Conversion of SBVR Behavioral Business Rules to UML Diagrams: Initial Study of Automation

Paweł Kaliniak

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden
This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.
Contact Information:
Author(s): Paweł Tadeusz Kaliniak
E-mail: pawel.kaliniak@gmail.com

University advisor:
PhD Ludwik Kuźniarz
Blekinge Institute of Technology, Sweden

External university advisor:
PhD Bogumiła Hnatkowska
Wrocław University of Technology, Poland

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden
Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
Abstract

**Context.** Automation of conversion of business rules into source code in software development project can reduce time and effort in phase of development. In this thesis we discuss automatic conversion of behavioral business rules defined in Semantics of Business Vocabulary and Rules (SBVR) standard, to fragments of Unified Modeling Language diagrams: activity, sequence and state machine. It is conversion from Computation Independent Model (CIM) into Platform Independent Model (PIM) levels defined by Model Driven Architecture (MDA). PIM in MDA can be further transformed into Platform Specific Model which is prepared for source code generation.

**Objectives.** Aim of this thesis is to initially explore field of automatic conversion of behavioral business rules - conversion from SBVR representation to fragments of UML diagrams. It is done by fulfilment of objectives defined as following:
- To find out properties of SBVR behavioral rule which ensure that the rule can be automatically converted to parts of UML behavioral diagrams (activity, sequence, state machine).
- To propose mapping of SBVR constructs to constructs of UML behavioral diagrams.
- To prepare guidelines which help to specify SBVR behavioral business rules in such way that they can be automatically transformed into fragments of selected UML behavioral diagrams.

**Methods.** Expert opinion and case study were applied. Business analysts from industry and academia were asked to convert set of SBVR behavioral business rules to UML behavioral diagrams: activity, sequence and state machine. Analysis of the set of business rules and their conversions to UML diagrams was basis for fulfilment of objectives.

**Results.** 2 syntax and 3 semantic properties were defined. Conversion rules which define mapping for SBVR behavioral business rules constructs to UML constructs were defined: 5 rules for conversion to activity diagram, 6 for conversion to sequence diagram, 5 for conversion to state machine diagram. 6 guidelines which are intended to help in specification of behavioral business rules that can be automatically transformed into UML diagrams according to the presented conversion rules, were defined.

**Conclusions.** Research performed in this thesis is initial study of automatic conversion of behavioral business rules from SBVR notation to UML behavioral diagrams notation. Validation of defined properties, conversion rules and guidelines can be done in industry as future work. Re-execution of research for larger and more diverse set/sets of behavioral business rules taken from industry projects, sufficiently broad access to business analysts from industry and academia could help to improve results.

**Keywords:** Behavioral business rules, SBVR, SBVR to UML, automatic conversion, UML behavioral diagrams.
Acknowledgements

I would like to express my gratitude to Triune God, who was helping me during whole time of studies. I thank my both supervisors: PhD Ludwik Kuźniarz and PhD Bogumila Hnatkowska who supported me in the research by giving valuable advices and comments. I am glad that I could learn a lot from you. Furthermore I would like to thank experts in the field of business analysis who took part in my research: Bogumila Hnatkowska, Rafał Krzemiński, Anita Walkowiak and Hanna Wesolowska.

I am also very grateful to my family and fiancée Maria Obuchowicz who were supporting and motivating me from the very beginning. Thank you my brother Piotr Kaliniak for good advice in choosing area of the research and you my friend Alex Bramah-Lawani for advices in an initial stage of the research.

Last but not least, I would like to thank Capgemini Poland for giving me an access to project ”Direct Delivery” which was used in this master thesis.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>ix</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Abbreviations</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Objectives</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Research questions</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Thesis structure</td>
<td>4</td>
</tr>
<tr>
<td><strong>2 Related Work</strong></td>
<td>6</td>
</tr>
<tr>
<td>2.1 Formal specification of SBVR</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Business Rules Manifesto</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Transformation of SBVR business design to UML models</td>
<td>9</td>
</tr>
<tr>
<td><strong>3 Research Method</strong></td>
<td>12</td>
</tr>
<tr>
<td>3.1 Description of methods</td>
<td>12</td>
</tr>
<tr>
<td>3.2 Case study definition</td>
<td>13</td>
</tr>
<tr>
<td>3.3 Form preparation</td>
<td>14</td>
</tr>
<tr>
<td>3.4 Searching for experts</td>
<td>14</td>
</tr>
<tr>
<td>3.5 Threats to validity</td>
<td>15</td>
</tr>
<tr>
<td>3.5.1 Business rules bias</td>
<td>15</td>
</tr>
<tr>
<td>3.5.2 Business analysts bias</td>
<td>15</td>
</tr>
<tr>
<td>3.5.3 Business analysts not available</td>
<td>15</td>
</tr>
<tr>
<td>3.5.4 Business analysts cannot participate in the whole research</td>
<td>16</td>
</tr>
<tr>
<td>3.5.5 Bias in the form for business analysts</td>
<td>16</td>
</tr>
<tr>
<td>3.6 Method overview</td>
<td>16</td>
</tr>
<tr>
<td><strong>4 Data Collection</strong></td>
<td>18</td>
</tr>
<tr>
<td>4.1 Units of analysis</td>
<td>18</td>
</tr>
<tr>
<td>4.1.1 Business rule 1</td>
<td>18</td>
</tr>
<tr>
<td>4.1.2 Business rule 2</td>
<td>25</td>
</tr>
<tr>
<td>4.1.3 Business rule 3</td>
<td>31</td>
</tr>
<tr>
<td>4.1.4 Business rule 4</td>
<td>38</td>
</tr>
<tr>
<td>4.1.5 Business rule 5</td>
<td>44</td>
</tr>
<tr>
<td>4.1.6 Commonalities - summary</td>
<td>49</td>
</tr>
<tr>
<td><strong>5 Results</strong></td>
<td>51</td>
</tr>
<tr>
<td>5.1 Properties</td>
<td>51</td>
</tr>
<tr>
<td>5.1.1 Syntax properties</td>
<td>52</td>
</tr>
<tr>
<td>5.1.2 Semantic properties</td>
<td>52</td>
</tr>
</tbody>
</table>
5.2 Mapping of SBVR constructs to UML constructs

5.2.1 Domain description

5.2.2 Codomains of functions

5.2.3 Mapping functions

5.2.4 Comparison of M1 and M2

5.2.5 Test examples

5.3 Guidelines

6 Analysis and Discussion

7 Conclusions and Future Work

8 Dictionary

9 Bibliography

10 Appendix A: Form for business analyst

11 Appendix B: UML class diagram extracted from facts

12 Appendix C: Expert 2 Raw data

13 Appendix D: Expert 3 Raw data
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Behavioral Rule Statement and its Kinds</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Kinds of Behavioural Elements of Guidance</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Conversion of BBR by Raj</td>
<td>10</td>
</tr>
<tr>
<td>3.1</td>
<td>Activity diagram of the research method</td>
<td>17</td>
</tr>
<tr>
<td>4.1</td>
<td>Conversion of BBR1 to a part of AD by expert 1</td>
<td>19</td>
</tr>
<tr>
<td>4.2</td>
<td>Conversion of BBR1 to a part of AD by expert 2</td>
<td>19</td>
</tr>
<tr>
<td>4.3</td>
<td>Conversion of BBR1 to a part of AD by expert 3</td>
<td>19</td>
</tr>
<tr>
<td>4.4</td>
<td>Conversion of BBR1 to a part of AD by expert 4 - solution A</td>
<td>20</td>
</tr>
<tr>
<td>4.5</td>
<td>Conversion of BBR1 to a part of AD by expert 4 - solution B</td>
<td>20</td>
</tr>
<tr>
<td>4.6</td>
<td>Conversion of BBR1 to SD by expert 1</td>
<td>21</td>
</tr>
<tr>
<td>4.7</td>
<td>Conversion of BBR1 to SD by expert 2</td>
<td>21</td>
</tr>
<tr>
<td>4.8</td>
<td>Conversion of BBR1 to SD by expert 3</td>
<td>22</td>
</tr>
<tr>
<td>4.9</td>
<td>Conversion of BBR1 to a part of SMD by expert 1</td>
<td>23</td>
</tr>
<tr>
<td>4.10</td>
<td>Conversion of BBR1 to a part of SMD by expert 2</td>
<td>24</td>
</tr>
<tr>
<td>4.11</td>
<td>Conversion of BBR1 to a part of SMD by expert 3</td>
<td>24</td>
</tr>
<tr>
<td>4.12</td>
<td>Conversion of BBR1 to a part of SMD by expert 4</td>
<td>25</td>
</tr>
<tr>
<td>4.13</td>
<td>Conversion of BBR2 to a part of AD by expert 1</td>
<td>26</td>
</tr>
<tr>
<td>4.14</td>
<td>Conversion of BBR2 to a part of AD by expert 2</td>
<td>26</td>
</tr>
<tr>
<td>4.15</td>
<td>Conversion of BBR2 to a part of AD by expert 3</td>
<td>27</td>
</tr>
<tr>
<td>4.16</td>
<td>Conversion of BBR2 to SD by expert 1</td>
<td>28</td>
</tr>
<tr>
<td>4.17</td>
<td>Conversion of BBR2 to SD by expert 2</td>
<td>28</td>
</tr>
<tr>
<td>4.18</td>
<td>Conversion of BBR2 to SD by expert 3</td>
<td>29</td>
</tr>
<tr>
<td>4.19</td>
<td>Conversion of BBR2 to a part of SMD by expert 1</td>
<td>30</td>
</tr>
<tr>
<td>4.20</td>
<td>Conversion of BBR2 to a part of SMD by expert 2</td>
<td>30</td>
</tr>
<tr>
<td>4.21</td>
<td>Conversion of BBR2 to a part of SMD by expert 3</td>
<td>30</td>
</tr>
<tr>
<td>4.22</td>
<td>Conversion of BBR3 to a part of AD by expert 1</td>
<td>31</td>
</tr>
<tr>
<td>4.23</td>
<td>Conversion of BBR3 to a part of AD by expert 2</td>
<td>32</td>
</tr>
<tr>
<td>4.24</td>
<td>Conversion of BBR3 to a part of AD by expert 3</td>
<td>32</td>
</tr>
<tr>
<td>4.25</td>
<td>Conversion of BBR3 to a part of AD by expert 4 - solution A</td>
<td>33</td>
</tr>
<tr>
<td>4.26</td>
<td>Conversion of BBR3 to a part of AD by expert 4 - solution B</td>
<td>33</td>
</tr>
<tr>
<td>4.27</td>
<td>Conversion of BBR3 to SD by expert 1</td>
<td>34</td>
</tr>
<tr>
<td>4.28</td>
<td>Conversion of BBR3 to SD by expert 2</td>
<td>35</td>
</tr>
<tr>
<td>4.29</td>
<td>Conversion of BBR3 to SD by expert 3</td>
<td>35</td>
</tr>
<tr>
<td>4.30</td>
<td>Conversion of BBR3 to a part of SMD by expert 1</td>
<td>36</td>
</tr>
<tr>
<td>4.31</td>
<td>Conversion of BBR3 to a part of SMD by expert 2</td>
<td>36</td>
</tr>
<tr>
<td>4.32</td>
<td>Conversion of BBR3 to a part of SMD by expert 3</td>
<td>37</td>
</tr>
<tr>
<td>4.33</td>
<td>Conversion of BBR3 to a part of SMD by expert 4 (variable life insurance policy)</td>
<td>37</td>
</tr>
<tr>
<td>4.34</td>
<td>Conversion of BBR4 to a part of AD by expert 1</td>
<td>38</td>
</tr>
<tr>
<td>4.35</td>
<td>Conversion of BBR4 to a part of AD by expert 2</td>
<td>39</td>
</tr>
<tr>
<td>4.36</td>
<td>Conversion of BBR4 to a part of AD by expert 3</td>
<td>39</td>
</tr>
</tbody>
</table>
4.37 Conversion of BBR4 to SD by expert 1 ........................................ 40
4.38 Conversion of BBR4 to SD by expert 2 ........................................ 41
4.39 Conversion of BBR4 to SD by expert 3 ........................................ 41
4.40 Conversion of BBR4 to a part of SMD by expert 1 .......................... 42
4.41 Conversion of BBR4 to a part of SMD by expert 2 .......................... 43
4.42 Conversion of BBR4 to a part of SMD by expert 3 .......................... 43
4.43 Conversion of BBR4 to a part of AD by expert 1 ........................... 44
4.44 Conversion of BBR4 to AD by expert 2 ....................................... 45
4.45 Conversion of BBR4 to a part of AD by expert 3 ........................... 45
4.46 Conversion of BBR5 to SD by expert 1 ....................................... 46
4.47 Conversion of BBR5 to SD by expert 2 ....................................... 47
4.48 Conversion of BBR5 to SD by expert 3 ....................................... 47
4.49 Conversion of BBR5 to a part of SMD by expert 1 .......................... 48
4.50 Conversion of BBR5 to a part of SMD by expert 3 .......................... 49

5.1 Executable Node ........................................................................... 56
5.2 Object Node .................................................................................. 56
5.3 Decision Node with control flows .................................................. 57
5.4 Activity Edge linking 2 activities ................................................... 57
5.5 Activity Partition ........................................................................... 58
5.6 Lifeline ......................................................................................... 58
5.7 Combined Fragment ...................................................................... 59
5.8 Message ....................................................................................... 59
5.9 State with compartments ............................................................... 60
5.10 Transition .................................................................................... 60
5.11 Package ........................................................................................ 60
5.12 Example of conversion to AD with CR_AD1 conversion rule .......... 63
5.13 Example of conversion to AD with CR_AD2 conversion rule .......... 63
5.14 Example of conversion to AD with CR_AD3 conversion rule .......... 64
5.15 Example of conversion to AD with CR_AD4 conversion rule .......... 64
5.16 Example of conversion to AD with CR_AD5 conversion rule .......... 64
5.17 Example of conversion to SD with CR_SD1 conversion rule ........... 67
5.18 Example of conversion to SD with CR_SD2 conversion rule ........... 67
5.19 Example of conversion to SD with CR_SD3 conversion rule ........... 68
5.20 Example of conversion to SD with CR_SD4 conversion rule ........... 68
5.21 Example of conversion to SD with CR_SD5, CR_SD6 conversion rules 69
5.22 Example of conversion to SMD with CR_SMD2 conversion rule ....... 71
5.23 Example of conversion to SMD with CR_SMD3 conversion rule ....... 72
5.24 Example of conversion to SMD with CR_SMD1, CR_SMD4, CR_SMD5 conversion rules ................................. 72
5.25 Class diagram for facts extracted from "Direct Delivery" project ...... 74
5.26 BBR 1 - Direct Delivery project - a part of AD ............................. 75
5.27 BBR 1 - Direct Delivery project - SD ............................................ 75
5.28 BBR 1 - Direct Delivery project - a part of SMD for package ......... 76
5.29 BBR 1 - Direct Delivery project - a part of SMD for transport type 76
5.30 BBR 2 - Direct Delivery project - a part of AD ............................. 77
5.31 BBR 2 - Direct Delivery project - SD ............................................ 77
5.32 BBR 2 - Direct Delivery project - a part of SMD for package ......... 78
5.33 BBR 3 - Direct Delivery project - a part of AD ............................. 78
5.34 BBR 3 - Direct Delivery project - SD ............................................ 79
5.35 BBR 3 - Direct Delivery project - a part of SMD for package ......... 79
11.1 UML class diagram of insurance domain
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Mapping of SBVR SE constructs to UML activity diagram constructs described in “Transformation of SBVR business design to UML models” [17]</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Mapping of SBVR SE constructs to UML sequence diagram constructs described in “Transformation of SBVR business design to UML models” [17]</td>
<td>11</td>
</tr>
<tr>
<td>5.1</td>
<td>Mapping of SBVR SE constructs to AD constructs</td>
<td>62</td>
</tr>
<tr>
<td>5.2</td>
<td>Mapping of SBVR SE constructs to SD constructs</td>
<td>66</td>
</tr>
<tr>
<td>5.3</td>
<td>Mapping of SBVR SE constructs to SMD constructs</td>
<td>70</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Software development projects start with analysis phase [14]. During this phase functional and non-functional requirements are defined. Besides them, business rules (see section 8 for definitions) are analysed by business analysts. Business rules are not only defined at this stage of software life cycle, as they represent logic of business they can vary in time and can be added in maintenance phase of application. Because they are often written in natural language it is hard to verify their consistency [19]. Specification of business rules is important, because business rules are later used in development phase.

Software engineers usually have to write application business logic that is stored in the document by hand, however this process is difficult and prone to errors [18]. Time and money is usually spent on correcting such errors which could have been avoided if there exists a scheme that could produce diagrammatic representations of these business logic and automatically produce source code [11]. This problem is faced in the field of software engineering.

Object Management Group (OMG) provides Semantics of Business Vocabulary and Business Rules (SBVR) standard to model business rules. Definition of SBVR can be found in [8] and is: "[SBVR] specification defines the vocabulary and rules (...) for documenting the semantics of business vocabularies and business rules for the exchange of business vocabularies and business rules among organizations and between software tools." SBVR was designed to be easy to understand by people from business (eg. managers) who don’t know jargon of software developers [15]. Because it uses formal logic, it can be easily processed by a computer [15]. It is an integral part of Model Driven Architecture (MDA) standard provided by OMG. MDA "unifies every step of development and integration from business modeling, through architectural and application modeling, to development, deployment, maintenance, and evolution integrate" [4]. It is a standard that allows conversion of business rules specified in SBVR into source code of software [11][9]. It is done not directly but with mediation of 3 layers of MDA [11][17]:

- Computational Independent Model (CIM) - eg. SBVR specifications,
Chapter 1. Introduction

- Platform Independent Model (PIM) - eg. UML, OCL, BPMN models,
- Platform Specific Model (PSM) - eg. UML models of solutions for specific technologies/languages (eg Java, C++);

A pilot study performed on articles on SBVR extracted from Summon@BTH and Google Scholar as search engines and "SBVR to UML" and "SBVR conversion" as search strings revealed that not every operative business rule (called also behavioral business rule - see section 8) is automatically transformable to UML activity diagram [17]. During the pilot study with Google Scholar and Summon@BTH, first 30 records per input string were checked - it was done for both searching engines separately what is totally about 120 entries. The pilot study didn’t reveal any clear list of properties of SBVR specifications which ensure that operative business rule is automatically convertible into one of UML behavioral diagrams (activity, state machine, sequence diagram). Raj notes that "Automatable rules are generally in the form of if-then or EventCondition-Action (ECA) format" [17], however there are more constructs to be dealt with in automation of SBVR to UML conversions than if-then rules and ECA formats [17].

A study of contemporary literature in SBVR reveals that there is lack of clearly defined properties of SBVR operative business rules which ensure that rule is automatically convertible into one of UML behavioral diagrams (activity, sequence, state machine diagram) or part of such diagram (eg. part of flow). There does not exist clearly defined mapping of SBVR constructs (eg. "then" construct) to UML behavioral diagram’s construct (eg. state machine transition construct). Raj [17] provides mapping of SBVR constructs used in specifying behavioral business rules to UML activity and sequence diagrams’ constructs [17] but it is only for rules that are "if then else" / "Event Condition Action" rules. Initial study revealed that are no guidelines for business analysts which help in specification of SBVR operative business rules which are automatically convertible to UML behavioral diagrams (activity, sequence, state machine). This lack of defined properties of SBVR business rules is the knowledge gap which to explore.

Results of planned research (see chapter 3) cover described "knowledge gap" areas.

1.1 Abbreviations

To improve readability following abbreviations are used:

- AD - Unified Modeling Language activity diagram, plural - ADs,
- BBR - behavioral business rule, plural - BBRs,
Chapter 1. Introduction

- BR - business rule, plural BRs,
- M1 - mapping of SBVR constructs to UML constructs provided in [17]:
  - M1 AD - Unified Modeling Language activity diagram’s constructs are mapped,
  - M1 SD - Unified Modeling Language sequence diagram’s constructs are mapped,
- M2 - mapping of SBVR constructs to UML constructs provided in this master thesis (see section 5.2.3):
  - M2 AD - Unified Modeling Language activity diagram’s constructs are mapped,
  - M2 SD - Unified Modeling Language sequence diagram’s constructs are mapped,
  - M2 SMD - Unified Modeling Language state machine diagram’s constructs are mapped,
- RQ - research question, plural - RQs
- SBVR SE - Semantics of Business Vocabulary and Business Rules (SBVR) Structured English,
- SD - Unified Modeling Language sequence diagram,
- SMD - Unified Modeling Language state machine diagram,

1.2 Objectives

Aim of this thesis is to initially explore field of automatic conversion of behavioral business rules - from SBVR representation to UML diagrams representations. Aim is reached by fulfilment of following objectives:

- To find out properties of SBVR behavioral rule which ensure that the rule can be automatically converted to UML behavioral diagrams (activity, sequence, state machine).
- To propose mapping of SBVR contracts to UML behavioral diagram’s constructs (activity, sequence, state machine diagrams are considered).
- To prepare guidelines which help to specify SBVR behavioral business rules that can be automatically transformed to UML behavioral diagrams (activity, sequence, state machine) or fragments of UML behavioral diagrams.
1.3 Research questions

- **RQ1.** What are properties of SBVR behavioral rules specifications which ensure that behavioral rule can be automatically converted to UML behavioral diagram (activity, sequence, state machine) or fragment of UML behavioral diagram? We seek to specify such properties which could be used in investigation of SBVR operative business rule to check if it can be automatically transformed to UML behavioral diagram/it’s fragment or not (activity, sequence, state machine diagrams are considered).

- **RQ2.** What is mapping of SBVR behavioral rules specifications constructs to constructs of UML behavioral diagrams (activity, sequence, state machine)? We seek to specify mapping of SBVR behavioral business rule constructs to constructs of UML behavioral diagrams. Properties found in RQ1 are necessary for behavioral business rule to allow automatic conversion of rule to UML diagram.

- **RQ3.** What are guidelines for analysts, which help in specification of SBVR behavioral business rules automatically convertible to UML behavioral diagrams (activity, sequence, state machine) or fragments of UML behavioral diagrams? We seek to specify such guidelines for business analysts which could be used in requirements gathering and analysis phase of software development process. Documentation created with usage of these guidelines will contain SBVR operative business rules that can be automatically transformed to UML behavioral diagrams or fragments of diagrams (activity, sequence, state machine diagrams are considered) according to the mapping - answer for RQ2.

1.4 Thesis structure

This master thesis has following chapters:

- **Introduction** (current chapter) - provides background for research, objectives and research questions,
- **Related Work** - provides overview of materials related to research topic of this thesis: business rule definitions, Business Rule Manifesto and SBVR formal specification,
Chapter 1. Introduction

- Research Method - describes way of research performed to answer research questions defined in "Introduction": case study and expert opinion,
- Data Collection - describes collected data for case study research method,
- Results - defines properties of SBVR behavioral business rules, mapping of SBVR constructs to UML constructs and guidelines which are answer for research questions,
- Analysis and Discussion - contains a discussion about "Results" chapter. Compares mapping of SBVR constructs to UML constructs (see section 5.2) with mapping provided by Raj (see section 2.3),
- Conclusions and Future Work - defines steps for industrial validation of properties, mapping and guidelines in "Results" chapter,
- Dictionary - defines fundamental notions for this thesis,
- Appendices A-D - contain expert opinions used in the case study (chapter "Method").
Chapter 2

Related Work

This chapter presents documents related to the field of modelling of behavioral business rules in SBVR and their automatic conversion to UML behavioral diagrams.

2.1 Formal specification of SBVR

According to [8, p.117] behavioral business rule is: "a prescribed, suggested, or self-imposed guide for conduct or action: a regulation or principle (...)".

Figure 2.1: Behavioral Rule Statement and its Kinds

![Diagram](image)

Taken from "Semantics of Business Vocabulary and Business Rules (SBVR)" [8, p.120]

SBVR formal [8] defines behavioral business rule as business rule that is a claim of obligation. It is expressed by operative business rule statement. Operative
business rule statement has 3 categories (see figure 2.1):

- obligation statement:
  
  - Example 1: "It is obligatory that each driver who is authorized for a rental is qualified" [10, p.4].
  - Example 2: "It is obligatory that the fuel level of the rented car of a rental is full at the actual pick-up date-time of the rental" [10, p.5].

- prohibition statement
  
  - Example: "It is prohibited that the rental duration of a rental is greater than 90 rental days." [10, p.5].

- restricted permission statement
  
  - Example: "It is permitted that a rental is open only if an estimated rental price of the rental is provisionally charged to a credit card that is in the name of the renter who is responsible for the rental." [10, p.5].
Figure 2.2: Kinds of Behavioural Elements of Guidance

Taken from "Semantics of Business Vocabulary and Business Rules (SBVR)" [8, p.117]

Behavioural business rule is as a category of business rule and element of governance (see figure 2.2). Element of governance is "element of guidance that is concerned with directly controlling, influencing, or regulating the actions of an enterprise and the people in it" [8, p.100]. Business rule is "rule that is under business jurisdiction" [8, p.98]. Business rule is a category of element of guidance. Element of guidance is "means that guides, defines, or constrains some aspect of an enterprise" [8, p.28]. Advice is "a claim of permission or of possibility" [8, p.99] that is opposite to rule, which is "a claim of obligation or of necessity" [8, p.29].

2.2 Business Rules Manifesto

Definition of properties which are answer for RQ1 (see 1.3) cannot disregard Business Rules Manifesto. SBVR is created in the spirit of the manifesto. Business Rules Group created manifesto and was one of co-developers of SBVR [6]. Annex F of SBVR formal documentation [9] reminds key concepts of the manifesto:
• "Deliberate Knowledge, Not A By-Product. Rules build on facts, and facts build on concepts as expressed by terms. Terms express business concepts; facts make assertions about these concepts; rules constrain and support these facts. Rules are basic to what the business knows about itself that is, to basic business knowledge. Rules need to be nurtured, protected, and managed." [9, p.1]

• "Declarative, Not Procedural. Rules should be expressed declaratively in natural-language sentences for the business audience. A rule is distinct from any enforcement defined for it. A rule and its enforcement are separate concerns.” [9, p.1]

• "Well-Formed Expression, Not Ad Hoc. Business rules should be expressed in such a way that they can be validated for correctness by business people. Business rules should be expressed in such a way that they can be verified against each other for consistency.” [9, p.1]

• "Of, By, and For Business People, Not IT People. Rules should arise from knowledgeable business people.” [9, p.1]

2.3 Transformation of SBVR business design to UML models

"Transformation of SBVR business design to UML models" [17] defines activity diagram elements with their SBVR corresponding elements. For the purpose of comparison with results of RQ2, the mapping of SBVR constructs to activity diagram constructs is presented in table 2.1.

Table 2.1: Mapping of SBVR SE constructs to UML activity diagram constructs described in "Transformation of SBVR business design to UML models" [17]

<table>
<thead>
<tr>
<th>SBVR construct</th>
<th>UML activity diagram element</th>
</tr>
</thead>
<tbody>
<tr>
<td>facts before &quot;if&quot; keyword which are built upon transitive verbs eg. &quot;user inserts card&quot; - what means &quot;insertion of card into machine&quot;</td>
<td>activity node eg. for BBR &quot;It is obligatory that each atm request exactly one password if a user inserts card&quot;, authors suggest &quot;request_password()&quot; as an activity node - merged verb (&quot;request&quot;) and second term of fact (&quot;password&quot;) (see figure 2.3).</td>
</tr>
</tbody>
</table>
fact after "if" keyword | guard of activity edge which target activity node corresponds to fact before "if" keyword. Eg. for BBR "It is obligatory that each atm request exactly one password if a user inserts card", authors suggest 'insert card == true' as a guard (see figure 2.3)

and | Fork Node and Join Node. Incoming transition of Fork Node has guard which is statement after "if" keyword. Outgoing transitions are pointed toward activities (built upon facts) combined with "and" keyword eg. for rule "It is obligatory that each atm print exactly one receipt and each atm eject the card if bank return badAccountmessage" guard for incoming transition of Fork Node is "return badAccountmessage == true" and outgoing transitions of Fork Node are pointing toward activities: "atm print receipt" and "atm eject card". Both activities have outgoing activity edges which are connected with Join Node.

Figure 2.3: Conversion of BBR by Raj

![Diagram](image)

Taken from "Transformation of SBVR business design to UML models" [17, p.35]

* Verbs used in facts can be either transitive or non-transitive. Transitive verb is a verb which implies action [17, p.32] and non-transitive doesn’t. [17, p.32] Transitive verbs are verbs which have active and passive voice - they connect subject (one who performs action) with object. Some binary verbs are transitive and every unary verb is non-transitive - does not have passive voice.

Although authors of "Transformation of SBVR business design to UML models" mention such UML constructs like: Activity Group, Activity Partition, they don’t define what SBVR constructs corresponds to them.

The paper [17] has also section "Mapping SBVR to UML Sequence Diagram components" [17, p.34] which presents mapping defined in table 2.2.
Table 2.2: Mapping of SBVR SE constructs to UML sequence diagram constructs described in ”Transformation of SBVR business design to UML models” [17]

<table>
<thead>
<tr>
<th>SBVR construct</th>
<th>UML sequence diagram element</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject (term)</td>
<td>lifeline with name subject (term)</td>
</tr>
<tr>
<td>object (term)</td>
<td>lifeline with name object (term)</td>
</tr>
<tr>
<td>fact (eg. ”atm ejects card”)</td>
<td>message with same name as in AD eg. ”ejects_card()”; source of message is subject term and destination is object term.</td>
</tr>
</tbody>
</table>

Presented mapping in table 2.2 makes an impression of draft. There is no construct which covers scenarios with alternative path of execution like: combined fragment.
Chapter 3

Research Method

This chapter presents description of methods used in the research of this master thesis. Chosen methodology was case study mixed with expert opinion. Elements of case study elements: context, case, unit of analysis are defined. Process of seeking for experts and experts’ background are described. Validity are listed and mitigation is presented.

3.1 Description of methods

- RQ1 and RQ2 - For these questions exploratory method was chosen because conversion of SBVR to UML is a field of study in which much research has not yet been done [17] [12]. Exploratory methods are defined in [20] as methods which have following purpose: "finding out what is happening, seeking new insights, and generating ideas and hypotheses for new research". Runeson et al. present case study as methodology for exploratory research whereas experiment, survey and action research are presented as methodologies for explanatory, descriptive and improving purposes respectively (see [20, p.15]). To answer RQ1 and RQ2 a case study mixed with expert opinion was applied. According to [20] case study is "suitable research methodology for software engineering research since it studies contemporary phenomena in its natural context". Case study was be performed on behavioral business rules and group of business analysts’ opinions. Each opinion showed conversion of SBVR behavioral business rule to each of UML diagram: activity, state machine, sequence. Because of confidentiality, I couldn’t obtain industrial examples of behavioral business rules in reasonable time and with usage of reasonable effort. Instead, examples of behavioral business rules included in online available projects and in a literature were investigated. Five rules were extracted from West Virginia insurance regulations [3]. Each business rule was expressed in SBVR Structured English notation which is "an English vocabulary for describing vocabularies and stating rules" [8]. Business analysts were asked to convert extracted behavioral business rules to UML
diagrams. In case when behavioral business rule was not transformable, the business analysts were asked to give a short explanation of the problem. Both UML and SBVR are contemporary phenomena. I already experienced that companies where I worked/work use UML (e.g. Exence S.A., Capgemini Poland) and at least one company in Wrocław uses SBVR (Aion Sp. z o.o.). Real life context of case study was simulated. Business analysts were real life practitioners or university lecturers who had experience in the field of business analysis. Prepared business rules were cases of real life behavioral business rules.

- **RQ3** - this question was answered based on properties found by answering RQ1. Each statement of guidelines covered at least one property found by answering RQ1. Because properties (answer for RQ1) and mapping functions (answer for RQ2) are strictly connected we cannot tell that guidelines are general. They support specification of BBRs which have properties (answer for RQ1) and can be converted automatically to UML diagrams according to mapping functions (answer for RQ2). BBR which has these properties and match BNF pattern defined in 5.2.1 can be converted according to the mapping functions.

### 3.2 Case study definition

Opinions of experts were particular conversions of SBVR behavioral business rules to UML diagrams. Experts didn’t define explicitly properties of SBVR behavioral business rules which ensure that particular rule can be automatically converted to UML diagram. Case study was a method by which properties of SBVR behavioral business rules were extracted. According to Guidelines for conducting and reporting case study research in software engineering [20] case study consists of: context, case and unit of analysis. In the context of this master thesis these elements are defined as followed:

- **context** - transformation of behavioral business rules to UML diagrams: sequence, activity, state machine,

- **case** - list of behavioral business rules extracted from West Virginia insurance regulations [3] and their transformations to UML diagrams (sequence, activity, state machine) done by selected business analysts,

- **unit of analysis** - each behavioral business rule with it’s transformation to UML diagrams (sequence, activity, state machine) done by selected business analysts;
3.3 Form preparation

To answer RQ1 and RQ2 author prepared a form for business analysts (see Annex A). The form contains a list of behavioral business rules extracted from insurance regulations of US state West Virginia published on site West Virginia, Offices of the Insurance Commissioner [3]. Extraction consisted of 5 phases:

1. Identification of behavioral business rule statements.

2. Definition of terms used in business rules.

3. Definition of facts used in business rules.

4. Classification of each chosen rule statement into one of 3 types (see ”SBVR formal specification” in section 2.1):
   - obligation statement
   - prohibition statement
   - restricted permission statement

5. Notation of rule in SBVR Structured English

Additionally, UML class diagram was prepared based on terms and facts (see Annex B). The form and the UML class diagram were validated by a lecturer before they were sent to experts in the field of business analysis. The lecturer had knowledge of SBVR and UML standards.

3.4 Searching for experts

Author of this master thesis was looking for business analysts at universities: Wroclaw University of Technology and Blekinge Tekniska Högskola. Only two were found - 9 people were queried. Other polish universities lecturers were queried and no business analyst was found - 7 emails were sent. 9 other individuals (business analysts from industry) were queried and 4 of them decided to take part in, but finally 2 of them refused due to lack of time. 3 centers providing workshops and trainings in the field of business analysis were queried but there was no positive response.

Overview of experts Four experts with following background were found:

- Expert 1 - had experience in lectures connected with business analysis and experience in industry as business analyst and software developer
• Expert 2 - had experience in industry in the role of business analysis and software developer

• Expert 3 - had experience only as lecturer in the field of software engineering (including business analysis)

• Expert 4 - had experience both in industry and as a lecturer in the field of business analysis

3.5 Threats to validity

3.5.1 Business rules bias

Description

Extracted BBRs which are enclosed in form for business analysts (see section 3.3), may be biased due to arbitrary selection. Chosen BBRs may not cover different categories of BBRs (see ”SBVR formal specification” in section 2.1)

Mitigation

Extracted rules contained at least one type of BBR (3.3).

3.5.2 Business analysts bias

Description

Selected business analysts (see section 3.4) may come from same industry or same university environment.

Mitigation

Selected business analysts were chosen from different IT companies. It was hard to predict university background.

3.5.3 Business analysts not available

Description

Business analysts (see section 3.4) may not be found - no positive response for an invitation to take part in the research.
Mitigation

Not only particular business analysts will be asked to take part in the research but also training centres for business analysts will be queried.

3.5.4 Business analysts cannot participate in the whole research

Description

Business analysts (see section 3.4) may not want to participate in whole research - eg. don’t want to convert every business rule to UML diagrams.

Mitigation

Partial answers given by business analysts will be accepted.

3.5.5 Bias in the form for business analysts

Description

Form used in the research (3.3) can be created improperly (eg. extracted rules can be incorrectly converted to SBVR notation).

Mitigation

Form used by business analysts will be validated by the lecturer who has knowledge in the field of business analysis.

3.6 Method overview

The diagram below presents overall view on steps which were performed during the research. Details of steps are given in sections: 3.3, 3.4 and the chapter 4.
Figure 3.1: Activity diagram of the research method
This chapter contains a case considered in this thesis and its initial analysis. According to section ‘Case study definition’ 3.2 the case is list of behavioral business rules extracted from West Virginia insurance regulations [3] and their transformations to UML diagrams (sequence, activity, state machine) done by experts in the field of business analysis.

Only 4 business analysts were reached to get their opinions (sequence, state machine, activity diagrams created based on given 5 business rules enclosed in form). For BBR 5 expert 3 left comment: ”probably [rule has] an error - this [life insurance policy] concept doesn’t exist on the left side [before if statement]”. Expert 4 due to lack of time, provided only partial data - activity diagrams and state machine diagrams for BBR1 and BBR3. Expert 1 and expert 4 provided digital versions of diagrams (Enterprise Architect and Visual Paradigm modelling tools were used respectively). Expert 2 and 3 provided data in non-digital way - in such case diagrams were draw in Astah by the author of this thesis. Comments and descriptions in Polish were translated to English. To check raw data of expert 2 and 3, see Appendix C and Appendix D respectively. Expert 2 was asked to repeat conversion of BBRs to SDs because the initial conversion contained technical details (Platform Specific Model).

4.1 Units of analysis

According to section ‘Case study definition’ 3.2 unit of analysis of the case is each behavioral business rule with its transformation to UML diagrams (sequence, activity, state machine) done by selected business analysts.

4.1.1 Business rule 1

It is obligatory that a variable life insurance policy which is issued by an insurer, complies the minimum requirements.
Activity diagram

Figure 4.1: Conversion of BBR1 to a part of AD by expert 1

Figure 4.2: Conversion of BBR1 to a part of AD by expert 2

Figure 4.3: Conversion of BBR1 to a part of AD by expert 3
Chapter 4. Data Collection

Commonalities
- Experts: 1, 2, and 3 used "Variable life insurance policy complies the minimum requirements" as guard for activity edge.
- Expert 4 and expert 2 described technical details like throwing of an exception ("throw MinimumRequirementsException") or reading data ("Read minimum requirements").

Differences
- Only expert 3 used transitive verb "issue" as name for operation.
Sequence diagram

Figure 4.6: Conversion of BBR1 to SD by expert 1

Comments: ”If there are multiple Minimum Requirements connected with Policy, there should be additional class managing requirements and a loop for messages from this managing class to Minimum Requirements”

Figure 4.7: Conversion of BBR1 to SD by expert 2
Commonalities

- Every expert used lifeline to model "Insurer" and "Variable Life Insurance Policy".

- Expert 1 and 3 used transitive verbs as names for messages which are operations.

Differences

- Only expert 3 did not choose lifeline to model "minimum requirements".

- Only expert 3 chose combined fragment where minimum requirements were used in guard statement. Other experts chose lifeline for "minimum requirements" which executed operation to check requirements: "check requirements()" or "complies()"

- Only expert 3 used terms as arguments for operations.

- Expert 2 used transitive verbs as names for messages which are not operations.
Chapter 4. Data Collection

State machine diagram

Figure 4.9: Conversion of BBR1 to a part of SMD by expert 1

Comments:
"These are states which can be concluded from this short rule. However, I’m not convinced if such approach will lead to optimal and readable outcome as a whole.

It is not clear if such states will be needed. This fulfillment of requirements could be also a value of certain field and serve as a guard to go to Issued state.

It’s difficult to say without the context and understanding how it works."
Chapter 4. Data Collection

Figure 4.10: Conversion of BBR1 to a part of SMD by expert 2

Figure 4.11: Conversion of BBR1 to a part of SMD by expert 3
Commonalities

- Experts: 1, 2 and 4 called states in SMDs with names which were not directly available in BBR1 statement eg.: "valid", "cancelled", "deleted".
- Expert 1 and 3 used transitive verb in passive form ("issued") as name of state.
- Expert 1, 2 and 3 used triggers for transitions.
- Expert 3 and 4 used guards for transitions.
- Expert 2 and 4 used initial pseudostate.

Differences

- Only expert 4 used merge symbol.
- Only expert 2 used final state.

4.1.2 Business rule 2

It is obligatory that each variable life insurance policy is filed with the Commissioner and approved by the Commissioner.
Activity diagram

Figure 4.13: Conversion of BBR2 to a part of AD by expert 1

Figure 4.14: Conversion of BBR2 to a part of AD by expert 2

Comments:
"I am not sure what does requirement "variable life insurance policy is filled with the Commissioner" mean. In typical case I can ask stakeholder about clarification."
Chapter 4. Data Collection

Figure 4.15: Conversion of BBR2 to a part of AD by expert 3

- Expert 1 and 3 used verbs and terms used in facts BBR2 as names for executable nodes in ADs (eg. "Fill Variable Life Insurance Policy" or "approves (VLIP)"")
- Every expert used guards for transitions in ADs.

Commonalities

- Expert 1 and 3 used verbs and terms used in facts of BBR2 as names for executable nodes in ADs (eg. "Fill Variable Life Insurance Policy" or "approves (VLIP)"")
- Every expert used guards for transitions.

Differences

- Only expert 2 used technical statements (eg. "Set output parameter to FALSE") as names for executable nodes.
- Only expert 3 used object nodes with names equal to terms used in facts.
Sequence diagram

Figure 4.16: Conversion of BBR2 to SD by expert 1

Comments:
"it’s not clear wether policy should be approved by the same Commissioner or other one? If other - second Actor should be created."

Figure 4.17: Conversion of BBR2 to SD by expert 2
Figure 4.18: Conversion of BBR2 to SD by expert 3

Commonalities

- Every expert used lifeline to model commissioner and variable life insurance policy.

Differences

- Only expert 1 modelled that operations "fills" and "approves" are operations executed by variable life insurance policy (arrow is on the side of executor).

- Only expert 2 modelled that commissioner checks if variable life insurance policy is filed and approved and does not perform operation of approving nor filing.

- Only expert 3 modelled that operations "fills" and "approves" are operations executed by commissioner. Only expert 3 modelled them as messages sent from commissioner to himself / herself.

- Only expert 3 used loop combined fragment.

- Only expert 3 modelled that operations performed by commissioner change state of variable life insurance policy ("approved", "filled").
State machine diagram

Figure 4.19: Conversion of BBR2 to a part of SMD by expert 1

Figure 4.20: Conversion of BBR2 to a part of SMD by expert 2

Figure 4.21: Conversion of BBR2 to a part of SMD by expert 3

Commonalities

- Experts: 1, and 3 called states with names that refer to passive voice of verbs used in facts of BBR2 ("approved", "filled").
- Every expert used actions for transitions which referred to verbs used in facts of BBR2 (eg. "approve", "approved by the Commissioner", "approved").
Differences

- Only expert 2 did not use verbs used in facts of BBR2 to call states ("new", "valid").
- Only expert 2 used join pseudo state.

4.1.3 Business rule 3

It is prohibited that an insurer issues a variable life insurance policy if the insurer is not licensed to do a life insurance business;

Activity diagram

Figure 4.22: Conversion of BBR3 to a part of AD by expert 1

Comments:
"This check could be done before or after issuing policy - it depends from the context."
Figure 4.23: Conversion of BBR3 to a part of AD by expert 2

Figure 4.24: Conversion of BBR3 to a part of AD by expert 3
Commonalities

- Experts: 1, 2, and 3, created executable node with name connected with issuing variable life insurance policy ("Issue Variable Life Insurance Policy", "Issue a variable life insurance policy", "Issue VLIP").
- Expert 1 and 3 modelled that insurer is licensed to do life insurance business with guard of activity edge.
- Expert 2 and 4 used activity final construct.
- Every expert used decision node construct.
- Expert 2 and 4 described technical details in part of AD ("Read Insurer", "LicensedInsurerException").
Differences

- Only expert 4 modelled that insurer is licensed to do life insurance business with executable node ("Verification whether the insurer of the VLI policy is licensed to do it")

- Only expert 2 used initial node.

Sequence diagram

Figure 4.27: Conversion of BBR3 to SD by expert 1

Comments:
"It’s not clear whether Insurer issues this policy manually or if this is done automatically when Commissioner fulfills and approves a policy. Who is an actor - user of a system in this scenario?

Also the sequence of actions is not clear when considering rule 2 and 3 - there are no data which activity is first and how the flow between those rules looks like.”
Commonalities

- Expert 2 and 3 used alt combined fragment with guard which refers to insurer’s license to do a life insurance business ("[Insurer is licensed]", "[not licensed to do a life insurance business] / [else]")

- Expert 1 and 2 claim that issuing ("issue", "issue a") is performed by a variable life insurance policy because variable life insurance policy is on the side of message’s arrow.
• Every expert defined different number of lifelines (2, 3, 4 lifelines)

Differences

• Only expert 1 created 2 lifelines with name "Insurer" - was it done by mistake?

• Only expert 2 defined issuing variable life insurance policy as message from insurer to himself / herself.

• Only expert 3 defined operation of issuing variable life insurance policy as message from insurer to himself / herself.

• Only expert 3 modelled that insurer changes state of VLIP (message "issued by (Insurer)").

State machine diagram

Figure 4.30: Conversion of BBR3 to a part of SMD by expert 1

![State machine diagram](image1)

Figure 4.31: Conversion of BBR3 to a part of SMD by expert 2

![State machine diagram](image2)
Commonalities

- Expert 1 and 2 modelled SMD of insurer.
- Expert 3 and 4 modelled SMD of variable life insurance policy.
- Expert 1, 2 and 3 called states with passive voice of verb used in fact: ”Licensed”, ”NotLicensed”, Issued by (Insurer).
- Expert 1 and 3 used fact after ”if” keyword as guards for transitions (”Insurer is licensed”, ”Insurer is licensed to do a life insurance business”)
• Expert 2 and 4 called states with names which were not verbs/nouns used in facts of BBR3: ”New”, ”Valid”, ”Invalid”

Differences

• Only expert 2 used fact after ”if” keyword as trigger of transition (”Accepted as licensed Insurer”).
• Only expert 3 used change of state ”issued by(insurer)” as trigger of transition.
• Only expert 2 used final state.
• Only expert 4 used initial pseudostate and merge symbol.

4.1.4 Business rule 4

It is permitted that the Commissioner approves life insurance business that is run by an insurer only if financial condition that has an insurer is sufficient.

Activity diagram

Figure 4.34: Conversion of BBR4 to a part of AD by expert 1
Chapter 4. Data Collection

Figure 4.35: Conversion of BBR4 to a part of AD by expert 2

Figure 4.36: Conversion of BBR4 to a part of AD by expert 3

Commonalities

- Every expert used fact "financial condition is sufficient" as guard for activity edge.
- Every expert modelled approving of life insurance business as executable node: "Commissioner approves Life Insurance Business", "Approve Life Insurance Business", "approves (LIB, Insurer)"
- Every expert used at least one decision node.

