

# Integration of Use Case Models and BPMN Using Goal-Oriented Requirements Engineering

Usman Zafar<sup>1\*</sup>, Moshiur Bhuiyan<sup>2</sup>, P. W. C. Prasad<sup>1</sup>, Farzana Haque<sup>2</sup>

<sup>1</sup>School of Computing and Mathematics, Charles Sturt University, Sydney, NSW 2010, Australia.

<sup>2</sup>Service Consulting, Enterprise Cloud Systems Pty Limited, Sydney, NSW 2560, Australia.

\* Corresponding author. Email: a.krishna@curtin.edu.au

Manuscript submitted January 10, 2017; accepted April 8, 2017.

doi: 10.17706/jcp.13.2.212-221

---

**Abstract:** This paper proposes a methodology that transform Use Case diagram into BPMN diagram using GORE technique. The success of software projects depends upon the stakeholder requirements satisfaction and support for organizations business process. Use Cases is one of the most commonly used notations to elicit requirements from an actor's point of view. Business process modelling notation (BPMN) is commonly used to model organizational business processes and to effectively communicate and conceptualize configuration of processes to relevant stakeholders. Hence, there is a need to be able to transform actors' goals and objectives into operational business processes and vice versa. Goal oriented requirement engineering (GORE) suggests analysing high level goals of the system to contextualize business processes. This paper argues that both Use cases and BPMN should be contextualized and should coevolve.

**Key words:** BPMN, GORE, UML use case, business process modelling, transformation, requirements elicitation.

---

## 1. Introduction

Modelling is fundamental in most engineering disciplines. Modelling is also considered fundamental for software engineering for ensuring efficiency, quality, stakeholder satisfaction and project success. Several different modelling diagrams are developed which can be used for design activities and requirements engineering. Common modelling languages are State Diagram, Use Case Diagram, Class Diagram and Sequence Diagram these languages are part of UML (Unified Modelling Language) [1]. The process of requirements elicitation of software projects is becoming complex with an increase in the stakeholder demands, expectations and change of requirements. This increases the need for modelling and visualizing requirements. Use Cases are one of the most popular ways to model requirements. They describe the interaction of users (actors) with the system [2]. From the user point of view to elicit software requirements, Use Cases are the key medium [3]. Requirements elicitation using the Use Cases help to increase participation of customers and this process also supports iterative development. During review process, visual representation of use cases is extremely helpful [4]. The problem with Use Cases is that as more functionality is described in the overview of the whole system, the execution orders of Use Cases and their dependency is lost. In order to restore the global overview of the Cases and order the Use Cases depending upon precondition and post condition, Business process model notation (BPMN) is commonly used in the industry [2]. Business modelling helps in describing business process which is important to the system being developed or to understand the organizations processes. Business processes can be modelled

using BPMN; BPMN makes the system easier to understand for technical and non-technical users [5]. The notations for BPMN are categorized in many fields based on the conceptual features. In all analysis, one common principal can be recognized which is that some notations are more audience centric i.e. technical or non-technical audience. Other notations focus on specific aspects neglecting the other aspects of business process. This increases the needs of an enterprise view for the business process to enhance and maintain the ability of the rich models to understand and communicate business process [6].

To capture the context of software systems in which processes are performed, Goal-Oriented Requirements Engineering (GORE) is used. The strategic goals that an organization is pursuing and the hierarchy of these goals into objectives for which actors are assigned in the organization rational is provided by the high level business requirements for more detail system requirements. In the last decade for different Business Process (BP) management activities GORE technique is prescribed by several methods [7]. Thus this makes GORE important for the success of BPMN.

While this work is proposing a methodology to transform Use Case diagram into corresponding BPMN diagram using GORE technique, this methodology transforms visual models by keeping system context intact and analysing requirements using both important models for requirements engineering, so the gap between requirements can be minimized and software designers can analyse the system using both modelling languages with less resources and time, but with efficient results.

## **2. Background**

### **2.1. Related Work**

There is some related work on Use Case Model and Business process model, generation of these models and existing transformation between these two models. There are some other transformations as well which translate some other models into BPMN model.

Cockburn suggests possibility of getting business processes by applying Use Cases [2]. He proposed a template in [3] but he has not explained any rules on how to proceed from template. An approach which uses a complete text as a label to create a mapping between a template of Use Case and activity using the Use Case flow in which each step is transformed to activity diagram. The problem with this approach was it does not produce end to end business process [8]. Another approach which transforms existing Use Case template into Event Process Chain (EPC) representation, to drive an end to end scenario textual string such as precondition and post-condition has been used [9]. An approach proposed by [10] which translates RAD (Role Activity Diagram) process into corresponding BPMN model. This approach based on two stages in the first step they identify notations of BPMN for the construction of RAD modelling. Then they introduce an algorithm which translates RAD to BPMN. This algorithm gets the input of RAD business process and generates the BPMN model by interpreting RAD model visually.

All these approaches do not transform Use Case diagram to BPMN. They transform textual Use Cases to BPMN and not a single approach evaluates or improve the existing case before translating it to the BPMN. These approaches do not analyse system high level goals and they don't inspect what is required to achieve the goal. The other important factor is that none of these approaches used the GORE technique as we discussed previously. GORE is used to capture the context of business process and it creates a hierarchy of goals which are transformed into objectives for which actors are assigned in the organization, so a BPMN diagram created without GORE should be lacking some context and missing high-level goals. This can result in incomplete documentation or inappropriate requirements, and translating Use Cases on the basis of precondition and post-condition can result in processes which are not important to the system. The work in this paper will use GORE to identify the high level goals which are the core functionality of the system it is supposed to perform and inspect and find out what is required to achieve the goal. We will also write some

textual description of the Use Cases before translating Use Case diagrams to BPMN diagrams and we will use scenarios, triggers, and pre-condition, post-condition to connect the process in the BPMN diagram.

### 2.2. Models and Transitions

Software development based on models rather than code can raise abstraction level, because it allows analysts or developers to deal with complex problems within the application domain [11]. However, these modelling languages have different use. For example, Use Case Diagram is used to model user requirements and BPMN diagrams are used to model organization process. Sometimes we need both models to get the required result. Having a transition method which transforms one model to another can benefit a lot to system analyst, designers and requirements elicitation team. Transforming models depending upon particular steps or programs is called transformation. Transformation creates the correspondence between source model elements to the target model elements of a certain model language [11]. Different transformation exists which transform different models into BPMN model [12]-[14].

### 2.3. Use Case Diagram

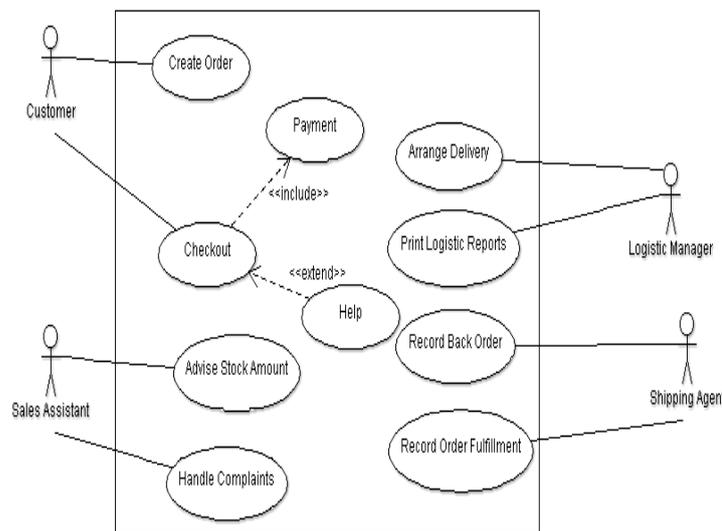


Fig. 1. Initial use case model.

Table 1. Use Case Elements

Use Case Element	Description
	The interaction between the external entities and the system is a use case diagram these external entities are considered as actors which can be a human user external system or application.
	This ellipse is use to represent a use case. Use case represent a single functionality system performs this represent the user needs or the tasks user perform using the system.
	Relationship among the use case and actor is represented by straight line which is also called association this tells us which actor use which functionality or which actor performs which task.

Use Case describes interaction of an actor (user) with the system. Use Case is related to actors and the actions performed by these actors. Event analysis helps to identify goals and actors, what they do, who they are and other analysis which aid Use Case search process [15].

Use Case diagrams provide system functionality overview [15]. Use Case diagrams are also used to gather design requirements of a system including the requirements of the internal and external users. To gather the functionalities of the system which is being analysed, actors are identified and use cases are prepared [16]. Initial Use Case diagram which we will be using throughout our research is shown in Fig. 1.

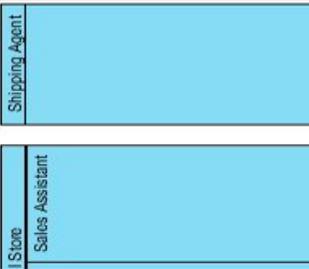
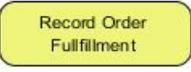
Some of the main elements of the Use Case Diagram are shown in Table 1.

### 2.4. Business Process Modelling Notation (BPMN)

BPMN is used by organizations to communicate and understand internal procedures of a business with a graphical notation [17]. Software system developers usually face the challenge of developing a system that aligns with organizations business process. They must understand business processes which require more collaboration with analysts and domain experts, because business modelling notation BPMN is used to express a business process in a Business Process Diagram. BPD diagram is usually created by a flowcharting technique which is to some extent similar to Activity Diagram of UML [18]. The purpose of BPMN is to provide a unique notation which is understandable to the organization stakeholders and which facilitates interaction between designers, business developers, analyst and technical manager who will automate the organization process. BPMN can be used to model internal as well as public business process. BPMN model can be used to model a part of the business process as well as a whole business process, processes modelled at discrete fidelity levels. The description of a process model based on a general point of view and interactions between various participants should be shown [18].

A BPMN diagram consists of small elements which make it understandable for technical and non-Technical observers. BPMN elements are categorized in four categories: flow objects, connecting objects, swim-lanes and artefacts [19]. Some elements of BPMN diagram are shown in the Table 2.

Table 2. BPMN Diagram Elements

Notation	Description
	<p>Pools are used for partitioning set of activities from other pools. Lanes exist inside the pool which categorize activities lanes extend full length of the pool either horizontally or vertically. In our case Retail Store is a pool which contains Sales Assistant. Another pool is Shipping agent which is an external actor.</p>
	<p>An event is represented by a circle events have a trigger or flow of or an impact. Green circle represents the start event and its trigger can be anything For example, Customer creates an order. A circle with bold border and pink color represent an end event which shows the process is completed.</p>
	<p>A gateway is used to control the decision making including merging and joining of paths and it also control sequence flow divergence and convergence.</p>
	<p>An activity which is a generic term for work that will be performed by a designated participant an activity can be atomic or non-atomic.</p>

### 2.5. Goal Oriented Requirements Engineering (GORE)

For requirements elaboration, elicitation, documentation and negotiation Goal Oriented Requirements Approach (GORE) refers to high level goals [20]. Goal oriented approach is widely used in requirements engineering for requirements elicitation. Its primary purpose is to answer why the system is being constructed expresses the rationale and the proposed systems justification. Instead of focusing on specific requirements goals help stakeholders in more communication with the stakeholders and in a language which analyst as well as stakeholder can understand [21]. Goals are actually the useful abstract of stakeholders' interest, needs and expectations and they provide a realistic view of the system and an effective way to analyse requirements [22]. Goal Oriented approach helps in early and late requirements for the system. Early modelling approach analyses stakeholder interests and needs whereas late modelling approach analyses the future objectives and how these objectives can be operational as system components [23]. Goal Oriented Requirements Engineering (GORE):

Goal is a specific and measurable target intended to be achieved at a given point in time to satisfy the stakeholder's objective [24]. Understanding and negating the stakeholder's problem, we can also extract intention statements from the following: [25] Enterprise policies, Mission Statement of enterprise, Goals set by enterprise, Diagrams which shows the enterprise workflow, Scenarios written or explained by stakeholders, disclosed in requirements documents, Analysis of similar or current system, elaborating other goals. A goal model is a hierarchy which relates high level goals with low level system requirements. Hard goal describes the functionality of the functions the different tasks system will perform and a defined set of criteria for satisfaction e.g., User can check Order status. A soft goal is a goal describing the systems desired qualities which do not have any clear criteria for achievement e.g., our goal was to update order and the order is updated but there is a disagreement between some stakeholders whether it is properly updated or not [26]. We should use goal model in the Initial stage of requirements engineering. When focus is on identifying problems, inspecting solution for the system and alternatives, always use before UML modelling.

### **3. Proposed Methodology**

We are proposing a methodology for obtaining high level goals of a system by using the Goal Oriented Requirements Elicitation (GORE) approach. GORE emphasis more on understanding the organizations context, why this system was needed and do the requirements satisfy the stakeholder's needs. Goal identification is not an easy process goal and can be available in documents or stakeholders can explicitly state goals. But most probably goals will be implicit and goal elicitation process should be taking place. Initial analysis of current system or Organization can be used for goal identification [27]. Goals can also be identified from documents, observing existing system and reviewing interview reports. It is also observed that stakeholders explain their requirements in terms of actions or operations performed daily rather than goals. So we can identify goals by looking for action words. [28]. After applying Goal Oriented Approach and identifying high level goals we will transform Use Case model into BPMN model to verify our requirements and to identify the gap between requirements.

#### **Steps to Follow for the Methodology:**

##### **Step 1: Analyse use cases to obtain high level goals**

This step identifies actors of the system. Actors can be all the individuals or other systems who interact with the system. Identify High level goals by analysing the major expectations or requirements of the stakeholders which should be satisfied by the system, identify goals alternative e.g. what will happen if we are unable to achieve a particular task for example unable to process payment; then we will ask user for providing account details again until system approves the payment and at the end goals which are specific to domain for example "Create Order", "Print Logistic Report", "Advice Stock Amount" etc.

This step inspects in detail what it really need to satisfy goal in our domain of retail we can look into the

things e.g. What products customer demand, sufficient inventory in the store, examine that Logistic reports are printed in the expected manner, customer card details are correct and funds are enough to complete the order.

### **Step 2: Update use case diagram**

In this step we will update our use case diagram depending upon the outcome of step 1 and step 2.

### **Step 3: Develop detailed use case description**

In this step we will create a template to write Use Case details. The template should identify Scenario, precondition, trigger, success condition and alternatives. We proposed a template in methodology implementation in Table 3.

### **Step 4: Sequence refined sub-goals using scenarios**

To get the process we have to sequence sub-goals which we refined in the previous steps using scenarios. We have to consider each scenario of the system so that we can refine sub-goals of that scenario to achieve high level goals.

### **Step 5: Sequenced goals represent process**

The scenarios we analysed to identify sub goals of the system actually represent each process of the system in BPMN. Sub goals are actually subtasks in the BPMN process. If we sequence these goals depending upon the scenarios for example, which scenario will start the whole process? What will happen next? And how will this process complete? All these questions will help in sequencing goals. These questions can be answered by different stakeholders of the system.

### **Step 6: Connect identified process**

In this last step we will connect process/Tasks in each pool of the BPMN diagram depending upon the pre-condition, post-condition and trigger of the processes.

## **4. Methodology**

This project is using the Use Case model shown in Initial Use Case Diagram (Fig. 1). In this implementation Use Case model will be transformed to BPMN mode. The whole process will follow the following steps.

### **Step 1: Analyse use cases to obtain high level goals**

First step will identify actors in the Fig. 1 which clearly shows that they are all main actors except from one "Supplier". Without this we cannot understand the whole case because they deliver the goods. Each actor represents a pool in the BPMN diagram so we will create pools with the Supplier name, customer, shipping agent etc. as shown in Fig. 3. Now to identify high level goals we analyze Fig. 1. We can see that some tasks performed by customers are missing, for example: a customer can change his/her mind and cancel the order and, before creating the order, the customer has to register to be able to create orders. As we added a new actor it will also have one goal which is Deliver Goods.

### **Step 2: Inspect what is really needed to satisfy the goal**

If we inspect the use case diagram in Fig. 1 on the basis of the questions in the methodology, what products do customers demand? Do the stores have sufficient inventory? this arises questions like the sales assistant can search for the products which are more in demand and also by giving the sales assistant and customer an option of search product which will tell the user which product is available or not and how many products are still in stock.

### **Step 3: Update use case diagram**

The updated use case diagram of the system is shown in the Fig. 2.

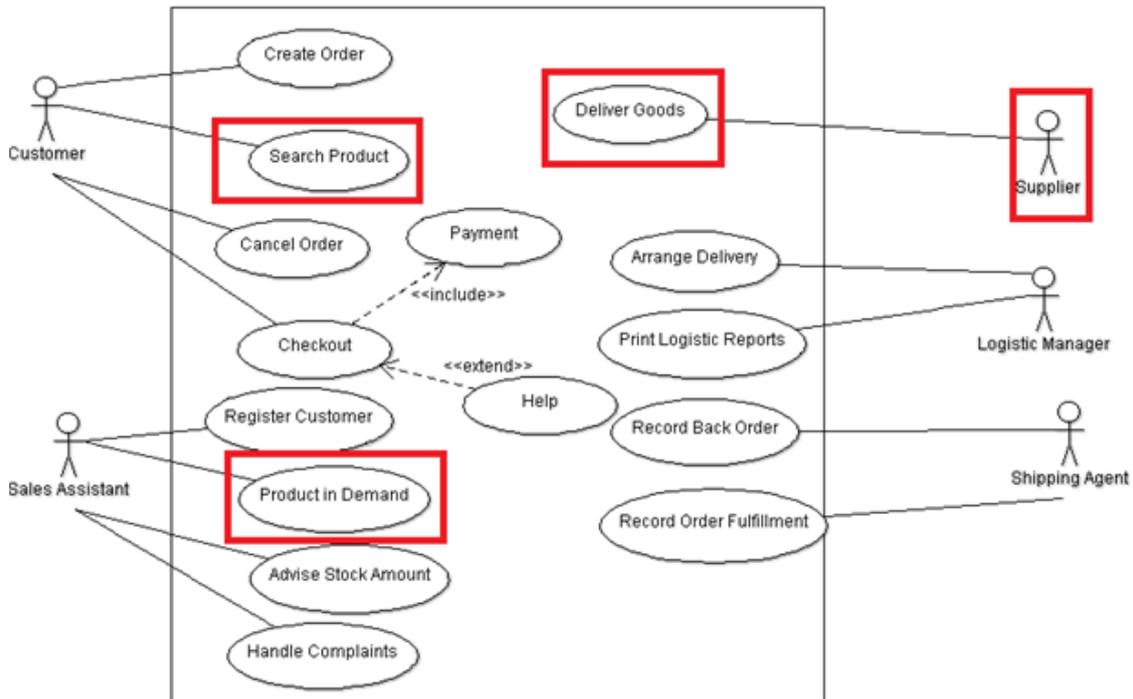


Fig. 2. Updated use case diagram.

The Fig. 2 shows that we identified one actor and three use cases that might be very important for the success of the system.

**Step 4: We are using the template which almost covers all the requirements of the proposed methodology.**

Table 3. Use Case Detail Template

<b>Use Case Name:</b>	Create New Order
<b>Actor(s):</b>	Customer, Order Clerk
<b>Scope:</b>	Order Entry Subsystem
<b>Goal Level:</b>	User goal
<b>Success End Condition:</b>	Order successfully created
<b>Precondition:</b>	Customer already exist
<b>Trigger:</b>	Customer click on Create Order
<b>Scenario:</b>	Customer is placing an order
<b>Main Success Scenario:</b>	Add one or more products to the order System will add discounts on individual items or on entire order System will add shipping rates and applicable taxes User will enter the payment details System will verify the payment details and will accept the payment
<b>Alternative:</b>	5a. Payment Details are invalid 5a1. System will prompt to the user payment details are incorrect. 5a2. User will enter the details again 5a3. System will verify the payment details

From the template above we can clearly identify the trigger, and who will trigger the use case, who will execute and how this use case will be completed and what output it will give to us.

**Step 5: Sequence refined sub-goals - possibly using scenarios**

To get the process we have to sequence sub-goals which we refined in the previous steps using scenarios. Let's consider the scenario "Create Order" which starts the whole process: when user creates the order

this will automate the system as Customer can create his/her own order without sales assistant help.

Another scenario is Logistic Manager Arrange delivery: he/she can use the system to assign staff who will arrange delivery and make sure the store don't run out of the products it's also a system use case.

**Step 6: Sequenced goals represent process**

The scenarios we analysed in the step 5 are actually goals of the overall system and processes in the BPMN. The sub goal representing the tasks needs to be performed to complete the one process in BPMN. As we identified a pool for each actor, in the step 1 a pool contains all the process/Tasks performed by an actor. Now depending upon these questions for example which scenario will start the whole process? What will happen next? And how will this process be complete? We can identify the order for the scenario/process, how this system will start and how all the process will be completed. In our case the whole process will initiate when user will create an order and the process will end when Goods or order is delivered to the customer.

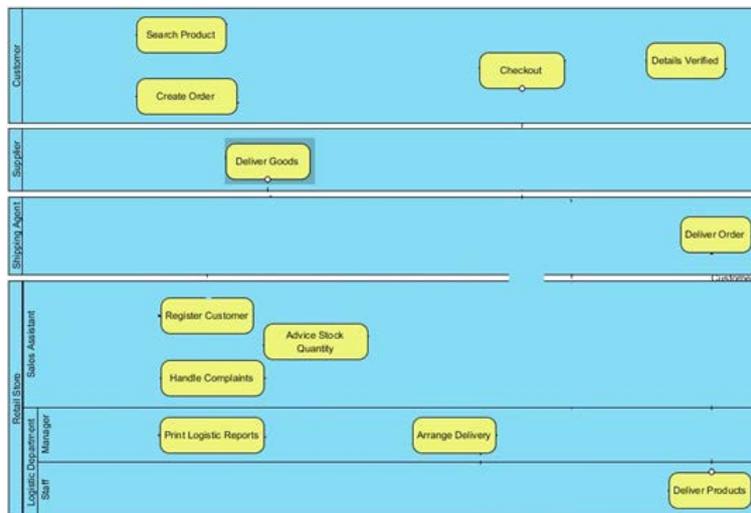


Fig. 3. Pools and processes.

**Step 7: Connect identified process**

To connect the process identified in the Fig. 3 we have to use the pre-condition and post-condition of the use cases which we identified in the step 4. For Create order it tell us this process begins when customer will create order or search for product and after adding products to the order, the customer will checkout for that. Customer's details will be verified before completing the process. A complete diagram is shown in the Fig. 4.

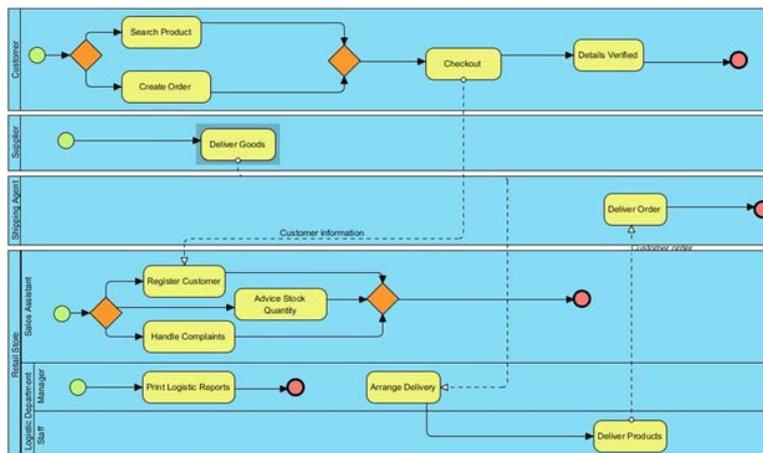


Fig. 4. Transformed BPMN model.

## 5. Conclusion

In this paper we proposed a new methodology which transforms Use Case model into corresponding BPMN model using GORE approach. This methodology enables the analyst to analyse the requirements from users point of view and then transform it to operational business processes in organisational context. In the future we plan to extend our work to include transformation of BPMN models into corresponding Use Case models. We also plan to apply this methodology in an industrial project scenario and assess its effectiveness and performance with the comparable methodology and techniques.

## References

- [1] Nisreen, I. (2012). An ontology for software requirements modelling. *IEEE*.
- [2] Lübke, D., & Schneider, K. (2008). Visualizing use case sets as BPMN processes. *Barcelona: IEEE*. 21-25.
- [3] Cockburn, A. (2001). Retrieved from <http://alistair.cockburn.us/get/2465>
- [4] Sinha, A., & Paradkar, A. (2010). *Use Cases to Process Specifications in Business Process Modeling Notation*. Miami: IEEE.
- [5] Tanuska, T. (2011). *The Proposal of Functional User Requirements Generation*. Shanghai: IEEE.
- [6] George, K. (2007). *Combining i\* and BPMN for Business Process Model Lifecycle Management*.
- [7] Geert, P. (2013). *Investigating Goal-Oriented Requirements Engineering for Business Processes*.
- [8] Gutiérrez, C. (2008). *Visualization of Use Cases through Automatically Generated Activity*, 83-96.
- [9] Lubke, D. (2006). *Transformation of Use Cases to Epc Models*, 137-156.
- [10] Yousef, R., Odeh, M., Coward, D., & Sharieh, A. (2009). *Translating RAD Business Process Models into BPMN*, 75-83. London: IEEE.
- [11] Bertero, A., Silvestre, L., & Bastarrica, M. (2012). *T2M and M2T Transformations between Software Processes and Software Process Models*. Valparaiso: IEEE.
- [12] Luba, S., Farzana, H., Moshiur, B., Aneesh, K. (2014). Risk measure propagation through organisational network. *COMPSAC Workshops*, 217-222.
- [13] George, K., Aleksandar, V., Moshiur, B., Aneesh, K., & Aditya, G. (2006). Combined approach for supporting the business process model lifecycle. *PACIS*, 76.
- [14] Anna, M., Moshiur, B., Farzana, H., Luba, S., Aneesh, K., & Chandana, W. (2015). Citizen's charter driven service area improvement. *APSEC*, 401-408.
- [15] Latifaah, R. (2012). *Syntax-Driven Semantic Analysis for Constructing Use Case Diagrams from Software Requirement Specifications in Indonesian*.
- [16] Point, T. *UML - Use Case Diagrams*. Retrieved from [http://www.tutorialspoint.com/uml/uml\\_use\\_case\\_diagram.htm](http://www.tutorialspoint.com/uml/uml_use_case_diagram.htm)
- [17] OMG. Business Process Model and Notation. Retrieved from <http://www.bpmn.org/>
- [18] Fazziki, A. E., Lakhrissi, H., Yetognon, K., & Sadgal, M. (2012). *A Service Oriented Information System: A Model Driven Approach*, 466-473. