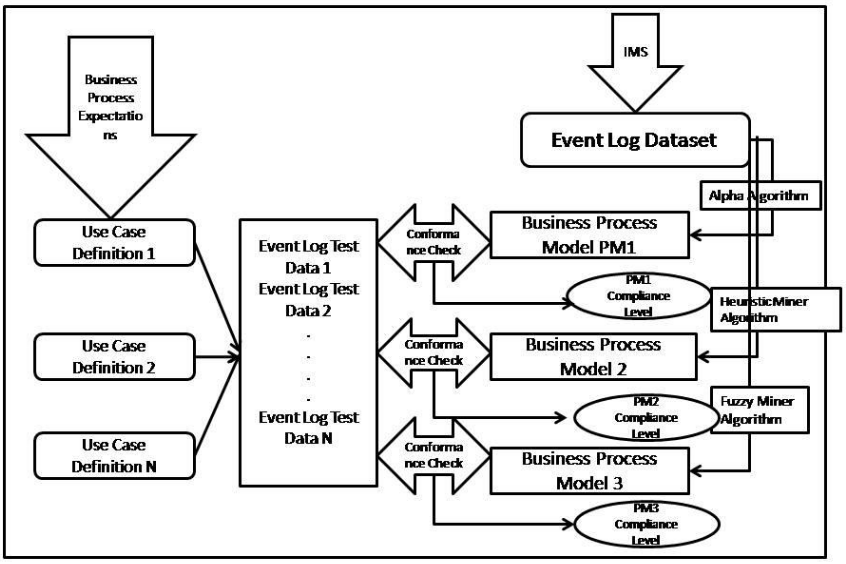


Fig. 7.2.1 Functional validation high level definition

Validation is a crucial component in any organization and it should be applied in all the areas inside an organization, including in the management of processes used for accomplishing business objectives. Validation of business processes inside an organization is even more important as the quality of the process models could impact all business areas.

We are composing automatically the collaborative business process based solution. Because it is the result of an automatic action, validation is a mandatory step.

Considering the change management business process, the following use cases are identified and transformed into test datasets:

| **USE CASE NO** | **DESCRIPTION** |
|---|---|
| Use Case 1 | The items could be created by support team, opened, analyzed, implemented, tested and released by engineering team and accepted by support team |
| Use Case 2 | The items could be created by support team, opened, analyzed and cancelled by engineering team and accepted by support team |
| Use Case 3 | Items could be created by support team, opened, implemented, tested and released by engineering team |
| Use Case 4 | Items should always be tested after implementation |
| Use Case 5 | Items should always be cancelled only after an analysis |

| | |
|---|---|
| | is performed |
| Use Case 6 | Items should always be created by users within group "Support" |
| Use Case 7 | Items should always be implemented by users within Group "Engineering Team" |
| Use Case 8 | Items should always be analyzed after delivery |

Table 6.2 Business objectives modeled as use cases

The compliance of the collaborative process model with the test dataset represents the factor for deciding if the resulted business process is suitable for accomplishing the business objectives or if it needs further adjustments. The use case approach is an efficient and effective technique for collecting essential requirements from the stakeholders of the collaborative scenario, helping them engage and focus on the real needs.

# 8. CONTINOUS IMPROVEMENT

In Ch. 7 we proposed a Business Activity Monitoring solution implementing a custom rule definition language created for ensuring the operational support and assistance in business process execution. We propose extending the BAM solution by analyzing the recorded information about the rules broken in business process execution.

The alert system designed is able to analyze in real time the execution with the purpose of indentifying problems while these could be corrected and the execution could be improved. If a rule is broken, an alert is generated. Such alerts allow acting while the execution could be improved. The main target is obtaining a better performance and better business results.

The collaborative business process based solution is the result of an automatic action. Due to this, one direction was not to enforce the business process but to consider it as a reference for supporting the collaboration. One of the reasons taken into account was that a business process must be accepted by the people executing activities inside an organization, it should not be enforced, due to a personalized implementation and execution, a lot of benefits could arise. A rule which is constantly broken in the same area indicates that a process is difficult to follow and must be improoved. Practically the business process model is rebuilt using monitoring results and statistical information. Our results are detailed in [81].

Another objective we've had in mind for the collaborative business process based solution was the flexibility. Flexibility in BPM supports localized small changes in business process definitions, the ability of process models to evolve without a complete refactoring. The business environment is under continous change, organizations in order to cope with such a dynamic environment must achieve agility and adaptation abilities. Such systems are called Adaptive Process Management Systems. All the changes occuring in a business process execution are recorded and used further to improve business processes. Any change which is occuring in multiple process instances could be reflected in changes in process schema definition.

Our target was not to offer a complete solution for Business Process Improvement but to add learning capabilities to the collaborative business process. For this purpose, we defined a new service, Optimization Service, which is able to record the repeatedly broken rules in business process execution. The record information is analyzed further and an update of the process model could be scheduled for adapting the process definition to the behavior supported by the users. The business process is used as a reference. Any broken rule in the system by several users could have as root cause an inconsistency in the process model and could be followed by a refactoring of the process model. The Optimization Service, which records the repeatedly broken rules performs an update of the business process to include the behavior desired by the users. Optimization Service has as major goal this action.

Rules at event level check the appearance of specific events or verify specific attributes in activity execution. Such rules broken are less likely to influence the business process flow. But rules broken at trace level could represent a valuable input for improvements. The result of the broken rules at trace level constitutes the input for the Optimization Service for scheduling the process model updates.

The Optimization Service compares the business process execution logs with the predefined rules in the system. If rules are difficult to formalize, a process model could be referenced instead. Instead of verifying a rule, conformance checking techniques to validate the execution with a reference process model is performed instead.

To exemplify the behaviour, we propose analyzing the business process defined using BPMN notation in Fig. 8.1.

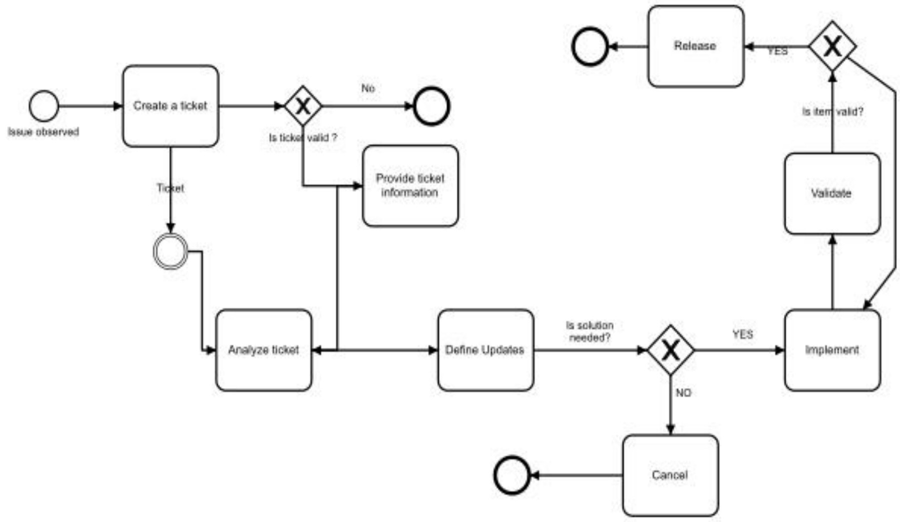


Fig. 8.1 Support management business process

We've defined some rules at trace level which are used for further business process improvements:

- Rule 1: Identify if a ticket is implemented without an analysis

```
EXISTS
(trace.event.concept:name EQ Analyze ticket)
AND NOT
 (
   (trace.event.concept:name EQ Implement ).timestamp
         GT
   (trace.event.concept:name EQ Analyze ticket).timestamp)
```

- Rule 2: Identify if Cancel is not performed by the same user who created the ticket

```
NOT
  (
    (trace.event.concept:name EQ Create a ticket).org:resource
       EQ
    (trace.event.concept:name EQ Cancel).org:resource
)
```

- Rule 3: Identify if a ticket is released without specification

```
NOT
 (
   (trace.event.concept:name EQ Release ).timestamp
         GT
```

(trace.event.concept:name EQ Define Updates).timestamp)

We operate with an abstract representation of the business process. Each process model element is abstracted in a directed attributed graph component by using two functions, one which maps nodes to labels and one which maps nodes to types. Activities model atomic work which cannot be decomposed and are modelled as nodes in the resulted graph. Events are used for modelling external or internal conditions that could influence the execution. Events are of different types, each process is initiated by a start event and could be terminated by an end event. Events are also modelled as separate nodes in the directed attributed graph. Gateways are control structures which direct the execution flow by verifying conditions. Gateways could be of different types, inclusive for parallel paths, exclusive in which a single path could be followed or event based gateways. Gateways are also modelled as nodes in the resulted graph.

