

# Prominence of Rule Engine in Business Process Model for business tumour and Data Analytics

Patel Palak

*Department of Computer Application  
Bholabhai Patel College of Computer Studies , Gandhinagar , KSV University*

Patel Bhavna

*Department of Computer Application  
Bholabhai Patel College of Computer Studies , Gandhinagar , KSV University*

Dr. Abhijeetsinh Jadeja

*Department of Computer Application  
Bholabhai Patel College of Computer Studies , Gandhinagar , KSV University*

**Abstract - In this paper we are focus on Business Big Data and it's problems. When we focus on business data there is large volume of data and to handle large volume of data there are already so many algorithm are already available. But on other end Business Data Process (BDP) is one of the emerging part to knob large volume of data with proper architecture. In this paper main emphasis is on existing BDP models and it's comparison as literature review we proposed architecture for better –faster – multiple platform BDP process. The proposed model explain various Rule Engine for numerous layers. The proposed model layers elucidate all the various need and process for large volume of data when we communicated about large business process. Rule Engine is a new era concept for handle not only big data but in future we can use is as Data Analytic for getting expedient benefit in every business process. It kinds current system more feasible in every juncture of business. Rules architecture for business process explained on web Service Layer(SL) – Business Process Layer (BPL) and finally then Rules Layer (RL) in the impartial is that when we change in the Rule Layer and it's database it reflects to User / Client end and without presence database side we can immediate implement consequence in our Business Process very smoothly. Finally the proposed model effect is on process centric and technical requirement and server support on any less or normal configure Web Server.**

**Key Word : Business Big Data , Business Data Process ,Data Analytic, Rule Engine , Service Layer , Business Process Layer , Rule Layer , Web Service , Web Server.**

## I. CONCEPT OF BPM

### 1.1 Introduction of Rule engine in BPM

A business rules engine is a software system that executes one or more business rules in a runtime production environment.[1] The rules might come from legal regulation ("An employee can be fired for any reason or no reason but not for an illegal reason"), company policy ("All customers that spend more than \$100 at one time will receive a 10% discount"), or other sources.

Rule engine software is commonly provided as a component of a business rule management system which, among other functions, provides the ability to: register, define, classify, and manage all the rules, verify consistency of rules definitions ("Gold-level customers are eligible for free shipping when order quantity > 10" and "maximum order quantity for Gold-level customers = 10"), define the relationships between different rules, and relate some of these rules to IT applications that are affected or need to enforce one or more of the rules.[2] Then conflict resolution is provided by the agenda. It arranges the order of actions, which has been selected to be run.[3] The agenda is an important part of a rule engine, because the execution of an action may invalidate a rule and in consequence cancel another action. To sum it up, a rule engine can be presented as follows:

A rule engine perfectly fits to:

- calculate a discount of orders based on the amount or price
- determine an insurance policy dependent on the age of the driver and the value of the car
- make a diagnosis
- detect imminence of failure
- solve riddles

1.2 Application layer and Rules Layer of BPM

The Web services layer exposes the existing application layer functionality as services.[4] Multiple business processes can then reuse these services, thereby fulfilling the promise of a service-oriented architecture (SOA). Web services implement functional and domain logic. Functional methods are typically stateless and medium-grained. Web services may for example contain utility methods, entity operations, and inquiry methods for system data. You can implement Web services using multiple technologies and hide differences among implementation platforms.[5] Thus, this layer is well suited for:

- Implementing medium-grained methods for a particular entity/domain area
  - Integrating legacy code/third-party party tools
  - Encapsulating logic, custom code, and implementation from the application layer
- Rules Layer.*

The rule engine is typically the home for complex logic that involves a number of interdependencies between entities and order-dependent logic calculation. Extracting business rules as a separate entity from business process leads to better decoupling of the system, which, in consequence, increases maintainability.[6] Rules engines allow for evaluation of rules sets in parallel and in a sequential order. In addition, rules have the ability to evaluate the values of input and intermediate data and determine if a rule should be fired.[7] This modular design provides a simpler and more maintainable solution than traditional Java procedural code. Furthermore, as I mentioned previously, rules are declarative and allow high-level GUI editing by business analysts. Modern rule engines execute extremely quickly and provide built-in audit logging. The typical traits of a rules layer are

- Contains coupled and complex logic
- Supports efficient business logic evaluation using parallel execution
- Contains complex return structure built from multiple business rule evaluations
- Allows for translation of domain logic into simple rules
- Implements highly volatile business policy

Because rules are exposed as services in the Web services layer, they can be reused across all inter-enterprise application, making the development of new applications and integrations easier

1.3 Rule engine architecture in BPM

Ernest Friedman-Hill, creator of Jess, has suggested the following general architecture for any rules engine, both for business rules applications and rules-based systems:

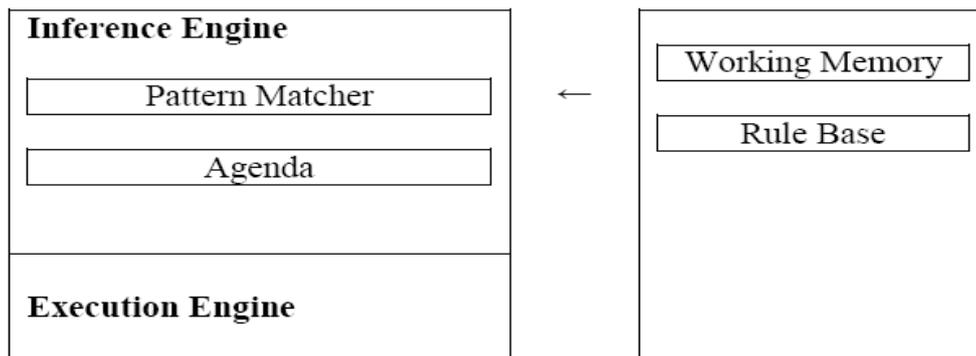


Figure 2.3: General Architecture of Rule Engine

**Working Memory:** The data that rules work on. Also called the fact base.

**Rule Base:** The list of all possible rules. They may be stored as text strings, as objects in a database, or may be compiled in some way (by a rules compiler). A popular form for compiled rules is a Rete network.[8]

**Pattern Matcher:** Determines which rules to apply to the data in working memory. Generally, this is often the slowest part of the rules engine, if there are thousands of facts and thousands of rules. Working Memory and Rule Base are the inputs for the Pattern Matcher.

Agenda: Determines the order of execution of rules. In many rules engines, the agenda is recreated after each rule execution, in case the optimal ordering of the rules changes because of the previous rule execution.

Execution Engine: The part of the rules engine that is responsible for applying rules to data and then performing the action part of the rules that have fired. The range of possible actions is different in different rules engines.[9]

## II. REVIEW OF LITERATURE

### 2.1 A Glance of Centric Technology of BPM

Viewed from a purely technical perspective, BPM is a merging of a number of existing technologies and approaches. Its primary roots are in the process management capabilities of work flow tools but it also includes capabilities that derive from process modelling, application integration, process analytic, rules management, and collaboration portals. However, a BPM suite is not just a sum of these parts.[10] It brings together all these technology elements into a single platform that manages the life cycle of a process starting from definition, through deployment, execution, measurement, change, and re-deployment.[11]

More significantly, it involves a fundamental change in the way that we think about the structure of IT systems, applications, and infrastructure. BPM promotes a process-centric view of IT where the management of end-to-end processes is separated from the underlying applications, their connections, and the data.[12] It involves the creation of an independent process layer. This layer contains a complete view of all the activities necessary to execute a particular business process and it can manage the flow of these activities whether they involve different applications, people, or a combination of both. This independent process layer complements both existing and future investments in applications, content repositories, and integration tools.

There are four critical components of a BPM Suite:

- Process Engine – a robust platform for modeling and executing process-based applications, including business rules
- Business Analytic — enable managers to identify business issues, trends, and opportunities with reports and dashboards and react accordingly
- Content Management — provides a system for storing and securing electronic documents, images, and other files
- Collaboration Tools — remove intra- and interdepartmental communication barriers through discussion forums, dynamic workspaces, and message boards

BPM also addresses many of the critical IT issues underpinning these business drivers, including:

- Managing end-to-end, customer-facing processes
- Consolidating data and increasing visibility into and access to associated data and information
- Increasing the flexibility and functionality of current infrastructure and data
- Integrating with existing systems and leveraging emerging service-oriented architecture (SOAs)
- Establishing a common language for business-IT alignment

### 2.2 Functional Elements process in BPM

BPM suite needs to have the following six functional elements to provide a comprehensive independent process layer.

- Defining Your Processes: Modelling and Simulation –

It is important that business users and IT collaboratively engage in process definition. Look for a graphical modelling and simulation environment where business users can define, refine, and change processes and IT can deploy processes all from one location. This ensures that the processes implemented meet organizational goals.[13]

- Defining Your Business Rules: Business Rule Engine –

The rules governing business processes tend to change more frequently than the processes themselves. Look for a business rule engine that extracts business rules from the process, improving process flexibility. Flexibility is further increased if you can put complete control of business rules in the hands of business people, enabling them to make changes to rules or create new rules as their needs dictate, rather than as the availability of IT resources permits.[14]

- Integrating People, Processes, and Applications: an SOA Foundation –

Service oriented architecture (SOA) ensures long-term BPM flexibility. By deploying BPM on an SOA foundation, organizations are able to transform monolithic CRM, ERP, and legacy applications into discrete reusable services, which then serve as building blocks that can be combined, organized, and orchestrated to support complex business processes.[15] Changes to services can be made independently of processes and vice versa.

• Running Your Processes: Execution Engine –

The heart of a BPM suite is the process engine that executes the business process as you have defined it. Look for an execution engine that tracks the state of the process at any given time and ensures that the correct sequence of process steps is followed as defined by the business. It should be able to handle high volume, mission critical processes, including support for in-flight process changes, ad hoc routing, and more.

- • Managing Your Processes: Real-time Monitoring and Process Optimization –
- Real-time monitoring makes it possible to address problems as they arise, or even before, and that systems involved in executing processes are sound and functioning. The last component required for effective process management is analytic, enabling users to slice and dice past process performance data, providing a complete picture of the process and how to optimize it.[16]
- • Connecting Users to Processes: Presentation and Collaboration –
- People play a vital role in most business processes. A BPM suite should provide a rich environment for people to play their part in the process and efficiently complete necessary work items. Rich extensible Web 2.0 interfaces enable organizations to manage all touch points of the process from anywhere at any time and enable unprecedented productivity.
- Using the above functional elements the underneath functionality can be added to the BPM
- BPM also includes capabilities derived from process modelling, application integration, process analytic, rules management, and collaboration portals
- A BPM suite should bring these elements together into a single platform that provides the basis for the independent process layer.
- Handle a wide range of different processes
- Manage very complex processes
- Scale to thousands of different processes and millions of different process instances
- Enable the rapid creation and deployment of new processes
- Allow a non-technical business user to rapidly define and deploy a process change
- Allow true 24x7 operations.

*2.3 Rules Architecture for Business process*

Integrating a rules engine within a process management framework requires some investment up front. Before you attempt this integration, it is important to delineate rules from process. Hence, a major decision in system architecture is how to implement business policies, business processes, and supporting business logic. Indeed, the architect must communicate or invent best practices so that designers know where each of the relevant technologies—BPEL, business rules, Java/Web services—should be applied when designing system functionality.[17]

Figure: business logic is spread across three different layers of the IT infrastructure: business process, Web services, and rules. Let's address each in turn.

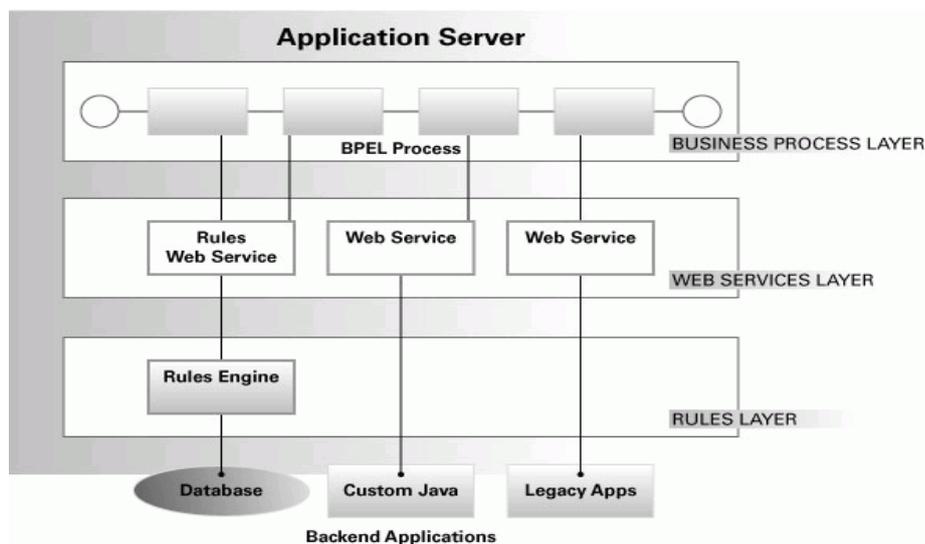


Figure 2.2: Architecture: separating rules from processes

### III. NEED OF RESEARCH

#### 3.1 Why required Multilayer Rule Engine Architecture

- 1) Working for different department in a business organization  
Various departments of business organization can get each and every information of another department through the use of rule engine architecture. Proposed Multilayer Rule Engine architecture work for different department in any organization for ex: an organization contain sales department, purchase department, store department, etc.... then any department can access different department knowledge base.
- 2) To fulfil the need of all types of business application business system must be independent of any platform. [18] This rule engine architecture will be furnished all types of needs of business application and this platform will be independent hence you can access your application anywhere
- 3) Sometimes rules of the application are getting changed than it is difficult to maintain application so that we use rule engine architecture.
- 4) It is used to improve business process management after being separate business logic and application code.
- 5) The application will get direct connection with database and if user will fetch rules directly from database than it can create some problems.
- 6) Need for a model for any kind of organization from small scale to large scale irrespective of large geographical location. This architecture is useful for small business to large business like inventory shop to big financial application. so that proposed multilayer Rule engine architecture is useful for small scale business to large scale business.
- 7) The commercial rule engine which is already used it demands for more investment and also take more time.
- 8) Using Rule engine architecture in business process we can satisfy customer care service and we can reduce cost. And we can launch new product in market.
- 9) Rules are always written in if/else/then statement. If there are more if/else statements in the application and, user wants to change those rules and redeploy your application again then it is very difficult.

#### 3.2 Propose Model of Multilayer Rule Engine

The proposed Multilayer engine architecture can be used to architect automated different departmental activity in an overall business enterprise system. Above model can be applied to any kind of business sector for instance production, health care, telecommunication, investment sector and service oriented industries, etc.... (Hotel industry, Bank, etc.), that needs to implement a set of business rule as an enterprise service. The proposed architecture is a User centric flexible Multilayer Rule Engine architecture, which can be implemented for any distributed units of any Business Organization. The architecture describes the Three Layers as underneath:

### Multilayer Rule Engine Architecture

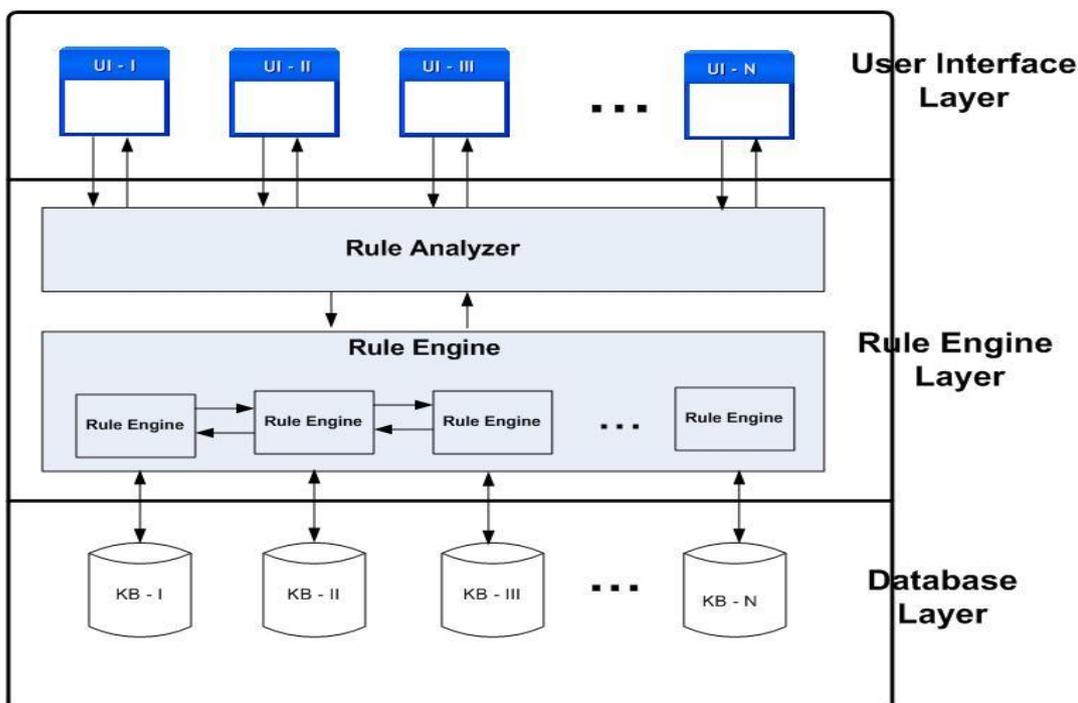


Figure 3: Proposed Multilayer Rule Engine Architecture

#### User Interface Layer:

User Interface is can any business application and can use any Business Stock holder. Comprises of user interface within the business environment that is accessibility to various stack holder like

- Top level stack holder (top management)
- Middle level stack holder (operational manager, production manager, finance manager, marketing manager, sales manager)
- Lower level stack holder (supervisor, clerk, helper staff, etc...)
- Customer

The technology use for the User Interface Layer can be any web based tool like java, .Net, C#.net etc...

#### Rule Engine Layer :

Here, Rule Engine layer is divided into two parts one is Rule Analyzer and second is Rule Engine. In a Rule Engine there are other small rule engines doing particular Departmental work. Here, in the Rule engine layer rule engine uses the Pattern Matching Algorithm like Rete Algorithm, Rete I Algorithm, Rete II Algorithm, Leap Algorithm. Rule engine fetches the rules from Knowledge Base and use pattern matching algorithm for matching rules. Here, work of Rule Analyser is to fetch proper rules from rule engine and provide it to the user. Using those rules user can take decision. When user demands for new rule to the Rule engine layer then Rule Analyser checks whether the rule is existing or not. If rule does not exist then Rule Analyser adds new rule to the Knowledge Base. For adding new rule to the Knowledge base Rule Analyser use FTL (Free marker Template Language) for Java application and use MVC 2 Template for .Net application. Business Analyst use Rule Analyser for modifying existing rules in Knowledge Base.

Here, there are some steps how to convert User interface value into Rule Language using free marker Template Language:

- Read value from the User Interface.
- Generate the HASH Map using User Interface value.
- Then after will be access to FTL API.
- Then HASH map and FTL file will be sending to the FTL Service.
- Then API will return string Object.
- Then those strings will be written in template file.

If your application written in .Net Language then you can use MVC 2 template language which converts User Interface value into Rule Language Template.

#### *Database Layer:*

Rule Engine can work with any type of Database. Here, Database layer contain different Knowledge Base of different departments. Database layer stores business logic in the form of Rules and those rules can be made by any Rule Language. There are different Rule Language are available like XML Rule Language, Excel files which contains Decision tables, SWRL ( Semantic Web Rule Language), NRL( the Natural Rule Language), BRML(Business Rule Mark-up Language), OWL(Ontology Web Language), DSL( Domain Specific Language) etc. Rule Engine of different department can access knowledge base of any department.

### IV. RESULT AND CONCLUSION

#### *4.1 Comparative Study of Propose Model with Existing Rule Engine*

The proposed Research work is on developing Rule engine architecture for enhancing Business Process management and the study of the Application of the proposed architecture in different areas of Business Process where Multilayer Rule Engine architecture can be used. I got the following comparative results between proposed Rule engine architecture and other.

Comparative Study	Proposed Multilayer Rule Engine	Drools Rule Engine	JBoss Rules Rule Engine	Biztalk Rule Engine
Complexity	Less	High	High	High
Technical Requirements/server Support	Less/ Normal Web Server	JBoss Server	JBoss Server	Biztalk Server 4.0
User / Process Centric	User and Process Centric	Process Centric	Process Centric	Process Centric
Web Tools Support	Java, .Net	Java	Java	Java
Database	Any	Any	Any	Any

Table 4.1 Comparative Study between proposed Multilayer Rule Engine and others Rule Engine Architecture

Comparatively this proposed Rule Engine Architecture is very much easy to implement than others. This architecture supports all web based tools like Java and .Net where others just support Java. This architecture requires less technical requirements than others. Proposed Rule Engine architecture uses the normal web server whereas JBoss Rules and Drools Rule engine Architecture uses JBoss Server and Biztalk Rule Engine Architecture use Biztalk server 4.0. Proposed Multilayer Rule Engine Architecture support any kind of Databases. The proposed Multilayer Rule Engine is User centric and Process centric whereas others are Process centric only.

#### *4.2 Result and Conclusion*

In this Paper I have presented a strategy for separating Business logic from Application logic and proposed Multilayer Rule Engine Architecture which is divided in to three layers: User Interface Layer, Rule Engine Layer, and Database Layer. You can utilize this technology to improve Business Process management. Using Rule Engine with Business Process management can help to Decision Support is the main goal of Multilayer Rule Engine Architecture by automating and standardizing complex Rule based Decision. It allows business rules to be defined, tested, applied and administrated directly by the business user. It allows business user to

check rules for accuracy and completeness, ensuring that every possible scenario has one response and the response will be always same because rules are defined and tested by the business user, all business rules reflect business logic.

Finally, by making use of proposed Multilayer Rule Engine Architecture you can enhance your Business Process Management. As this Rule engine architecture is user centric the business application and user remains in good touch. So that user can get the information easily and fast.

If this architecture will be implemented great benefits can be achieved such as:

- One can provide good service to the customer.
- Decision can be take on the spot.
- The application remains in control even if the rules change frequently in it.

The objective of my research is to enhance Business Process Management. This Multilayer Rule Engine Architecture helps to achieve this objective in better way.

#### REFERENCES

- [1] T. M. Mitchell, Machine Learning. MIT Press and The McGraw-Hill companies, Inc., 1997
- [2] A. Lindsay, D. Dawns, and K. Lunn, "Business processes – attempts to find a definition," *Information and Software Technology*, vol. 45, no. 15, pp. 1015–1019, December 2003, Elsevier
- [3] OMG, "Business Process Model and Notation (BPMN): Version 2.0 specification," Object Management Group, Tech. Rep. formal/2011-01-03, January 2011.
- [4] I. S. Bajwa, M. G. Lee, and B. Bordbar, "SBVR Business Rules Generation from Natural Language Specification," in AAAI Spring Symposium: AI for Business Agility. AAAI, 2011. [Online]. Available: <http://www.aaai.org/Library/Symposia/Spring/ss11-03.php>
- [5] A. Sinha and A. Paradkar, "Use cases to process specifications in business process modeling notation," in *Web Services (ICWS)*, 2010 IEEE International Conference on, 2010, pp. 473–480. KRZYSZTOF KLUZA, GRZEGORZ J. NALEPA: TOWARDS RULE-ORIENTED BUSINESS PROCESS MODEL GENERATION 945
- [6] G. J. Nalepa, K. Kluza, and S. Ernst, "Modeling and analysis of business processes with business rules," in *Business Process Modeling: Software Engineering, Analysis and Applications*, ser. Business Issues, Competition and Entrepreneurship, J. Beckmann, Ed. Nova Science Publishers, 2011, pp. 135–156.
- [7] A. Lig eza and G. J. Nalepa, "A study of methodological issues in design and development of rule-based systems: proposal of a new approach," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 117–137, 2011.
- [8] B. Silver, *BPMN Method and Style*. Cody-Cassidy Press, 2009. W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, 1st ed. Springer Publishing Company, Incorporated, 2011.
- [9] K. Kluza, K. Kaczor, and G. J. Nalepa, "Enriching business processes with rules using the Oryx BPMN editor," in *Artificial Intelligence and Soft Computing: 11th International Conference, ICAISC 2012: Zakopane, Poland, April 29–May 3, 2012*, ser. Lecture Notes in Artificial Intelligence, L. Rutkowski and [et al.], Eds., vol. 7268. Springer, 2012, pp. 573–581. [Online]. Available: <http://www.springerlink.com/content/u654r0m56882np77/>
- [10] J. R. Nawrocki, T. Nedza, M. Ochodek, and L. Olek, "Describing business processes with use cases," in *BIS*, 2006, pp. 13–27.
- [11] D. Lubke, K. Schneider, and M. Weidlich, "Visualizing use case sets as bpmn processes," in *Requirements Engineering Visualization*, 2008. REV '08., 2008, pp. 21–25.
- [12] M. Atzmueller and G. J. Nalepa, "A textual subgroup mining approach for rapid ARD+ model capture," in *FLAIRS-22: Proceedings of the twenty-second international Florida Artificial Intelligence Research Society conference: 19–21 May 2009, Sanibel Island, Florida, USA*, H. C. Lane and H. W. Guesgen, Eds., FLAIRS. Menlo Park, California: AAAI Press, 2009, pp. 414–415, to be published.
- [13] W. van der Aalst, "On the automatic generation of workflow processes based on product structures," *Computers in Industry*, vol. 39, no. 2, pp. 97–111, 1999.
- [14] Claire Costello, "Orchestrating Supply chain Interactions using Emerging Process Description Languages and Business Rules".
- [15] Anca Andreescu, "A General Software Development Process Suitable for Explicit Manipulation of Business Rules".
- [16] Milan Milanović, "Modeling Service Orchestrations with a Rule-enhanced Business Process Language"
- [17] Olegas Vasilecas, "Ensuring Consistency of Information Systems Rules Models".
- [18] Anis Charfi, "Hybrid Web Service Composition: Business Processes Meet Business Rules".