Differences

- Only expert 2 used initial node and activity final constructs.
Expert 2 described technical details as executable nodes eg.: "Read Insurer", "Read financial condition of the Insurer"

Expert 1 described technical details as guard of activity edge: "[Commissioner approval option selected]"

Sequence diagram

Figure 4.37: Conversion of BBR4 to SD by expert 1

Comments:
"It is too less information about the domain model - how to interpret certain classes - what do they mean, what responsibilities do they have. So this message flow could be in this direction or from Insurer to Life Insurance Business - it’s difficult to say right now.

It’s also not clear whether Commissioner should be aware of financial condition and on the basis of this knowledge approve the business or even not try to do so, or maybe just after the try of approval financial state should be checked automatically with success or fail message shown afterwards.”
Chapter 4. Data Collection

Figure 4.38: Conversion of BBR4 to SD by expert 2

Figure 4.39: Conversion of BBR4 to SD by expert 3

Commonalities

- Expert 1 and 2 modelled 3 terms: commissioner, life insurance business, insurer as lifelines.

- Expert 2 and 3 modelled fact: "insurer has sufficient financial condition" as a part of guard/whole guard of combined fragment: "insurer has sufficient financial condition", "financial condition that has an insurer is sufficient"
• Expert 1 and 2 claim that approving ("approve()", "approves") is performed by a life insurance business because life insurance business is on the side of message’s arrow.

Differences

• Only expert 3 modelled 2 terms: commissioner, life insurance business as lifelines.

• Only expert 3 defined operation of approving life insurance business as message from commissioner to himself/herself.

• Only expert 3 modelled that commissioner changes state of life insurance business (message "approved").

• Expert 2 described technical details as message from commissioner to life insurance business: "read".

State machine diagram

Figure 4.40: Conversion of BBR4 to a part of SMD by expert 1

Comments:
"It’s not clear what should be done if Insurer is not financially approved. Is he dissaproved (other state) or simply stays in previous state?”
Commonalities

- Every expert used passive voice of verb: "approve" to give name of state: "FinanciallyApproved", "Approved"

- Expert 1 and 3 used successful approval of life insurance business as trigger of transition ("ApproveFinancially[TRUE]", "approved")

Differences

- Only expert 2 called states with name which was not in BBR4: "New", "Denied".

- Only expert 1 used facts after "if" statement as trigger of transition: "Insurer has sufficient financial condition".
• Only expert 3 used facts after "if" statement as guard of transition: "financial condition that has an insurer is sufficient"

• Only expert 2 used initial pseudostate and merge symbol.

• Only expert 2 used trigger built upon facts after "if" statement ("Insurer has sufficient financial condition").

4.1.5 Business rule 5

It is obligatory that an insurer provides illustration which is dedicated for prospective purchaser if the life insurance policy is identified to be used with an illustration.

Activity diagram

Figure 4.43: Conversion of BBR4 to a part of AD by expert 1

Comments:
"It’s not clear what does it mean 'illustration which is dedicated for prospective purchaser'. Does it exist illustration not dedicated for purchaser? It is not shown in class diagram."
Commonalities

- Expert 1 and 3 created semantically same parts of ADs.
- Every expert used facts after "if" keyword as guards for activity edge: "[life insurance policy is identified to be used with an illustration]"
- Every expert used decision node.
Differences

- Only expert 2 used technical statements (e.g. "Read Life Insurance Policy for the Prospective Purchaser", "Error") as names for executable nodes.

- Only expert 2 used initial node and activity final constructs.

Sequence diagram

Figure 4.46: Conversion of BBR5 to SD by expert 1

Comments:
"It’s not clear whether Insurer provides illustration in a system or in real life? Is it in a scope of the system or not?

If it is done in the system, how is it done? Is it accessed by Purchaser account, send by e-mail or just printed from insurer account and handed to Purchaser? In all options diagram would like differently.”
Commonalities

- Every expert used verb "provide" from fact before "if" keyword as name of message.

- Expert 2 and 3 used combined fragment (alt and opt respectively) with guard created based upon facts after "if" keyword: "LIP is identified to be used with illustration", "the life insurance policy is identified to be used with an illustration".
**Differences**

- Every expert defined different number of lifelines (2, 3, 4 lifelines)
- Expert 1 claims that operations "checkIllustration()" and "provide()" are operations executed by life insurance policy and illustration respectively (arrow is on the side of executor).
- Only expert 3 defined operation of providing illustration as message from insurer to himself/herself.
- Only expert 3 modelled that operation performed by insurer changes state of life insurance policy ("Illustration Provided(illustration)").

**State machine diagram**

Figure 4.49: Conversion of BBR5 to a part of SMD by expert 1

Comments:
"It is not clear whether this is a constant state of Illustration or not. It depends on the context and the usage of this part of a system if we would need such state or not.

It's also not clear if we need to have such state of Policy."

**Expert 2**

Expert 2 didn’t provide SMD for BBR5.

Comments:
"I have no idea for state machine diagram. I don’t really know state of what might
be depicted.”

Figure 4.50: Conversion of BBR5 to a part of SMD by expert 3

Commonalities

- Expert 1 and 3 used triggers ("provideIllustration” and illustration provided respectively).

Differences

- Expert 1 provided a part of SMD for illustration, expert 3 for life insurance policy and expert 2 didn’t provide any part of SMD for BBR5.
- Only expert 3 used guard which is fact after "if” keyword.

4.1.6 Commonalities - summary

Facts after ”if” keyword are used as guards for:

- activity edges in AD,
- combined fragments in SD,
- transitions in SMD.

Guards in 3 listed above constructs are boolean statements. In case they are ”true” activity edge can be traversed in AD, messages covered by combined fragment in SD can be sent and state can be changed in SMD (transition can happen) respectively.

Verbs in active form are used in:

- executable nodes as names of activity in AD,
- activity edges as action in AD,
Chapter 4. Data Collection

- messages as name of message in SD (both in active and passive voice, passive when change of state is sent),
- transitions as triggers in SMD.
- states as names (passive voice of verb) in SMD.

In the first 4 items above, verbs refer to activity modelled by: executable node, action of activity edge, message and trigger of transition respectively. Verb in passive voice can be used as name of state because it refers to finished activity which changes state of object for which activity was performed.

Terms are used in:
- lifelines as names of lifelines in SD - terms which are subject of operation,
- SD as names of arguments for messages which are operations,
- AD as names of arguments for operations presented by executable nodes,
- object nodes as names of object nodes in AD,
- SMD as object which states SMD describes.

Terms denote things. Things can be modelled as lifelines in SD, arguments of operations which are messages in SD or operations presented by executable nodes in AD, object nodes in AD. States of things can be described by SMD.
Chapter 5

Results

In this chapter are answers for RQs stated in 1.3:

- Properties which ensure that BBR can be automatically converted into AD, SD and SMD.

- Mapping functions that allow automatic conversion of BBR into AD, SD and SMD if BBR match Backus—Naur Form (BNF) pattern defined in 5.2.1

- Guidelines which are intended to help in specification of BBRs which can be automatically converted to AD, SD and SMD according to 5.2

5.1 Properties

Properties in this section are mandatory for automated conversion of BBRs in SBVR SE to UML behavioral diagrams according to mapping provided in the section 5.2. They are answers for RQ1 defined in 1.3. Facts before "if" keyword have been called conclusion facts and facts after "if" keyword have been called hypothesis facts (see 5.2.1) eg. for BBR: "It is prohibited that an insurer issues a variable life insurance policy if the insurer is not licensed to do a life insurance business" conclusion facts are: "an insurer issues a variable life insurance policy" and hypothesis facts are: the insurer is not licensed to do a life insurance business. If BBR hasn’t got "if" then it has only conclusion facts. "If" keyword can be used in BBR implicitly eg. in rule: "It is obligatory that a variable life insurance policy which is issued by an insurer, complies the minimum requirements" implicit "if" is between insurer and complies. Two types of properties can be defined:

- syntax properties - define syntax of BBR

- semantic properties - refer to dependencies between terms
Chapter 5. Results

5.1.1 Syntax properties

- Explicitly defined if keyword.
  Justification: if keyword is not obligatory in case of occurrence hypothesis facts eg. BBR "It is obligatory that a variable life insurance policy which is issued by an insurer, complies the minimum requirements" hasn’t got if statement but has hypothesis fact which is "complies the minimum requirements. If delimits conclusion facts and hypothesis facts. The rule can be transformed to prohibition statement in order to have if statement: "It is prohibited that a variable life insurance policy is issued by an insurer if a variable life insurance policy does not comply the minimum requirements". BBR1 and BBR5 in the form (see 10) are both obligation statements. Comma sign used in BBR1 after term insurer was understood by business analysts as delimiter between activity (fact variable life insurance policy is issued by insurer) and condition (fact variable life insurance policy complies minimum requirements). It was similarly understood as if keyword in BBR5 - check commonalities of AD, SMD for BBR1 (4.1.1) and commonalities of AD, SD for BBR5 (4.1.5).

- <subject term> is listed every time a fact in conclusion facts uses it - no shortened version is used (it is allowed in SBVR SE [8, p.271]). For example Insurer issues variable life insurance policy and approves variable life insurance policy should be written in long form: Insurer issues variable life insurance policy and insurer approves variable life insurance policy.
  Justification: the property is not directly extracted from 'Units of analysis' (see 4.1) but is provided to simplify BNF (Backus —Naur Form) pattern in 5.2.1

5.1.2 Semantic properties

- And binary operator defines a sequence of execution. Fact on the left of and is preceding the fact on the right of and.
  Justification: SBVR Structured English doesn’t constraint precedence of facts in conjunction (see [8] p.141-143)). To provide property of automatic conversion we constrain SBVR SE to define precedence of facts. In the 'Commonalities - summary' (see 4.1.6) it is listed that verbs are used as names of executable nodes in AD, names of messages in SD, names of states in SMD. Based on this observation following conversion rules were created: CR_AD2, CR_SD2 and CR_SMD1 respectively. Because verbs are essential part of facts (see [8] and eg. [1]) and they were mapped to UML constructs which model particular activities then facts should also be ordered. This assumption is valid in case sequential execution of activities is performed
instead of parallel. Conversion rules presented in this thesis (see CR_AD3, CR_SD4 and CR_SMD3 in 5.2) define sequential execution of activities.

- Transitive verbs are used for every fact in conclusion facts. Justification: Transitive verbs are verbs which imply action (see 2.3). Activity diagram and sequence diagrams model actions ('Executable node' in activity diagram and 'message' in sequence diagram) and state machine diagram models states which are results of actions. For example BBR: ”It is prohibited that the rental duration of a rental is greater than 90 rental days.” cannot be converted because verbs: to be and to have (synonym: ”rental has a rental duration”) are not transitive. For following BBR: ”It is obligatory that an employee inspects package and an employee chooses a transport type that transports package” every verb in conclusion facts is transitive (inspects ->is inspected by, chooses ->is chosen by). In the 'Commonalities - summary’ (see 4.1.6) it is listed that verbs in active voice are used as names of activities in AD, names of messages in SD and verbs in passive voice are used as names of states in SMD. Verbs in conclusion facts are converted to UML constructs which include verbs in active and passive voice (see CR_AD2, CR_SD2 and CR_SMD1 in 5.2 - CR_AD2, CR_SD2 refer to verbs which have active and passive voice whereas CR_SMD1 refers to verbs which have passive voice).

- At least one term from the conclusion facts should be also in hypothesis facts in case if the instances of terms from conclusion facts are related to instances of terms from hypothesis facts. Keyword the should be used to show that duplicated term occurrence refers to same thing. Justification: Relationships between terms defined by facts are not sufficient to determine if things denoted by these terms are related eg. for BBR: ”It is permitted that the Commissioner approves life insurance business that is run by an insurer only if financial condition is sufficient” (compare with BBR4 in 4.1.4) we don’t know if particular financial condition refers to particular insurer mentioned in conclusion facts, even if insurer concept is related to financial condition concept by fact. In order to provide relation between things in the conclusion facts and things in the hypothesis facts, the rule can be corrected by providing insurer term (hypothesis facts part of rule): It is permitted that the Commissioner approves life insurance business that is run by and insurer only if financial condition that has the insurer is sufficient.

BBR4 and its conversions to SD done by expert 1 and 2 (see 4.1.4) show that business analysts understood rule as it refers to same insurer - despite that keyword an was used in the form. Keyword the ensures identity of both insurers.
5.2 Mapping of SBVR constructs to UML constructs

Mapping of SBVR constructs to UML constructs presented in this section is created based on section ”Units of analysis” [4.1] especially its subsection ”Commonalities - summary” [4.1.6].

Subsections of this section are following:

- Domain description - defines structure of BBRs in BNF notation for which mapping is provided,
- Codomains of a functions - refers to UML specification to define codomains of mapping functions from next subsection (Mapping functions),
- Mapping functions - this subsection defines 3 mapping functions:
  - SBVR SE constructs to UML AD constructs,
  - SBVR SE constructs to UML SD constructs,
  - SBVR SE constructs to UML SMD constructs;
- Test examples - shows usage of mapping functions applied to ”Direct Delivery” project from Capgemini Poland.

5.2.1 Domain description

Backus—Naur Form was chosen to represent BBRs. BNF is notation technique for context-free grammar and can be used to create parser [13]. Parser is not included in this master thesis.

Structure of BBRs in BNF notation for which mapping functions to UML are provided (see: 5.2.3) is following (root symbol is <automatable BBR>):

<automatable BBR>::= <modal operation><conclusion facts>| <modal operation><conclusion facts><if keyword><hypothesis facts>

<modal operation>::= It is obligatory that | It is prohibited that | It is permitted that

<if keyword>::= if | only if

$conclusion facts>::= <fact with binary transitive verb>| <fact with binary transitive verb><designation sequence>| <fact with binary transitive verb><binary operator><conclusion facts>| <fact with binary transitive verb><designation sequence><binary operator><conclusion facts>
The grammar of BBR enables to instantiate a subset of the whole BBRs. Non-terminal symbols:

- <transitive verb active voice>
- <transitive verb passive voice>
- <unary verb>

are not defined ("::=" sign) in the above description, because they refer to verbs used in facts.

Non-terminal symbols:

- <term>,
- <object term>,
- <binary operator>
• <subject term>

are not defined by terminal symbols because they refer to terms dictionary for BBRs which is prerequisite for writing BBRs. Terms in SBVR SE are English nouns - eg. taken from "Oxford English Dictionary" [5].

5.2.2 Codomains of functions

Next subsection 5.2.3 defines 3 functions which map SBVR SE constructs defined in 5.2.1 to constructs of AD, SD or SMD. Each function has a different codomain defined by the set of constructs which are either AD constructs or SD constructs, or SMD constructs. Some constructs are common for all types of UML diagrams. The full list of elements for AD, SD, SMD and for all types of UML diagrams can be found in UML documentation [2]. Below are presented only these elements which were used in mapping as possible function values.

UML Activity diagram

• Executable Node - "An ExecutableNode is an ActivityNode that carries out a substantive behavioral step of the Activity that contains it. (...) An ExecutableNode may also consume and produce data, but it must do so through related ObjectNodes." [2, p.401]

Figure 5.1: Executable Node

![Executable Node](image1)

Taken from [2, p.402]

• Object Node - "An ObjectNode is a kind of ActivityNode (...) used to hold value-containing object tokens during the course of the execution of an Activity." [2, p.394]

Figure 5.2: Object Node

![Object Node](image2)

Taken from [2, p.397]
• Decision Node - "A DecisionNode is a ControlNode that chooses between outgoing flows. A DecisionNode shall have at least one and at most two incoming ActivityEdges, and at least one outgoing ActivityEdge. (...) If any of the outgoing edges of a DecisionNode have guards, then these are evaluated for each incoming token."[2, pp.397-398]

Figure 5.3: Decision Node with control flows

• Activity Edge - "An ActivityEdge is a directed connection between two ActivityNodes along which tokens may flow, from the source ActivityNode to the target ActivityNode. (...) An ActivityEdge may have a guard, which is a ValueSpecification that is evaluated for each token offered to the edge. An offer shall only pass along an ActivityEdge if the guard for the edge evaluates to true for the offered token. An ActivityEdge without a guard is equivalent to one with a guard that evaluates to true for every token."[2, pp.373-374]

Figure 5.4: Activity Edge linking 2 activities

• Activity Partition - "An ActivityPartition is a kind of ActivityGroup for identifying ActivityNodes that have some characteristics in common. ActivityPartitions can share contents. They often correspond to organizational units in a business model."[2, p.384]
Chapter 5. Results

Figure 5.5: Activity Partition

UML Sequence diagram

- Lifeline - "Lifeline describes the time-line for a process, where time increases down the page. The distance between two events on a time-line does not represent any literal measurement of time, only that non-zero time has passed. Events on the same time-line are ordered linearly down the page, except where they occur within a parallel combined fragment, or along a lifeline within a coregion." [2] p.570

Figure 5.6: Lifeline

- Combined Fragment with alt operator - "The semantics of a Combined-Fragment is dependent upon the interactionOperator (...). The interactionOperator alt designates that the CombinedFragment represents a choice of behavior. At most one of the operands will be chosen. The chosen operand must have an explicit or implicit guard expression that evaluates to true at this point in the interaction. An implicit true guard is implied if the operand has no guard." [2] pp.580-581
• Message - "The semantics of a complete Message is simply the trace \(<\text{sendEvent}, \text{receiveEvent}\)\). (...) The signature of a Message refers to either an Operation or a Signal. The name of the Message must be the same as the name of the referenced Operation or Signal." [2, p.572]

UML State machine diagram

• State - "A State models a situation in the execution of a StateMachine Behavior during which some invariant condition holds. In most cases this condition is not explicitly defined, but is implied, usually through the name associated with the State." [?, p.306]
Transition - "A Transition is a single directed arc originating from a single source Vertex and terminating on a single target Vertex (the source and target may be the same Vertex), which specifies a valid fragment of a StateMachine Behavior. It may have an associated effect Behavior, which is executed when the Transition is traversed (executed).” [2, p.312]

Common for every type of UML diagram

- Package - "Packages provide the main generic structuring and organizing capability of UML. (...) A Package is a namespace for its members, which comprise those elements associated via packagedElement (which are said to be owned or contained), and those imported.” [2, p.239] A package can be used for across all types of UML diagrams. [2, pp.239,686]
5.2.3 Mapping functions

Presented in this section tables define 3 conversion functions. Each of them uses a subset of domain described in subsection: 5.2.1. Each table has following columns:

- first column which groups elements of column ”SBVR construct” by BNF non-terminal element (see section 5.2.1) or SBVR category of words (eg. keywords category),

- second column ”Rule ID” which contains identification numbers of conversion rules. Rules ID has pattern ”CR_T_N” where T is abbreviation of type of diagram (AD, SD or SMD) and N is rule number,

- third column ”SBVR construct” contains SBVR SE constructs eg. <subject term>, <verb>,

- fourth column ”UML TT diagram element” where TT is type of diagram (activity diagram, sequence diagram or state machine diagram) - contains constructs of particular type of UML diagram, which are values of mapping function for keys in ”SBVR construct” column.

Graphical illustration of function output for given input is presented below every table.
Table 5.1: Mapping of SBVR SE constructs to AD constructs

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>SBVR construct</th>
<th>UML activity diagram element</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR_AD1</td>
<td><code>&lt;subject term&gt;</code></td>
<td>Activity Partition with name of <code>&lt;subject term&gt;</code></td>
</tr>
<tr>
<td>CR_AD2</td>
<td><code>&lt;transitive verb active voice&gt;</code> &lt;transitive verb passive voice&gt;, <code>&lt;object term&gt;</code></td>
<td>Executable node with activity name &quot;&lt;transitive verb active voice&gt;(&lt;object term&gt;)&quot; connected with object node as input and object node as output. Object node is input for the executable node only if <code>&lt;object term&gt;</code> concerns fact which is not connected with other fact with <code>and</code> operator or it is in a first fact in facts connected with <code>and</code>. Then (as input) object node has name &quot;&lt;object term&gt;&quot;. The object node is also an output for the executable node and then it has name &quot;&lt;transitive verb passive voice&gt;&lt;object term&gt;&quot;.</td>
</tr>
<tr>
<td></td>
<td>any construct from <code>&lt;designation sequence&gt;</code></td>
<td>No UML construct.</td>
</tr>
<tr>
<td>CR_AD3</td>
<td><code>and</code></td>
<td>activity edge between: object node created on the basis of <code>&lt;object term&gt;</code> taken from fact on the left side of operator (source node) and executable node created on the basis of fact on the right side of operator (target node)</td>
</tr>
<tr>
<td>keywords</td>
<td>CR_AD4 <code>if</code>, <code>only if</code></td>
<td>Decision node and activity edge. Activity edge links decision node with first executable node created on the basis of <code>&lt;conclusion facts&gt;</code>.</td>
</tr>
<tr>
<td></td>
<td><code>the</code>, <code>a</code>, <code>an</code></td>
<td>No UML construct.</td>
</tr>
<tr>
<td></td>
<td><code>that</code>, <code>which</code></td>
<td>No UML construct.</td>
</tr>
<tr>
<td>&lt;hypothesis facts&gt;</td>
<td>CR_AD5 <code>every construct</code></td>
<td>Guard of activity edge that links decision node (see <code>if</code> statement) and first executable node created on the basis of <code>&lt;conclusion facts&gt;</code>. Guard is whole <code>&lt;hypothesis facts&gt;</code>. If modality operator is &quot;It is prohibited that &quot; then guard is negated.</td>
</tr>
</tbody>
</table>
Conversion rule ID: CR_AD1
Text: `<subject term> eg. <Commissioner>
Mapped element:

Figure 5.12: Example of conversion to AD with CR_AD1 conversion rule

Conversion rule ID: CR_AD2
Text: `<transitive verb active voice> <object term>, eg. issues VLIP
Mapped element:

Figure 5.13: Example of conversion to AD with CR_AD2 conversion rule

Conversion rule ID: CR_AD3
Text: `<subject term> <transitive verb active voice 1> <object term1> and <subject term> <transitive verb active voice 2> <object term2>, eg. employee inspects package and employee chooses transport type
Mapped elements:
Chapter 5. Results

Figure 5.14: Example of conversion to AD with CR_AD3 conversion rule

Conversion rule ID: CR_AD4
Text: `<subject term> <transitive verb active voice> <object term>` if, e.g., drone transports package if
Mapped elements:

Figure 5.15: Example of conversion to AD with CR_AD4 conversion rule

Conversion rule ID: CR_AD5
Text: if `<hypothesis facts>`, e.g., if a package is packed into a shipping container
Mapped elements:

Figure 5.16: Example of conversion to AD with CR_AD5 conversion rule
Sequence diagrams created by experts (see section 4.1) look similar except diagrams proposed by expert 3. The difference is about executing of operations. Expert 3 linked lifeline of <subject term> with itself with message "<transitive verb active voice>(<object term>)" (see for example figures 4.8 or 4.18) and then performed change of state of <object term> by sending message "<transitive verb passive voice>" from <subject term> to <object term>. Other experts did not link <subject term> with itself with message, but they linked <subject term> and <object term> with message that presented operation (see 4.6 4.17). Solution provided by other experts is semantically incorrect because message which represents operation cannot point lifeline with name "<object term>". In such cases lifeline pointed by a message is also executor of an operation presented by the message [2, p.576] what means that for referenced examples eg. "variable life insurance policy issues" or "minimum requirements check requirements".
<table>
<thead>
<tr>
<th>Rule ID</th>
<th>SBVR construct</th>
<th>UML sequence diagram element</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR_SD1</td>
<td><code>&lt;subject term&gt;</code></td>
<td>The lifeline with name of the <code>&lt;subject term&gt;</code></td>
</tr>
<tr>
<td>CR_SD2</td>
<td><code>&lt;transitive verb active voice&gt;</code> that connects <code>&lt;subject term&gt;</code> and <code>&lt;object term&gt;</code></td>
<td>2 messages: 1st one with name <code>&lt;transitive verb active voice&gt;</code>(&lt;<code>object term</code>&gt;) that connects subject with itself (sendEvent role and receiveEvent role is <code>&lt;subject term&gt;</code>), next (2nd) message with name <code>&lt;transitive verb passive voice&gt;</code>). Second message is connected with the <code>&lt;subject term&gt;</code> (sendEvent role) and <code>&lt;object term&gt;</code> (receiveEvent role)</td>
</tr>
<tr>
<td>CR_SD3</td>
<td><code>&lt;object term&gt;</code></td>
<td>The lifeline with name ”&lt;object term&gt;”</td>
</tr>
<tr>
<td></td>
<td>any construct from <code>&lt;designation sequence&gt;</code></td>
<td>No UML construct.</td>
</tr>
<tr>
<td>CR_SD4</td>
<td><code>and</code></td>
<td>No UML construct. Defines only precedence of messages - message created based upon <code>&lt;transitive verb active voice&gt;</code> on the left side of operator precedes message created based upon <code>&lt;transitive verb active voice&gt;</code> on the right side of operator.</td>
</tr>
<tr>
<td></td>
<td>&lt;keywords&gt;</td>
<td>Combined fragment that covers messages defined by <code>&lt;conclusion facts&gt;</code>.</td>
</tr>
<tr>
<td></td>
<td><code>if, only if</code></td>
<td>No UML construct.</td>
</tr>
<tr>
<td></td>
<td><code>the, a, an</code></td>
<td>No UML construct.</td>
</tr>
<tr>
<td></td>
<td><code>that, which</code></td>
<td>No UML construct.</td>
</tr>
<tr>
<td>CR_SD6</td>
<td>every construct</td>
<td>Guard of combined fragment defined by <code>if</code> or <code>only if</code>. Guard is whole <code>&lt;hypothesis facts&gt;</code> If modality operator is ”It is prohibited that ” then guard is negated.</td>
</tr>
</tbody>
</table>
Chapter 5. Results

Conversion rule ID: CR_SD1
Text: `<subject term>`, eg Commissioner
Mapped elements:

Figure 5.17: Example of conversion to SD with CR_SD1 conversion rule

Conversion rule ID: CR_SD2
Text: `<subject term> <transitive verb active voice> <object term>`, eg drone transports package
Mapped elements:

Figure 5.18: Example of conversion to SD with CR_SD2 conversion rule

Conversion rule ID: CR_SD3
Text: `<object term>`, eg package
Mapped elements:
Figure 5.19: Example of conversion to SD with CR_SD3 conversion rule

Conversion rule IDs: CR_SD4

Text: `<subject term> <transitive verb active voice 1> <object term1> and <subject term> <transitive verb active voice 2> <object term2>`, eg. employee inspects package and employee chooses transport type

Mapped elements:

Figure 5.20: Example of conversion to SD with CR_SD4 conversion rule
Conversion rule IDs: CR_SD5, CR_SD6
Text: if <hypothesis facts>, eg. if a package is packed into a shipping container
Mapped elements:

Figure 5.21: Example of conversion to SD with CR_SD5, CR_SD6 conversion rules
Table 5.3: Mapping of SBVR SE constructs to SMD constructs

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>SBVR construct</th>
<th>UML state machine diagram element</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;conclusion facts&gt;</td>
<td></td>
<td>No UML construct. Only &lt;object term&gt; states are considered</td>
</tr>
<tr>
<td>&lt;transitive verb passive voice&gt;</td>
<td>State with name equals to &lt;transitive verb passive voice&gt;. State is part of diagram that presents states of &lt;object term&gt; bound with the transitive verb. If the transitive verb is used more than once for same &lt;object term&gt; but with different prepositions (eg. see BBR 2 in subsection &quot;Test examples&quot;) then state is not duplicated.</td>
<td></td>
</tr>
<tr>
<td>&lt;object term&gt;</td>
<td>Package with name &lt;object term&gt;. The package covers all states created for the object term.</td>
<td></td>
</tr>
<tr>
<td>any construct from &lt;designation sequence&gt;</td>
<td>No UML construct.</td>
<td></td>
</tr>
<tr>
<td>and</td>
<td>if facts connected with and operator have the same &lt;object term&gt; then result of mapping are 2 states with names: &quot;&lt;transitive verb passive voice 1&gt;&quot; and &lt;transitive verb passive voice 2&gt; and transition - outgoing from state with name &quot;&lt;transitive verb passive voice 1&gt;&quot; and incoming to &lt;transitive verb passive voice 2&gt;.</td>
<td></td>
</tr>
<tr>
<td>if , only if</td>
<td>Existence of if , only if means that there is guard before first state constructed based on binary &lt;transitive verb passive voice&gt;</td>
<td></td>
</tr>
<tr>
<td>the, a, an</td>
<td>No UML construct.</td>
<td></td>
</tr>
<tr>
<td>that, which</td>
<td>No UML construct.</td>
<td></td>
</tr>
<tr>
<td>&lt;hypothesis facts&gt;</td>
<td>Guard of transition which target is first state created on the basis of &lt;conclusion facts&gt; - guard is whole &lt;hypothesis facts&gt;. If modality operator is &quot;It is prohibited that &quot; then guard is negated.</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 5. Results

Conversion rule ID: CR_SMD1
Text: `<transitive verb passive voice>`, eg. transported
Mapped element:

Conversion rule ID: CR_SMD2
Text: `<object term>`, eg. package
Mapped element:

Figure 5.22: Example of conversion to SMD with CR_SMD2 conversion rule

Conversion rule ID: CR_SMD3
Text: `<object term> is `<transitive verb passive voice 1> by `<subject term1> and `<object term> is `<transitive verb passive voice 2> `<subject term2>, eg. variable life insurance policy is filed with Commissioner and variable life insurance policy is issued by insurer
Mapped elements:
Figure 5.23: Example of conversion to SMD with CR_SMD3 conversion rule

Conversion rule IDs: CR_SMD1, CR_SMD4, CR_SMD5

Text: `<object term> is <transitive verb passive voice> <subject term> if <hypothesis facts>`, eg. `package is transported by drone if package is packed into a shipping container`

Mapped elements:

Figure 5.24: Example of conversion to SMD with CR_SMD1, CR_SMD4, CR_SMD5 conversion rules

Proposed mapping of SBVR constructs to SMD constructs refers to state machine of `<object term>`. Each distinct `<object term>` in BBR will lead us to generate separate SMD.

5.2.4 Comparison of M1 and M2

M1 AD maps transitive binary verbs in active voice and object term to executable node with name "transitive_verb_object_term()". M2 AD maps same constructs to executable node with name "transitive_verb (object_term)" and object node incoming to executable node and outgoing it, object node has name "(object_term)" (see CR_AD2 in section 5.2.3). M1 AD maps if "<subject term><transitive verb active voice><object term>" to activity edge guard "<transitive verb active voice>_<object term>= true". M2 AD maps same constructs to decision node and guard of outgoing transition, guard is "<subject term><transitive verb active voice><object term>" (see CR_AD4 and CR_AD5 in section 5.2.3). M1 AD maps and to fork, join nodes (parallel execution of activities). M2 AD maps the same construct to activities which are executed in sequence (see CR_AD3 5.2.3). M2 AD maps <subject term>(see section 5.2.1) to activity partition. M1 AD does not provide any AD construct for <subject term>.

Both M1 AD and M2 AD require explicitly defined "if" for BBRs which are executed under conditions. Both of them map <hypothesis facts> to guards of activity edge but guards don’t have the same names.
It is hard to compare M1 SD and M2 SD because M1 definition in [17] is ambiguous. Not ambiguous in M1 SD is that object terms and subject terms are mapped to lifelines what is the same as in M2 SD.

5.2.5 Test examples

Following examples of BBRs (taken from Direct Delivery project in Capgemini Poland) are used to present usage of defined mappings (see 5.2.3):

- It is obligatory that an employee inspects package and an employee chooses a transport type that transports package.

- It is prohibited that a package is transported by a drone if package is transported to Logistic Center.

- It is permitted that a package is transported by a drone only if a package is packed into a shipping container and a package is packed by an employee.

Terms:

- employee
- package
- transport type
- Logistic Center
- drone - category of transport type
- truck - category of transport type
- shipping container
Figure 5.25: Class diagram for facts extracted from "Direct Delivery" project

Below are: part of AD, SD and part of SMD created based on business rule 1, 2 and 3 respectively taken from "Direct Delivery" project. Diagrams / parts of diagrams are results of conversion of business rules with mapping defined in 5.2.3.

Business rule 1

It is obligatory that an employee inspects package and an employee chooses a transport type that transports package.
Figure 5.26: BBR 1 - Direct Delivery project - a part of AD


Figure 5.27: BBR 1 - Direct Delivery project - SD

Chapter 5. Results

Figure 5.28: BBR 1 - Direct Delivery project - a part of SMD for package


Figure 5.29: BBR 1 - Direct Delivery project - a part of SMD for transport type


Business rule 2

It is prohibited that a package is transported by a drone if package is transported to Logistic Center.
Figure 5.30: BBR 2 - Direct Delivery project - a part of AD


Figure 5.31: BBR 2 - Direct Delivery project - SD

Chapter 5. Results

Figure 5.32: BBR 2 - Direct Delivery project - a part of SMD for package


**Business rule 3**

It is permitted that a package is transported by a drone only if a package is packed into a shipping container and a package is packed by an employee.

Figure 5.33: BBR 3 - Direct Delivery project - a part of AD

Guidelines provided in this section answer RQ3 stated in 1.3. They cover properties defined in section 5.1 and are intended to help in specification of BBRs which can be automatically converted to AD, SD and SMD according to the conversion rules presented in 5.2. It means that we cannot use them to ensure that specified BBR is automatically convertible in general but according to conversion rules in 5.2. Each guideline presented in this section covers exactly one property.
1. Every time BBR can be executed under conditions, write it with if keyword.
   This guideline covers first syntax property defined in 5.1.1. Justification is same as for the property. By ensuring that BBR uses if keyword we can apply conversion rules CR_AD4, CR_SD5, CR_SMD4 presented in 5.2.3.

2. Every time same subject is responsible for different activities, repeat it in the business rule - do not shorten "A does B and makes C" to "A does B and makes C".
   This guideline covers second syntax property defined in 5.1.1. Justification is same as for the property. By ensuring that BBR is not shortened version we can match BNF pattern presented in 'Domain description' 5.2.1. By matching the pattern we can use conversion rules defined in 5.2.3.

3. Keep the order of activities in BBR when You connect them with and keyword.
   This guideline covers first semantic property defined in 5.1.2. Justification is same as for the property. By ensuring that BBR uses and keyword we can apply conversion rules CR_AD3, CR_SD4, CR_SMD3 presented in 5.2.3.

4. Use transitive verbs instead of non-transitive verbs if it is possible.
   This guideline covers second semantic property defined in 5.1.2. Justification is same as for the property. By having BBR that uses transitive verbs instead of non-transitive we can apply conversion rules CR_AD2, CR_SD2, CR_SD4, CR_SMD1, CR_SMD4 presented in 5.2.3.

5. Link hypothesis facts and conclusion facts with at least one common term.
   This guideline covers third semantic property defined in 5.1.2. Justification is same as for the property.

6. For every fact that uses transitive binary verb define its synonym so that active and passive form of verb is listed in both: fact and its synonym.
   This guideline is strictly connected with the guideline no. 4 and therefore covers second semantic property defined in 5.1.2. We don’t want only to have transitive verbs in BBRs but we also want to have both forms available: active voice to perform conversions: CR_AD2, CR_SD2 and passive voice to perform conversions CR_AD2, CR_SD2 and CR_SMD1. Both forms are stored in dictionaries as synonyms of facts (see list of facts in 10). Dictionaries with list of facts and terms which are used in definition of rules is necessary to define business rules in SBVR notation (see ’Annex G - EU-Rent Example’ 10 - annex of SBVR specification).
In section 5.1 were defined properties required by BBRs if we want to automatically convert them to UML diagrams according to mapping in section 5.2. There might be BBRs which don’t have properties from section 5.1 and can be automatically converted to UML behavioral diagrams: activity, sequence and state machine. To discover existence of another properties, which ensure that BBR is automatically convertible, industrial investigation could be conducted.

Due to limited access to business analysts and limited access to industrial BBRs (only Capgemini project Delivery Project was available) mapping functions in section 5.2.3 cover only restricted inputs (e.g. or and quantification keywords like at most, at least are not covered). Proposed mapping in section 5.2.3 is only one possible mapping. It is possible that for BBRs which match BNF pattern defined in section 5.2.1 another reasonable mapping functions can be defined.

M1 provided in section 2.3 discuss conversion of SBVR SE constructs to constructs of AD and SD. M2 provided in this master thesis covers conversion of SBVR SE to AD, SD and SMD. M1 SD provides mapping for subject term, object term and facts. M2 AD covers SBVR subject term whereas M1 AD does not. M2 AD M2 SD provides mapping for same constructs and in addition for: and keyword. M1 SD is not described in details and seems to be draft of mapping rather than ready solution. There are no examples of SDs enclosed in [17], which were created based on M1 SD. M2 provides detailed description with graphical examples (see 5.2.3). M1 does not provide mapping for SMD and M2 does. More details of comparison between M1 and M2 can be found in section 5.2.4. M2 provided in this thesis defines BNF pattern (see 5.2.1 which allows to create parser for BBRs (M1 does not provide BNF pattern). The parser with M2 allows to automate conversion of BBRs to UML diagrams (AD, SD and SMD).

Guidelines presented in section 5.3 were intended to help in specification of BBRs which can be automatically converted to AD, SD and SMD but they were not tested in real life practice of business analyst. Presented guidelines are not general - they are intended to support specification of BBRs which are automatically convertible according to conversion rules presented in the section 5.2.3.
Chapter 7

Conclusions and Future Work

Research questions defined in section 1.3 are answered (see chapter 5) and objectives defined in 1.2 are reached.

Conversion of BBR is still not deeply explored field. Properties defined in section 5.1, mapping of SBVR SE constructs to AD, SD, SMD constructs provided in section 5.2 and guidelines from section 5.3 should be validated in industry. Research was limited due to a small number of business analysts (only 4) who agreed to take part in the research, small number of BBRs included in the form for business analysts (see Appendix A) and limited access to BBRs taken from industry (only Capgemini Poland project "Direct Delivery"). Industrial validation could be performed in a following way:

1. BBRs taken from at least 2 industrial projects are extracted and stored in SBVR SE notation. Extracted BBRs cover different types of BBRs (see "SBVR formal specification" in section 2.1) and sufficiently big number of BBRs is extracted. Someone not involved in details of research checks diversity of extracted BBRs.

2. Extracted BBRs stored in SBVR notation are filtered - remain only these which fit to BNF representation defined in 5.2.1. In case when no BBRs remain, extraction process defined in step 1 is executed again.

3. BBRs which remain after filtering are converted to AD, SD, SMD according to mapping defined in 5.2.

4. at least 4 business analysts: 2 industry workers from different companies and 2 university lecturers (field of business analysis) are asked to assess and comment diagrams converted in step 3 given BBRs from step 2.

Negative assessment and comments from business analysts fetched in step 4 of validation are input for redefinition of M2. Changes in M2 affect properties 5.1 and guidelines 5.3.

In case there is a broader access to business analysts from industry, there is more
than 1 project from industry available for extraction of BBRs and there is sufficiently large number of diverse (see "SBVR formal specification" in section 2.1) BBRs extracted from these projects, method described in [3] can be re-executed to improve results.
Chapter 8

Dictionary

This chapter contains concepts and their definitions which are very important for this master thesis research.

- **business analysis** - "the set of tasks, knowledge, and techniques required to identify business needs and determine solutions to business problems. Solutions often include a systems development component, but may also consist of process improvement or organizational change" [7]

- **business analyst** - person that is performing business analysis. He/she is not only responsible for investigation of business systems (holistic view of systems) and evaluation of actions to make improvements in business system operations [16]. Business analyst is also responsible for documenting "the business requirements for the IT system support using appropriate documentation standards" [16]

- **business rule** - "a statement that defines or constrains some aspect of the business" [1]. Von Halle describes business rules as following: "business rules are the ultimate levers with which business management is able to guide and control the business. In fact, the business’s rules are the means by which an organization implements competitive strategy, promotes policy, and complies with legal obligations" [21]. Examples of business rules: "golden customer can buy in shop without waiting in queues for products", "special offer period starts in November and ends in December", "car cannot be rented to person without qualifications".

- **fact** - "Definition: proposition that is taken as true

Note: How one ascertains what is true, whether by assertion, observation, or other means, is outside the scope of this specification. However, taking a proposition as true must be consistent with epistemic commitment. The concept fact’ is here defined to be consistent with the operations of truth-functional logic, which produce results based on true and false." [8 p.28].
• SBVR operative business rule - ”a prescribed guide for conduct or action”\[8\] eg. ”his parents laid down the rule that he must do his homework before going out to play”\[8\]. It is called also behavioral business rule\[8\].

• term - ”verbal designation of a general concept that is in a given subject field” \[8, p.61\]. ”term denotes thing
  Definition: the thing is an instance of the concept that is represented by the term” \[8, p.36\]

• thing - ”the thing is the instance of the individual noun concept that is represented by the name” \[8, p.36\].
Chapter 9

Bibliography


Chapter 10

Appendix A: Form for business analyst

For each behavioral business rule given on the list (see below) convert it into fragments of UML diagrams: (1) activity, (2) state machine, (3) sequence.

Each business rule should be represented by 3 types of UML diagrams. It may happen that some rules are not transformable into some kind of UML diagram - in such case, please describe why do you think it is not transformable. Don’t worry that you cannot create whole diagram (e.g., activity diagram from its initial node to activity final node). Try to extract as much as it is possible from each behavioral business rule.


LIST OF BEHAVIOURAL BUSINESS RULES:

1. It is obligatory that a variable life insurance policy which is issued by an insurer, complies the minimum requirements.

2. It is obligatory that each variable life insurance policy is filed with the Commissioner and approved by the Commissioner.

3. It is prohibited that an insurer issues a variable life insurance policy if the insurer is not licensed to do a life insurance business.

4. It is permitted that the Commissioner approves life insurance business that is run by an insurer only if financial condition that has an insurer is sufficient.

5. It is obligatory that an insurer provides illustration which is dedicated for prospective purchaser if the life insurance policy is identified to be used with an illustration.
BUSINESS RULE NUMBER: ......

Type of diagram: activity diagram

Type of diagram: sequence diagram

Type of diagram: state machine diagram
Terms:

1. **Business** - implicitly understood.

2. **Commissioner** - individual **person** in US state who establishes **insurance policy regulations**. Here it is West Virginia insurance commissioner.

3. **Financial condition** - may be **hazardous** or **sufficient**. Property of **insurer**. Implicitly understood.

4. **Illustration** - a presentation or depiction that includes guaranteed and non-guaranteed elements of a **life insurance policy** over a period of years.

5. **Insurance policy** - contract between **insurer** and **insured**. Issued by **insurer**. Issued for **insured**. Defines terms on which indemnity cover is provided.

6. **Insurance policy regulations** - regulations which **insurance policy** has. These regulations are established by **Commissioner**.

7. **Insured** - **person** that has **insurance policy**.

8. **Insurer** - **person** that issues **insurance policy** and runs **life insurance business**. **Insurer** has **financial condition**

9. **Life insurance business** - category of **business** that is run by **insurer**.

10. **Life insurance policy** - category of **insurance policy** that can be identified to be used with **illustration**.

11. **Minimum requirements** - list of requirements which **variable life insurance policy** complies. **Commissioner** defines **minimum requirements**.

12. **Person** - means an individual, corporation, partnership, association, trust or fund eg. **insurer**, **insured**.

13. **Prospective purchaser** - **person** that buys **insurance**.

14. **Variable life insurance policy** - category of **life insurance policy** that complies **minimum requirements**.

Facts:

1. **Commissioner approves life insurance business**
   - Synonym: **life insurance business is approved by Commissioner**
2. **Commissioner** defines minimum requirements
   - Synonym: minimum requirements *are defined by* Commissioner

3. **Commissioner** establishes insurance policy regulations
   - Synonym: insurance policy regulations *are established by* Commissioner

4. financial condition *being hazardous*
   - Definition: the financial condition *is* hazardous

5. financial condition *being sufficient*
   - Definition: the financial condition *is* sufficient

6. illustration *is dedicated for* prospective purchaser

7. insurance policy *has* insurance policy regulations

8. insurance policy *is issued for* insured
   - Synonym: insured *has* insurance policy

9. insurance policy *is issued by* insurer
   - Synonym: insurer *issues* insurance policy

10. insurer *has* financial condition

11. insurer *is licensed to do* life insurance business

12. insurer *runs* life insurance business
   - Synonym: life insurance business *is run by* insurer

13. insurer provides illustration
   - Synonym: illustration *is provided by* insurer

14. life insurance policy *is identified to be used with* illustration

15. prospective purchaser *buys* insurance policy.
   - Synonym: insurance policy *is bought by* prospective purchaser.

16. variable life insurance policy *complies* minimum requirements
17. variable life insurance policy is approved by Commissioner
   
   • Synonym: Commissioner approves variable life insurance policy.

18. variable life insurance policy is filed with Commissioner
Chapter 11

Appendix B: UML class diagram extracted from facts

Figure 11.1: UML class diagram of insurance domain
Appendix C: Expert 2 Raw data
Chapter 12. Appendix C: Expert 2 Raw data

BUSINESS RULE NUMBER: 2

Type of diagram: activity diagram

Type of diagram: sequence diagram

Type of diagram: state machine diagram
BUSINESS RULE NUMBER: 4

Type of diagram: activity diagram

Type of diagram: sequence diagram

Type of diagram: state machine diagram
Expert 2 was asked to repeat conversion of rules into sequence diagrams because initial transformation was Platform Specific Model.
Chapter 12. Appendix C: Expert 2 Raw data
Chapter 13

Appendix D: Expert 3 Raw data

Rule 1. It is obligatory that a VLIP which is issued by an insurer comply the minimum requirements.

Discussion: kind of assertion; assumed semantics: to be issued must satisfy the condition.

Activity diagram

State machine for VLIP

Ev

Rule 2. It is obligatory that each VLIP is filled with the commissioner and approved by the commissioner.

Activity diagram

State machine for VLIP

Page 104
Chapter 13. Appendix D: Expert 3 Raw data

3. It is prohibited for an insurer to issue a VLIP if the insurer is not licensed to do a life insurance business.

Activity diagram:

- Insurer
- Issue (VLIP)
- Not licensed to do a life insurance business

Sequence diagram:

- Insurer
- Issue (VLIP)
- Insurer is licensed to do a life insurance business

State machine for VLIP:

4. It is permitted that the Commissioner approves life insurance business that is run by an insurer only if the financial condition that has an insurer is sufficient.

Activity diagram:

- Commissioner
- Financial condition
- Insurer
- Insurer is sufficient
- Approved

Sequence diagram:

- Commissioner
- Financial condition
- Insurer
- Insurer is sufficient
- Approved

State machine for life insurance business:

- Financial condition that has
- Insurer is sufficient
- Approved
Chapter 13. Appendix D: Expert 3 Raw data