One of the major targets for the current research is the independence in business process representation. For acomplishing this target, we used a function able to transform business process elements to graph elements and the opposite. The business process in Fig. 8.1 could be abstracted as in Fig. 8.2.

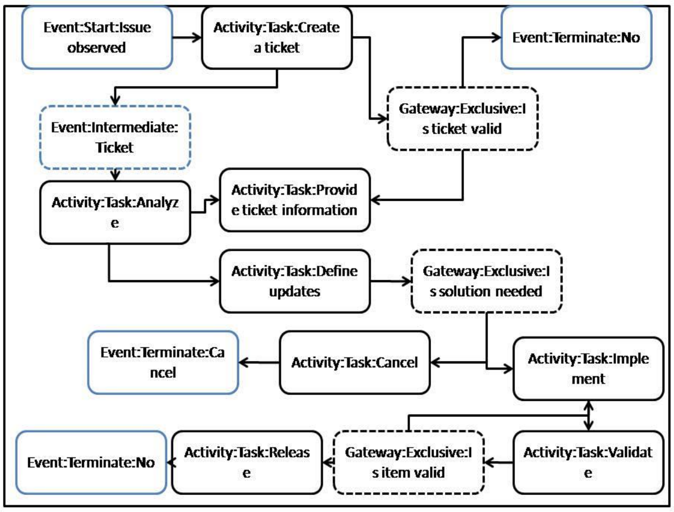


Fig. 8.2 Directed attributed graph representation of the business process

The business process is not enforced but instead, the rules broken could represent the input for further process enhancements. The operations for updating the process model are inserting and removing nodes and inserting and removing edges.

By analyzing the information extracted of Rule 1 broken constantly we conclude that Analysis activity is suitable before implementation if the ticket is of type „Feature Request" and „Change Request". So when new functionalities are added to the product. But when instead a bug is recorded, an issue of type „Problem Report", users did not consider that an Analysis helps. Therefore an update of the business process is performed. A gateway element is added which checks the type of the event ticket that started the business process execution. The exclusive gateway has two possible execution paths, one for „Feature Request" or „Change Request" which includes the analysis and one for "Problem Reports" which excludes analysis.

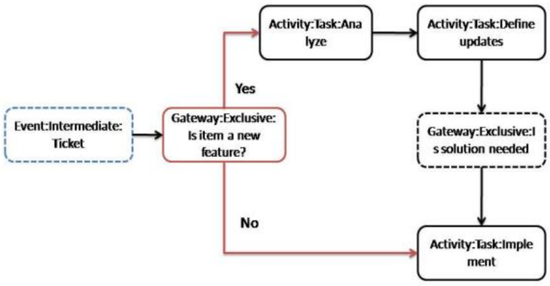


Fig. 8.3 Business process update by learning from Rule 1 event log

The updates are captured in basic graph operations:

- InsertNode(Node.Gateway, NodeType.Exclusive, "Is item a new feature", "trace.event.ticket.type")
- InsertEdge(Event:Intermediate:Ticket, Gateway:Exclusive:Is item a new feature)
- InsertEdge(Gateway:Exclusive:Is item a new feature, Activity:Task:Analyze,"trace.event.ticket.type == (Feature Request, Change Request)")
- InsertEdge(Gateway:Exclusive:Is item a new feature, Activity:Task:Implement,"trace.event.ticket.type == (Problem Report)")

Rule 2 constantly broken shows that Cancel activity could be executed after Analysis if the feasibility of the implementation shows an impact over many areas. Because of this an exclusive gateway is added for checking the feasibility of the solution proposed. If the solution is not feasible or if it has implications in many areas, the solution could be cancelled.

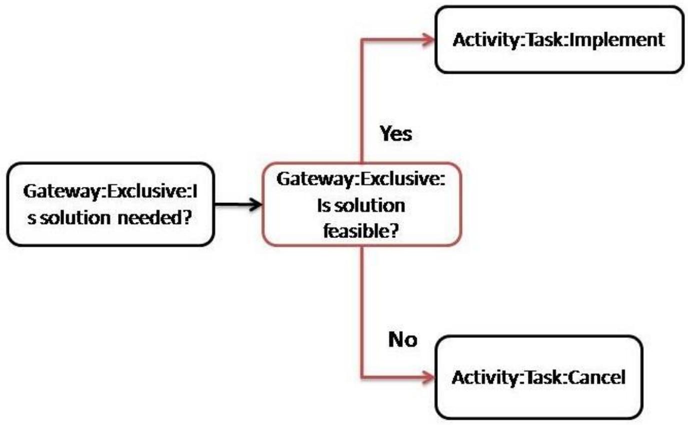


Fig. 8.4 Business process update by learning from Rule 2 event log

The updates are synthesized in basic graph operations:

- InsertNode(Node.Gateway, NodeType.Exclusive, "Is solution feasible", "trace.event.solution.feasability")
- InsertEdge(Gateway:Exclusive:Is solution needed, Gateway:Exclusive:Is solution feasible)
- InsertEdge(Gateway:Exclusive:Is solution feasible, Activity:Task:Implement,"trace.event.solution.feasability == true")
- InsertEdge(Gateway:Exclusive:Is solution feasible, Activity:Task:Cancel,"trace.event.solution.feasability == false")

Rule 3 is broken repeatedly by the end users. By analyzing the event log data the following conclusion is reached: an item could be released without specification if the Release Notes field is filled during Validation. Because of this, an exclusive gateway is added in the business process for verifying the field ReleaseNotes after Validate was executed is inserted.

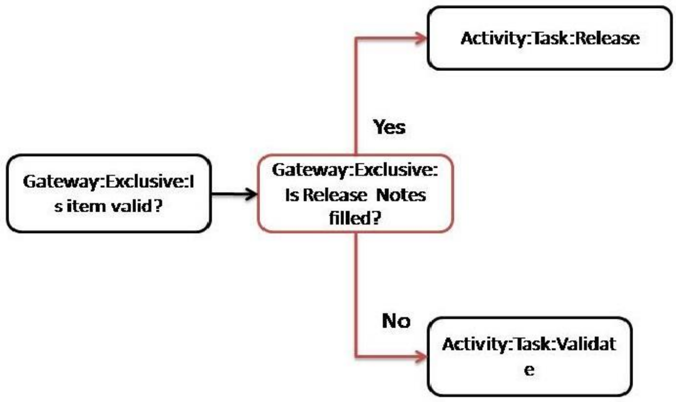


Fig. 8.5 Business process update by learning from Rule 3 event log

The updates could be captured in basic graph operations:

- InsertNode(Node.Gateway, NodeType.Exclusive, "Is Release Notes filled", "trace.event.releasenotes")
- InsertEdge(Gateway:Exclusive:Is item valid, Gateway:Exclusive:Is Release Notes filled)
- InsertEdge(Gateway:Exclusive:Is Release Notes filled, Activity:Task:Release,"NOT EMPTY(trace.event.releasenotes)")
- InsertEdge(Gateway:Exclusive:Is Release Notes filled, Activity:Task:Validate," EMPTY(trace.event.releasenotes)")

We propose a solution to integrate monitoring information in the process model for improvements and ensuring the business process evolution. The Optimization Service is able to adapt the business process to the internal profile of an organization. We consider that more freedom in business process definition in a controlled enviroment could bring benefits to the collaboration. The only disadvantage we foresee is that the updates in the process model affect the standardization as the resulted process model is custom. But standardization is not an input for the collaborative solution we proposed as we are targeting projects supported by processes realized in collaborations in a dynamic environment when time for setup is not available for formalizing the engagement between organizations.

# 9. EXEMPLIFICATION

We present in this chapter the methodology for supporting collaboration between two or more organizations on four business processes used in different environments for the same business case: **change management procedure**. Change management is the process of initiating, scheduling, implementing and evaluating changes to a system. A change management business process is composed of a set of activities executed by a team to implement modifications considered in a system.

The following change management business processes represented in Fig. 9.1, Fig. 9.2, Fig. 9.3, Fig. 9.4 are evaluated:

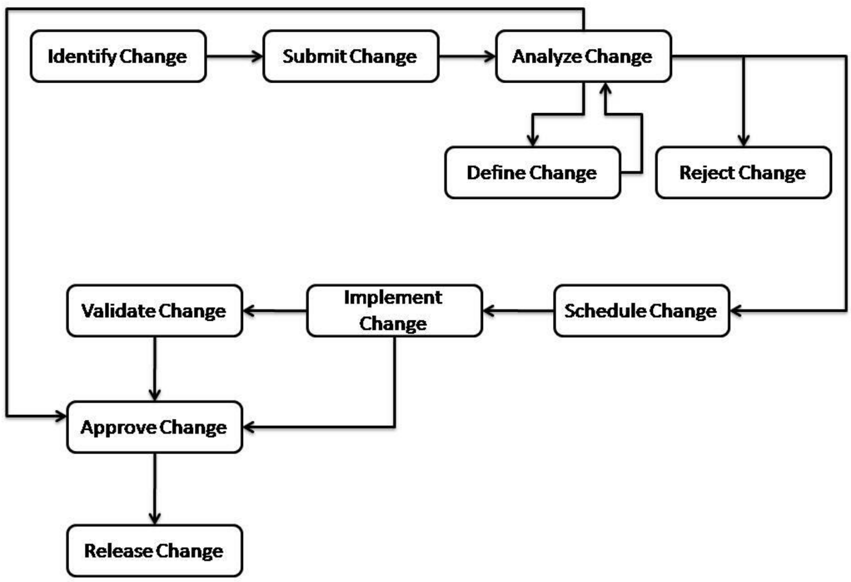


Fig. 9.1 Change management business process 1

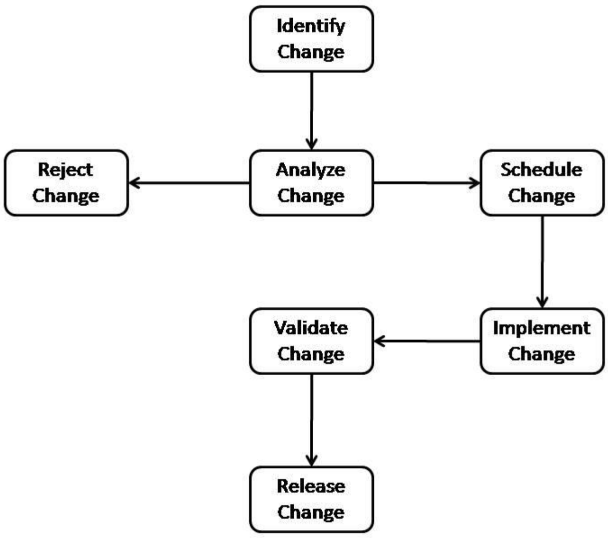


Fig. 9.2 Change management business process 2

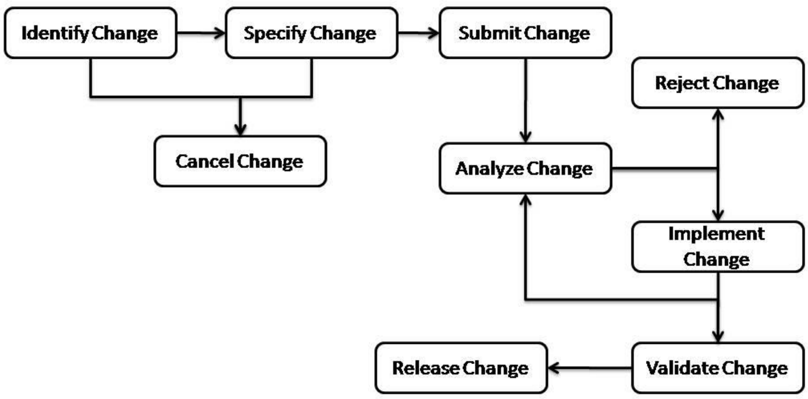


Fig. 9.3 Change management business process 3

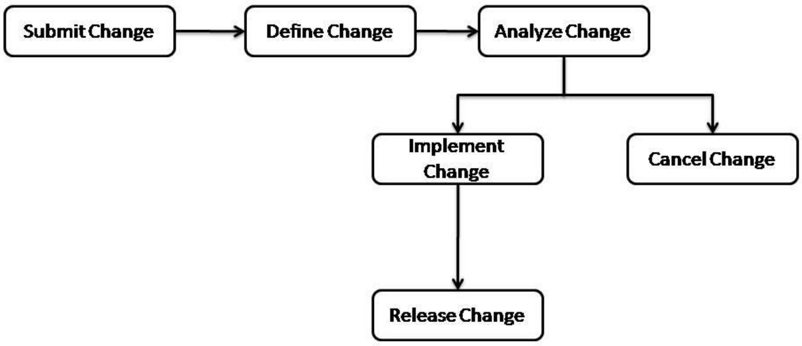


Fig. 9.4 Change management business process 4

## 9.1. Similarity Factor

In the first phase we analyze the business processes for quantifying the compatibility between organizations. Once the similarity is calculated we are able to identify the most suitable combination of organizations for collaboration. The chosen organizations will follow a common business process to consider changes in an existing system. The similarity results are presented Table 1.

As organizations belong to the same domain area, a syntactic activity label match is sufficient for similarity calculation and business process merge. WordLevent syntactic string similarity algorithm is chosen for label match.

| SIMILARITY | BP1 | BP2 | BP3 | BP4 |
|---|---|---|---|---|
| **BP1** | - | 0.60 | **0.80** | 0.67 |
| **BP2** | - | - | **0.82** | 0.60 |
| **BP3** | - | - | - | 0.67 |
| **BP4** | - | - | - | - |

Table 9.1 Business process similarity

Similarity between two business processes is computed following the procedure described in Ch. 5. First the similarity matrix is constructed which keeps the similarity of the activity labels reduced to nodes.

```
for (String nodeA : _leftGraphActivities)
```

```
  for (String nodeB : _rightGraphActivities)
    similarytyMatrix[nodeA,nodeB] = calculateLabelSimilarity(nodeA, nodeB);
```

The calculateLabelSimilarity could either use a syntactic or a semantic string similarity algorithm or an ontology mapping between the ontologies supporting the business processes. All syntactic and semantic algorithms presented are implemented and used for identifying the most suitable one for our target: compatibility between organizations from process perspective. In the current case study we are using WordLevent string comparison algorithm.

Label similarity matrix for BP1 and BP2 has the following values:

Sim[identify change(BP1) and submit change(BP2)] = 0.5
Sim[identify change(BP1) and define change(BP2)] = 0.6428571428571429
Sim[identify change(BP1) and analyze change(BP2)] = 0.42857142857142855
Sim[identify change(BP1) and cancel change(BP2)] = 0.5
Sim[identify change(BP1) and implement change(BP2)] = 0.4666666666666667
Sim[identify change(BP1) and release change(BP2)] = 0.5
Sim[specify change(BP1) and submit change(BP2)] = 0.6153846153846154
Sim[specify change(BP1) and define change(BP2)] = 0.6153846153846154
Sim[specify change(BP1) and analyze change(BP2)] = 0.46153846153846156
Sim[specify change(BP1) and cancel change(BP2)] = 0.5384615384615384
Sim[specify change(BP1) and implement change(BP2)] = 0.5333333333333333
Sim[specify change(BP1) and release change(BP2)] = 0.46153846153846156
Sim[submit change(BP1) and submit change(BP2)] = 1.0
Sim[submit change(BP1) and define change(BP2)] = 0.5
Sim[submit change(BP1) and analyze change(BP2)] = 0.46153846153846156
Sim[submit change(BP1) and cancel change(BP2)] = 0.5
Sim[submit change(BP1) and implement change(BP2)] = 0.5333333333333333
Sim[submit change(BP1) and release change(BP2)] = 0.46153846153846156
Sim[analyze change(BP1) and submit change(BP2)] = 0.46153846153846156
Sim[analyze change(BP1) and define change(BP2)] = 0.5384615384615384
Sim[analyze change(BP1) and analyze change(BP2)] = 1.0
Sim[analyze change(BP1) and cancel change(BP2)] = 0.5384615384615384
Sim[analyze change(BP1) and implement change(BP2)] = 0.5333333333333333
Sim[analyze change(BP1) and release change(BP2)] = 0.5384615384615384
Sim[implement change(BP1) and submit change(BP2)] = 0.5333333333333333
Sim[implement change(BP1) and define change(BP2)] = 0.5333333333333333
Sim[implement change(BP1) and analyze change(BP2)] = 0.5333333333333333
Sim[implement change(BP1) and cancel change(BP2)] = 0.4666666666666667
Sim[implement change(BP1) and implement change(BP2)] = 1.0
Sim[implement change(BP1) and release change(BP2)] = 0.5333333333333333
Sim[reject change(BP1) and submit change(BP2)] = 0.5833333333333334
Sim[reject change(BP1) and define change(BP2)] = 0.5833333333333334
Sim[reject change(BP1) and analyze change(BP2)] = 0.46153846153846156
Sim[reject change(BP1) and cancel change(BP2)] = 0.5
Sim[reject change(BP1) and implement change(BP2)] = 0.6
Sim[reject change(BP1) and release change(BP2)] = 0.6923076923076923
Sim[validate change(BP1) and submit change(BP2)] = 0.5
Sim[validate change(BP1) and define change(BP2)] = 0.5714285714285714
Sim[validate change(BP1) and analyze change(BP2)] = 0.5714285714285714
Sim[validate change(BP1) and cancel change(BP2)] = 0.5

Sim[validate change(BP1) and implement change(BP2)] = 0.466666666666666
Sim[validate change(BP1) and release change(BP2)] = 0.6428571428571429
Sim[cancel change(BP1) and submit change(BP2)] = 0.5
Sim[cancel change(BP1) and define change(BP2)] = 0.5
Sim[cancel change(BP1) and analyze change(BP2)] = 0.5384615384615384
Sim[cancel change(BP1) and cancel change(BP2)] = 1.0
Sim[cancel change(BP1) and implement change(BP2)] = 0.4666666666666667
Sim[cancel change(BP1) and release change(BP2)] = 0.46153846153846156
Sim[release change(BP1) and submit change(BP2)] = 0.46153846153846156
Sim[release change(BP1) and define change(BP2)] = 0.6153846153846154
Sim[release change(BP1) and analyze change(BP2)] = 0.5384615384615384
Sim[release change(BP1) and cancel change(BP2)] = 0.46153846153846156
Sim[release change(BP1) and implement change(BP2)] = 0.5333333333333333
Sim[release change(BP1) and release change(BP2)] = 1.0

Based on the label similarity matrix, nodes from one business process presented as source are mapped to nodes in the second business process presented as destination. The same process is applied in the opposite direction. Once the mapping is performed we are able to compute the set of added/deleted nodes, added/deleted edges and the similarity distance calculated using substituted nodes.

Mapping BP1->BP2 identify change->define change
Mapping BP1->BP2 implement change->implement change
Mapping BP1->BP2 analyze change->analyze change
Mapping BP1->BP2 cancel change->cancel change
Mapping BP1->BP2 specify change->submit change
Mapping BP1->BP2 validate change->release change
Mapping BP1->BP2 reject change->release change
Mapping BP1->BP2 submit change->submit change
Mapping BP1->BP2 release change->release change

The set of added/deleted nodes and added/deleted edges for BP1 and BP2 is:

Added nodes [define change]

Deleted nodes []

Added edges [(validate change : analyze change), (identify change : cancel change), (validate change : cancel change), (identify change : specify change), (submit change : cancel change), (validate change : release change), (implement change : validate change), (specify change : submit change), (submit change : analyze change), (analyze change : reject change), (submit change : implement change)]

Deleted edges [(validate change : analyze change), (identify change : cancel change), (validate change : cancel change), (identify change : specify change), (submit change : cancel change), (validate change : release change), (specify change : submit change), (submit change : analyze change), (analyze change : reject change), (submit change : implement change)]

The graph edit distance induced by this mapping is 1.4065934065934065.

The fractions of added/deleted nodes, added/deleted edges and substituted nodes are calculated according to the predefined formulas detailed in Ch.5.

fskipn = 0.06666666666666667
fskipe = 1.1666666666666667
fsubn = 0.2009419152276295

And the business process similarity for BP1 and BP2 computed is 0.6017081557897884.

The same mechanism is used for identifying the similarity between all business processes.

## 9.2. Collaborative Model

In the second phase, as the similarity of business processes is high, all organizations are compatible and could target collaborations for achieving common business results in change management activities.

The resulted merged directed graph of BP1 and BP2 contains all nodes present in all input graphs and is presented in Fig. 9.2.1. The edges are added in the resulted graph with weights. A weight represents the number of input graphs the edge is present in.

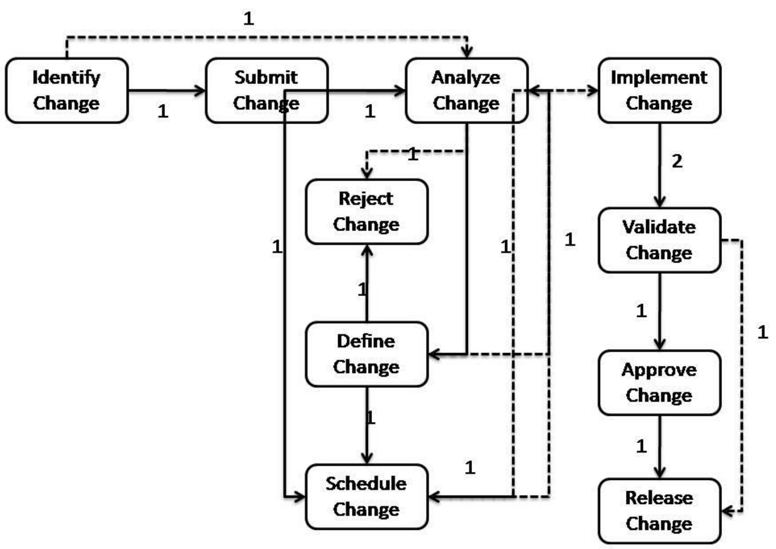


Fig. 9.2.1 Merge business process 1 and 2

After applying transitive reduction the edges covered by paths with a higher weight value are removed from the resulted graph. Removed edges are listed in Table 2 and are represented with dotts in the merged graph.

| REMOVED EDGE | PATH COVERAGE |
|---|---|
| [Identify Change, Analyze Change] | [[Identify Change, Submit Change], [Submit Change, Analyze Change]] |
| [Validate Change, Release Change] | [ [Validate Change, Approve Change], [Approve Change, Release Change] ] |
| [Analyze Change, Schedule Change] | [ [Analyze Change, Define Change], [Define Change, Schedule Change] ] |
| [Analyze Change, Reject Change] | [ [Analyze Change, Define Change], [Define Change, Reject Change] ] |

Table 9.2 Transitive reduction on merged graph BP1 and BP2

The common graph BP1 and BP3 is constructed on the same principle and presented in Fig. 9.2.2:

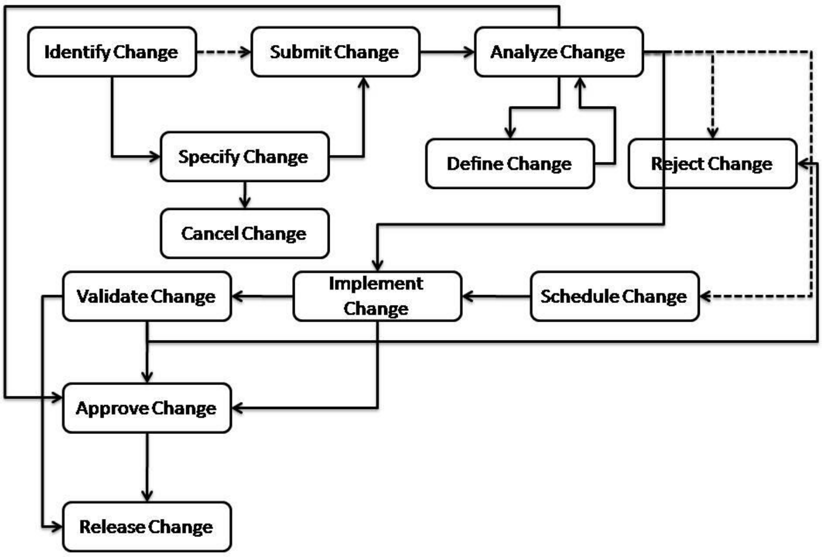


Fig. 9.2.2 Merge business process 1 and 3

After applying transitive reduction, edges covered by paths with a higher weight are removed considering the same principle as for BP1 and BP2.

| REMOVED EDGE | PATH COVERAGE |
|---|---|

| | |
|---|---|
| [Identify Change, Submit Change] | [[Identify Change, Specify Change], [Specify Change, Submit Change]] |
| [Analyze Change, Reject Change] | [ [Analyze Change, Define Change], [Define Change, Reject Change] ] |
| [Analyze Change, Schedule Change] | [ [Analyze Change, Define Change], [Define Change, Schedule Change] ] |

Table 9.3 Transitive reduction on merged graph BP1 and BP3

The common graph BP1 and BP4 is constructed on the same principle and presented in Fig. 9.2.3:

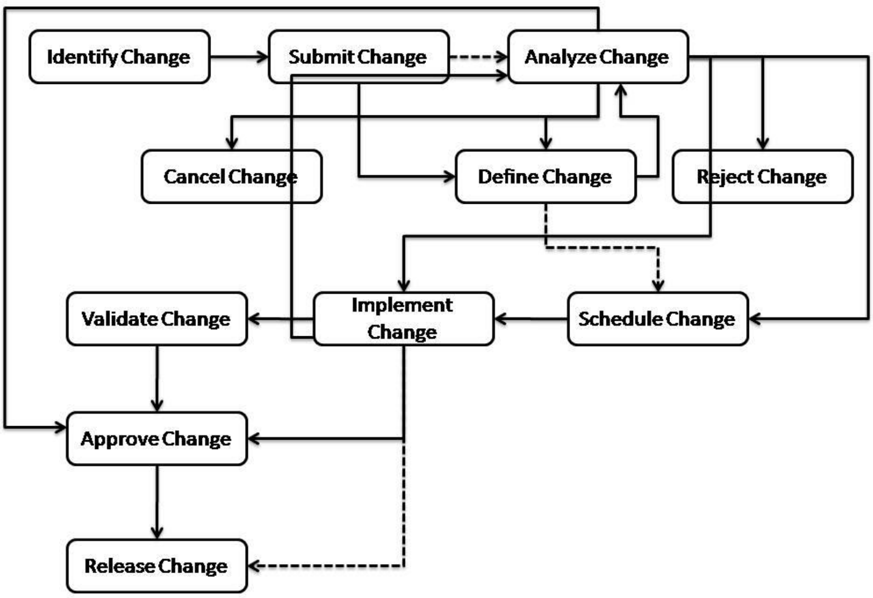


Fig. 9.2.3 Merge business process 1 and 4

After applying transitive reduction edges covered by paths with a higher weight are removed.

| REMOVED EDGE | PATH COVERAGE |
|---|---|
| **[Submit Change, Analyze Change]** | [ [Submit Change, Define Change], [Define Change, Analyze Change]] |

| | |
|---|---|
| **[Define Change, Schedule Change]** | [ [Define Change, Analyze Change], [Analyze Change, Schedule Change] ] |
| **[Implement Change, Release Change]** | [ [Implement Change, Validate Change], [Validate Change, Approve Change], [Approve Change, Release Change] ] |

Table 9.4 Transitive reduction on merged graph BP1 and BP4

The common graph BP2 and BP3 is constructed on the same principle and presented in Fig. 9.2.4:

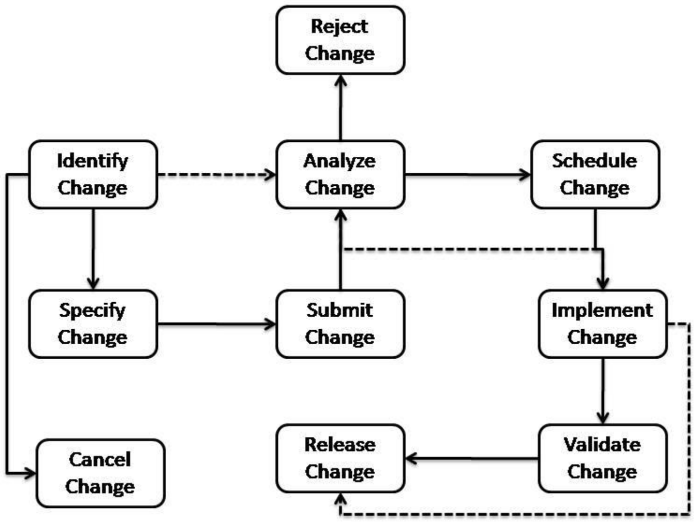


Fig. 9.2.4 Merge business process 2 and 3

After applying transitive reduction edges covered by paths with a higher weight are removed.

| REMOVED EDGE | PATH COVERAGE |
|---|---|
| **[Identify Change, Analyze Change]** | [[Identify Change, Specify Change], [Specify Change, Submit Change], [Submit Change, Analyze Change]] |
| **[Analyze Change,** | [[Analyze Change, Schedule Change], |

| | |
|---|---|
| **Implement Change]** | [Schedule Change, Implement Change]] |
| **[Implement Change, Release Change]** | [[Implement Change, Validate Change], [Validate Change, Release Change]] |

Table 9.5 Transitive reduction on merged graph BP2 and BP3

The common graph BP2 and BP4 is constructed on the same principle and presented in Fig. 9.2.5:

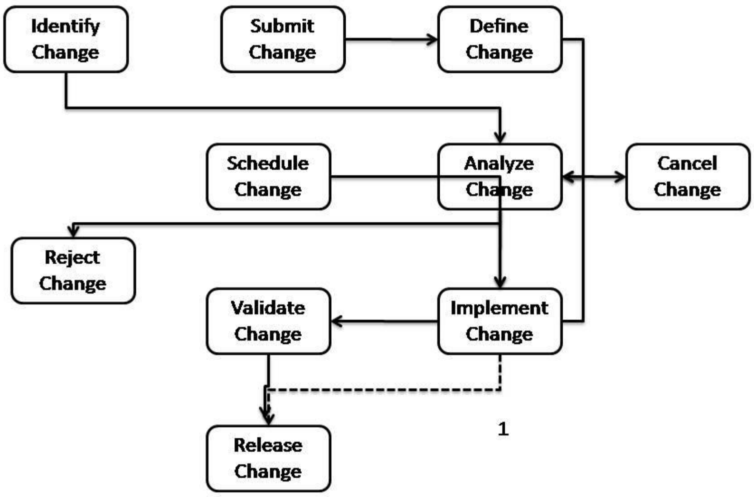


Fig. 9.2.5 Merge business process 2 and 4

After applying transitive reduction edges covered by paths with a higher weight are removed.

| **REMOVED EDGE** | **PATH COVERAGE** |
|---|---|
| **[Implement Change, Release Change]** | [[Implement Change, Validate Change], [Validate Change, Release Change]] |

Table 9.6 Transitive reduction on merged graph BP2 and BP4

The common graph BP3 and BP4 is constructed on the same principle and presented in Fig. 9.2.6:

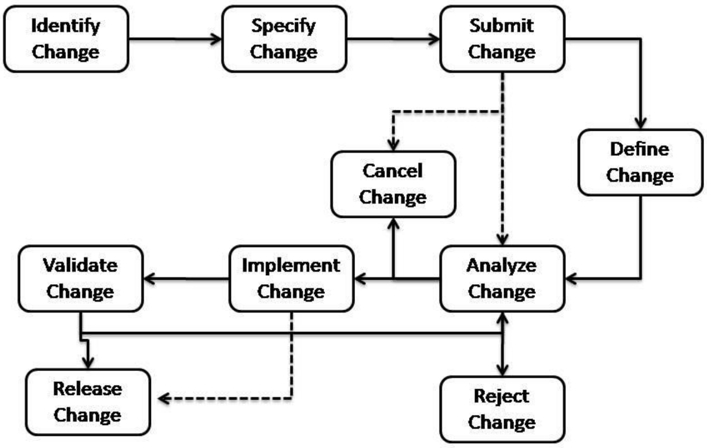


Fig. 9.2.6 Merge business process 3 and 4

After applying transitive reduction edges covered by paths with a higher weight are removed.

| REMOVED EDGE | PATH COVERAGE |
|---|---|
| **[Submit Change, Cancel Change]** | [[Submit Change, Analyze Change],<br>[Analyze Change, Cancel Change]] |
| **[Submit Change, Analyze Change]** | [[Submit Change, Define Change],<br>[Define Change, Analyze Change]] |
| **[Implement Change, Release Change]** | [[Implement Change, Validate Change],<br>[Validate Change, Release Change]] |

Table 9.7 Transitive reduction on merged graph BP3 and BP4

More than two organizations could start collaborations. Our methodology could be extended to several business processes used in different organizations.

## 9.3. Monitor Execution

Once a common business process is identified and it is applied for achieving business results in collaborations, we used the alert system to detect deviations in process execution by analyzing event logs of running cases. The following alerts are configured:

- Detect when users with Support permissions analyze an item

```
event.concept:name eq Analyze
        AND
event.org:group eq Support
```

- Detect when a release version is NOT included at delivery

```
event.concept:name EQ Deliver
        AND
NOT EMPTY(event.release)
```

- Detect if acceptance of the items is not performed by a user with Support profile

```
event.concept:name EQ Accept
        AND
NOT (event.org:group EQ Support)
```

- Detect if an item is implemented without analysis

```
EXISTS
(trace.event.concept:name EQ Analyze)
AND NOT
 (
   (trace.event.concept:name EQ Implement ).timestamp
        GT
   (trace.event.concept:name EQ Analyze).timestamp
)
```

# 10. EVALUATION

For computing the business process similarity factor, we are using adapted versions of the graph edit distance algorithm. Adaptations are introduced for the synchronization with business process representation, for considering the business context (syntactical label interpretation, semantical context, ontological reduction of the semantical space), for adapting to the custom relations between business processes: one-to-one, one-to-many (interconnect a node with a sub-process) and many-to-many (interconnect sub-processes).

Graph edit distance represents a class of algoritms of complexity NP-hard. A demonstration of this aspect could be found in [82]. Due to this disadvantage an exact algorithm cannot be applied on large graphs and heuristics are evaluated and implemented in this research. The heuristic we decided to use in this thesis is the Greedy algorithm which constructs a mapping between two process graphs by adding in each step the pair of nodes which increases the most the global graph similarity value. When the algorithm is started, all possible pairs of nodes are marked as open. One node could be added to a mapping only once; because of this, open pairs which could increase the global similarity in future steps are ignored. When there is no open pair which could be added to the mapping, the algorithm is ended. The complexity of the Greedy algorithm is $O(n^3)$, n is the number of nodes of the largest graph. Several other heuristics could be considered instead of the Greedy algorithm but for the purpose of this thesis the Greedy version of the Graph edit distance algorithm gives good results.

The merge algorithm used for composing the collaborative solution is based on the extraction of the maximum common regions for input graphs. This problem is NP hard as it's a reduction of the clique problem. As input graphs are reductions of process models, exact algorithms cannot be considered a feasible approach, exhaustive serach and branch and bound techniques are dismissed. We've used instead an Heuristic, all the possible subsets of the first graph are verified in the second graph. The complexity of this algorithm is $O(2^n)$. Once the set is available, the added and deleted nodes and edges are inserted and a common graph is composed. Resulted graph is reduced by applying a transitive reduction operation. The reduction is also a problem of NP hard complexity and an an heuristic is considered for retrieving results in a good execution time. Such an heuristic is to use each edge successively and delete it if its removal does not change the transitive reduction.

Practically all the algorithms are executed in polynomial execution time due to the Heuristics introduced. Surely several side effects could be possible as optimal solutions are not computed; the choice of the best option in the next step might influence the behaviour of the algorithm as a perspective and the results. The major advantage is the limited computational resources allocated.

# 11. CONCLUSIONS

A multi-organizational context for achieving business objectives is a clear goal for many organizations and the number is exponentially growing. For the current research we are targeting to offer technical support to ensure an easier setup of the collaboration and perspectives for success in realizing the expected business output.

Our major objective is to offer support for achieving projects in collaboration in a dynamic market considering the small amount of time available for the setup. The collaboration is based on a business process based solution.

The solution proposed is completely automated and it is constructed using as an input the event log data tracking the activities executed in organizations for the business case targeted for collaboration. As a comparison with existing collaborative solutions on the market, all of them require continous action for setting up and supporting the collaboration. The solution we propose offers a base for acomplishing business objectives in a collaborative configuration and it could be enhanced and improved further.

Another drawback considered when designing the business process based model is that most process based solutions existent on the market require a full refactoring of working modes and a complete change of the culture of an organization. Such a direction is suitable for very few organizations. Each organization is unique and a compatible partner for collaboration can be found without unifying and linearizing organizations. There is value in diversity and complementarianism. We propose finding the most suitable organization for a collaboration limited to a specific business case which represents the scope for the collaboration. Once the right partner is identified, the collaboration should be constructed on the business process information of all participating organizations. We ensure the solution is compliant with the expectations and we offer operational support.

In summary the major advantages of the solution we propose which is unique as a collaborative methodology are simplicity, automation, usability, easy setup phase.

As disadvantages we mention the lack of standardization as the merged business process based solution is custom considering the needs of both participating organizations, the possibility of introducing errors due to the event log data as a source for business process discovery which represents the base for the collaborative model, heuristics used as an alternative to exponential execution time of the algorithms which could lead to sub-optimal results.

Besides proposing an alternative to the existing collaborative solutions on the market having as a target the capability of the system to adapt to the high level of dynamism and to short term alliances project based, we contributed in all the areas of the solution:

- quantifying the compatibility between organizations
- composing a process based collaborative solution
- monitoring and controlling
- validation of process models with business objectives
- improvements and re-engineering

For the organizational compatibility factor, we improved the business process comparison algorithm by proposing a limitation of the search space to the ontologies of input business processes automatically identified. The business vocabulary and rules are computed and used for ontology specification. Business process comparison is executed in a limited search space of the ontology mapping result. The organizational compatibility factor is calculated in a polynomyal execution time of ms.

The collaborative solution proposed is based on an abstraction of business process definition, input business processes are reduced to graph format, with the purpose of considering the topology in the compatibility factor computation. We also achieved the independence of the maturity level of the participating organizations; collaborations should not be limited by the standardization level of the organizations. The compatibility factor is a single float number easy to interpret and use further. For the current thesis we proposed a business process based merge algorithm able to cope with any kind of business process format.

In case business processes are not formalized and are not available as input, these could be extracted by analyzing the event log data and applying process mining algorithms. Like this the compatibility could be measured even if organizations don't have clear processes followed internally.

Once a collaborative solution is formalized to support the collaboration, we offer techniques for monitoring the execution and offering operational support. The solution we proposed in this area is based on a standardized event log data format, XES, which is the format supported by process mining algorithms. The novelty is the rule definition language proposed operating with process mining concepts.

Another area in which we proposed alternatives is the validation of process models. We proposed including a functional validation besides conformance validation in which we ensure the business process is compliant with collaborative business objectives. The business targets are modeled as datasets and conformance of the process model is verified on the test datasets.

Once the business process is executed, we proposed as an improvement direction using the monitoring input for enhancing the process model and ensuring it is compliant with the user's expectation. This improvement direction in which broken rules are used further for updating the process model is the element of novelty in the area of improving the full collaborative solution.

# 12. LIMITATIONS

We identified the following limitations for the methodology and the collaborative solution proposed. One already mentioned is that a custom solution composed of input from all participating organizations is flexible, adaptive but not standard. The resulted business process is specific for the collaborative organizational context and could be improved by learning from monitoring results, again, a source of unstandardized transformation. Flexibility and easy adaptation is also a goal we've had in mind when designing the solution but standardization could be an advantage in a specific business context.

Another difficulty we are considering as a limitation is that the collaborative process model is composed out of input retrieved from all participant organizations. If the business processes are not formalized inside organizations process mining techniques are used for discovery and association with business cases. As the log data could have inconsistencies in a real environment because of different sources of information and formats, the resulting process models could have inconsistencies. As these process models are the source for the collaborative business process based solution, errors could be introduced such as incorrect process paths, bottlenecks. As a workaround we proposed several validation techniques to ensure the quality of the process model and the compliance with business objectives.

One of the concerns we are having with the proposed solution is represented by the difficulties to validate the similarity factor. The computed value is simple to interpret, this was a major goal for the current research but it's difficult to asses if the value is correct. One of the possible sources of inconsistencies are the heuristics used as alternatives to optimal algorithms versions to overcome the disadvantages of exponential execution time and high resource utilization.

The compatibilty factor is calculated after performing an automatic ontology extraction and mapping of ontologies for participating organizations. This could lead to unexpected results in which elements are incorrectly mapped, problem which could have as a major drawback an inconsistent similarity value. The solution could be represented by a manual action performed by a business analyst validating the resulted ontologies and the mapping perfomed.

The current collaborative solution is composed of the node content for each involved business process. No new nodes (activities, gateways, events) are added or removed as the logic is not interpreted. Such actions are possible in a manual phase performed by a business process analyst after the collaborative process model is extracted.

We proposed a custom business activity monitoring solution operating with process mining concepts. To use a standardized format for event log representation and ensuring operational support was a major target of the current research. The side effect could be the difficulties in representing the rules as the process mining concepts of events and trace are used.

The solution we propose has as a major business case realizing dynamic projects when time is not available for the setup of the collaboration. The goal is not to integrate business processes of different industies or composed for different business cases.

All the limitations we foresee could be easily solved through a manual contribution.

# 13. ACKNOWLEDGMENTS

This thesis was partially supported by the Sectoral Operational Programme Human Resources Development POSDRU/159/1.5/S/137516 financed from the European Social Fund and by the Romanian Government.

# 14. PUBLICATIONS


1. Maria Laura Sebu, Horia Ciocarlie – Applied process mining in software development (2014) – Proceedings of SACI, 9th IEEE International Symposium on Applied Computational Intelligence and Informatics

2. Maria Laura Sebu, Horia Ciocarlie - Process mining functional and structural validation (2014) – Proceedings of SOFA, 6th IEEE International Workshop on Soft Computing Applications IEEE

3. Maria Laura Sebu, Horia Ciocarlie - Business activity monitoring solution to detect deviations in business process execution (2015) Proceedings of SACI, 10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics

4. Maria Laura Sebu, Horia Ciocarlie - Business process similarity metric supporting one-to-many relationship (2015) – Proceedings of SACI 10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics

5. Maria Laura Sebu, Horia Ciocarlie - Cross-organizational compatibility detection with process mining (2015) - Buletinul Institutului Politehnic din Iasi, Universitatea Tehnică „Gheorghe Asachi" din Iaşi, Tomul LVIII (LXII), Fasc. 1

6. Maria Laura Sebu, Horia Ciocarlie - Merging Business Processes for a Common Workflow in an Organizational Collaborative Scenario (2015) Proceedings of ICSTCC 19th International Conference on System Theory, Control and Computing

7. Maria Laura Sebu, Horia Ciocarlie - Similarity of business process models in a modular design (2016) Proceedings of SACI 11th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics

8. Maria Laura Sebu, Horia Ciocarlie - Optimizing business processes by learning from monitoring results (2016) – Proceedings of SACI 11th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics

9. Maria Laura Sebu, Horia Ciocarlie (2016) - Collaborative Business Process Solution Considering an Ontological Dimension of Process Models; Proceedings of SOFA, 7th IEEE International Workshop on Soft Computing Applications IEEE

# References

[1] Emanuela Todeva, David Knoke (2005) - Strategic alliances and models of collaboration, Management Decision, Vol. 43 Issue 1, pp. 123–148

[2] B. Faqihi, N. Daoudi, and R. Ajhoun (2015) - From the Collaboration of Companies to the Interoperability of Information Systems (Concepts and Perspectives), International Journal of Information and Education Technology, Vol. 5, No. 3, pp. 208-214

[3] W.M.P. van der Aalst (2001) - Process Mining: Discovery, Conformance and Enhancement of Business Processes, Book Springer Publishing Company, ISBN:3642193447 9783642193446

[4] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster (2004) - Workflow Mining: Discovering Process Models from Event Logs, IEEE Transactions on Knowledge and Data Engineering, pp. 1128-1142

[5] A.J.M.M. Weijters and J.T.S. Ribeiro (2010) - Flexible Heuristics Miner (FHM), BETA Working Paper Series, Eindhoven University of Technology

[6] A.J.M.M. Weijters and W.M.P. van der Aalst (2003) - Rediscovering Workflow Models from Event-Based Data Using Little Thumb, Integrated Computer-Aided Engineering, pp. 151 - 162

[7] A.J.M.M. Weijters and W.M.P. van der Aalst (2006) - Process Mining with the Heuristics Miner Algorithm, Technical Report at Eindhoven University of Technology, pp. 1-34

[8] Wil M.P. van der Aalst, Minseok Song (2004) - Mining Social Networks: Uncovering interaction patterns in business processes, Proceedings of the 2nd International Conference on Business Process Management, Potsdam, pp. 244-260

[9] Maikel Leemans, Wil M. P. van der Aalst (2015) - Process Mining in Software Systems, Discovering Real-Life Business Transactions and Process Models from Distributed Systems, Proceedings of Model Driven Engineering Languages and Systems (MODELS), pp. 44-53

[10] Sander J.J. Leemans, Dirk Fahland, and Wil M.P. van der Aalst (2015) - Scalable Process Discovery with Guarantees, Proceedings of 16th International Conference BPMDS, Stockholm, Sweden, pp. 85 -101

[11] Grefen, P and Hoffner, Y (1999) - CrossFlow: Cross-Organizational Workflow support for virtual organizations, Proceedings of 9th IEEE Int. Workshop on Research Issues in Data Engineering, Sydney, Australia, pp. 90-91

[12] Chen, Q and Hsu, M (2001) - Inter-Enterprise Collaborative Business Process Management, Proceedings of International Conference on Data Engineering, Germany

[13] Van der Aalst, W.M.P (2000) - Process-oriented Architectures for Electronic Commerce and Interorganizational Workflow, Proceedings of Information Systems, pp. 639–671

[14] EDIMine project of Vienna University of Technology and Eindhoven University of Technology http://edimine.ec.tuwien.ac.at

[15]CoSeLoG project of Eindhoven University of Technology http://www.win.tue.nl/coselog/wiki/start

[16] ACSI project http://www.acsi-project.eu/

[17] Hoanga, H and Jungb, J (2013) - Ontology-based approaches for cross-enterprise collaboration: a literature review on semantic business process management, Enterprise Information Systems, Vol. 8, pp. 648-664

[18] Born, M. and C. Drumm, (2007) – SUPER, Raising Business Process Management Back to the Business Level, ERCIM Special Theme on Service-Oriented Computing, pp. 43–44.
[19] Yan, Z., E. Cimpian, M. Zaremba, and M. Mazzara (2007) - BPMO: Semantic Business Process Modeling and WSMO Extension, Proceedings of IEEE International Conference on Web Services, pp. 1185–1186
[20] Kirsten Lenz, Andreas Oberweis (2003) - Inter-organizational Business Process Management with XML Nets, Petri Net Technology for Communication-Based Systems, pp. 243-263
[21] W. van der Aalst, M. Dumas, A. R. C. Ouyang, and H. Verbeek. (2008) - Conformance checking of service behavior, ACM Transactions on Internet Technology, pp. 29–59
[22] W. van der Aalst and H. Verbeek (2008) - Process Mining in Web Services: The WebSphere Case, IEEE Bulletin of the Technical Committee on Data Engineering, pp. 45–48
[23] H. R. M. Nezhad, R. Saint-Paul, B. Benatallah, and F. Casati (2008) - Deriving Protocol Models from Imperfect Service Conversation Logs, IEEE Transactions on Knowledge and Data Engineering, pp. 1683–1698
[24] H. R. M. Nezhad, R. Saint-Paul, F. Casati, and B. Benatallah (2011) - Event Correlation for Process Discovery from Web Service Interaction Logs, VLDB Journal, pp. 17–444
[25] Robert Engel, Wil van der Aalst, Marco Zapletal, Christian Pichler, Hannes Werthner (2012) - Mining Inter-Organizational Business Process Models from EDI Messages: A Case Study from the Automotive Sector, Proceedings of 24th International Conference, CAiSE 2012, pp. 222-237
[26] Remco Dijkman, Marlon Dumas, Luciano Garcia-Banuelos (2009) - Graph Matching Algorithms for Business Process Model Similarity Search, Proceedings of 7th International Conference, pp. 48-63
[27] M. Neuhaus, K. Riesen, and H. Bunke (2006) - Fast suboptimal algorithms for the computation of graph edit distance, Lecture Notes in Computer Science, pp. 163–172
[28] S. Berretti, A. Del Bimbo, and P. Pala (2004) - A graph edit distance based on node merging, Proceeding of ACM International Conference on Image and Video Retrieval (CIVR), pp. 464–472
[29] R. Ambauen, S. Fischer, and H. Bunke (2003) - Graph edit distance with node splitting and merging, and its application to diatom idenfication, GbRPR, pp. 95–106
[30] Mikhail Zaslavskiy, Francis Bach, Jean-Philippe Vert (2010) - Many-to-Many Graph Matching: A Continuous Relaxation Approach, Proceedings of European Conference, ECML PKDD, pp. 515 – 530
[31] T. Caelli and S. Kosinov (2004) – A projection clustering method for inexact graph matching, IEEE Transaction Pattern Anal.Mach. Intell., pp. 515–519
[32] Y. Keselman, A. Shokoufandeh, M. F. Demirci, and S. Dickinson (2003) - Many to many graph matching via metric embedding, CVPR, pp. 850-857
[33] Thomas. R. Gruber (1995) - Towards Principles for the Design of Ontologies Used for Knowledge Sharing, International Journal of Human-Computer Studies, Volume 43, Issues 5–6, pp. 907-928
[34] Nicola Guarino (1998) - Formal ontology and information systems, Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, pp. 3–15
[35] Yannis Kalfoglou, Marco Schorlemmer (2003) – Ontology mapping: the state of the art, The Knowledge Engineering Review, Volume 18, Issue 01, pp. 1-31

[36] Branimir Wetzstein, Zhilei Ma, Agata Filipowska, Monika Kaczmarek, Sami Bhiri, Silvestre Losada (2007) – Semantic Business Process Management: A Lifecycle Based Requirements Analysis, Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management, pp. 1-12
[37] Marcello La Rosa, Marlon Dumas, Reina K, Remco Dijkma (2013) - Merging Business Process Model, ACM Transactions on Software Engineering and Methodology, Volume 22, Issue 2, pp. 96-113
[38] F. Gottschalk, W. M. P. van der Aalst, and M. H. Jansen-Vullers (2008) - Merging event driven process chains, Confederated International Conferences On the Move to Meaningful Internet Systems, pp. 418–426
[39] C. Li, M. Reichert, A. Wombacher (2009) - Discovering reference models by mining process variants using a heuristic approach, Proceedings of the 7th International Conference on Business Process Management, pp. 344–362
[40] S. Sun, A. Kumar, and J. Yen. Merging workflows (2006) - A new perspective on connecting business processes, Decision Support Systems, pp. 844–858
[41] J.M. Kuster, C. Gerth, A. Forster, G. Engels (2008) - A tool for process merging in business-driven development, CEUR Workshop Proceedings, pp. 89–92
[42] Wassim Derguech, Sami Bihri (2011) – Merging business process variants, Business Process Management Workshops, pp. 119-130
[43] Nesamoney, D. (2004) - BAM: event-driven business intelligence for the real-time enterprise, DM Review, Vol. 14 No. 3, pp. 38-40.
[44] Costello, C. and Molloy, O. (2008) - Towards a semantic framework for business activity monitoring and management, in AAAI Spring Symposium: AI Meets Business Rules and Process Management, Stanford, CA, pp. 1-11
[45] Etzion, O. and Niblett, P. (2010) - Event Processing in Action, Manning Books Publications, Cincinnati, OH
[46] Luckham, D. (2002) - The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems, Addison-Wesley Book, Boston, MA.
[47] Hackathorn, R. (2002) - Current Practices in Active Data Warehousing
[48] White, C. (2003) - BAM and Business Intelligence, DM Review Magazine, 2003
[49] Sola Adesola Tim Baines (2005)-Developing and evaluating a methodology for business process improvement, Business Process Management Journal, Vol. 11, Issue 1 pp. 37 - 46
[50] Reichert, M., Rinderle, S., Kreher, U. and Dadam, P. (2005) - Adaptive process management with ADEPT2, Proceedings of 21st International Conference on Data Engineering (ICDE'05), pp.1113–1114
[51] ProM Tools - http://www.promtools.org/
[52] SEMILAR, Semantic Similarity Toolkit - http://www.semanticsimilarity.org/
[53] OWL API - http://owlapi.sourceforge.net/
[54] JérômeEuzenat, David Loup, Mohamed Touzani, PetkoValtchev (2004) - Ontology Alignment with OLA, Proceedings of the 3rd EON Workshop, 3rd Intl. Semantic Web Conference, Hiroshima, pp. 59-68
[55] JGraphT - http://jgrapht.org/
[56] XES Standard - http://www.xes-standard.org/openxes/start
[57] Tomas Skersys, Rimantas Butleris, Kestutis Kapocius, Tomas Vileiniskis (2013) - An Approach for Extracting Business Vocabularies from Business Process Models in Information technology and control, pp. 178-190
[58] L. Nemuraite, T. Skersys, A. Sukys (2010) – VETIS tool for editing and transforming SBVR business vocabularies and business rules into UML&OCL models, 16th International Conference on Information and Software Technologies, pp. 377-384

[59] Karpovič, Jaroslav Nemuraitė, Lina (2011) – Transforming SBVR business semantics into Web ontology language OWL2: main concepts, Proceedings of the 17th international conference on Information and Software Technologies, pp. 231-238
[60] Jérôme Euzenat, David Loup, Mohamed Touzani, Petko Valtchev (2004) - Ontology Alignment with OLA, Proceedings of the 3rd EON Workshop, 3rd Intl. Semantic Web Conference, Hiroshima, pp. 59-68
[61] Natalia F. Noy (2004) - Semantic integration: a survey of ontology-based approaches, ACM SIGMOD Record: Special section on semantic integration, pp.65-70
[62] S. Castano, V. De Antonellis, S. De Capitani di Vimercati (2001) - Global viewing of heterogeneous data sources, IEEE Transactions on Knowledge and Data Engineering, pp. 277-297
[63] H. H. Do and E. Rahm, COMA (2001) – A system for flexible combination of schema matching approaches, Proceedings of the Very Large Data Bases Conference (VLDB), pp. 610-621
[64] Noy, N.F. and Musen, M.A. (2000) - PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment, Proceedings of the Seventeenth National Conference on Artificial Intelligence, Austin, TX, pp. 450-455
[65] Marc Ehrig, Steffen Staab (2004) - QOM: Quick ontology mapping, Proceedings of the International Semantic Web Conference (ISWC), pp. 683-697
[66] F. Giunchiglia, P. Shvaiko, and M. Yatskevich (2005) - Semantic schema matching, Technical Report DIT-05-014, University of Trento, pp. 61-75
[67] Jérôme David, Jérôme Euzenat, François Scharffe, Cássia Trojahn dos Santos (2011) - The Alignment API 4.0, Semantic web journal, pp. 3-10
[68] J. Sowa (1996) - Processes and participants, Proceedings of the 4th International Conference on Conceptual Structures, pp. 1-22
[69] G. Alonso, SU. Fiedler, C. Hagen, A. Lazcano, H.Schuldt, and N. Weiler (1999) - Wise: Business to business e-commerce, Proceedings of the 9th International Workshop on Research Issues on Data Engineering, pp. 132–139
[70] A. Lazcano, H. Schuldt, G. Alonso, and H Schek (2001) - WISE: Process Based E-Commerce, IEEE Data Engineering Bulletin, Vol. 24, No. 1, pp. 46–51
[71] Norta, A., & Grefen, P. (2007) - Discovering patterns for inter-organizational business process collaboration, International Journal of Cooperative Information Systems, pp. 507-544
[72] Gong, R., Li, Q., Ning, K., Chen, Y., & O'Sullivan, D. (2006) - Business process collaboration using semantic interoperability: Review and framework, Semantic Web–ASWC, pp. 191-204
[73] Maria Laura Sebu, Horia Ciocarlie – Applied process mining in software development (2014) – Proceedings of SACI, 9th IEEE International Symposium on Applied Computational Intelligence and Informatics
[74] Maria Laura Sebu, Horia Ciocarlie - Cross-organizational compatibility detection with process mining (2015) - Buletinul Institutului Politehnic din Iasi, Universitatea Tehnică „Gheorghe Asachi" din Iaşi, Tomul LVIII (LXII), Fasc. 1
[75] Maria Laura Sebu, Horia Ciocarlie (2016) - Collaborative Business Process Solution Considering an Ontological Dimension of Process Models; Proceedings of SOFA, 7th IEEE International Workshop on Soft Computing Applications IEEE
[76] Maria Laura Sebu, Horia Ciocarlie - Business process similarity metric supporting one-to-many relationship (2014) – Proceedings of SACI 10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics

[77] Maria Laura Sebu, Horia Ciocarlie (2016) - Similarity of business process models in a modular design, Proceedings of SACI 11th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics
[78] Maria Laura Sebu, Horia Ciocarlie (2015) - Merging Business Processes for a Common Workflow in an Organizational Collaborative Scenario, Proceedings of ICSTCC 19th International Conference on System Theory, Control and Computing
[79] Maria Laura Sebu, Horia Ciocarlie (2015) - Business activity monitoring solution to detect deviations in business process execution, Proceedings of SACI, 10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics
[80] Maria Laura Sebu, Horia Ciocarlie (2014) - Process mining functional and structural validation, Proceedings of SOFA, 6th IEEE International Workshop on Soft Computing Applications IEEE
[81] Maria Laura Sebu, Horia Ciocarlie (2016) - Optimizing business processes by learning from monitoring results, Proceedings of SACI 11th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics
[82] Zhiping Zeng, Anthony K.H. Tung, Jianyong Wang, Jianhua Feng, Lizhu Zhou (2009) - Comparing stars: on approximating graph edit distance, Proceedings of the VLDB Endowment, pp. 25-36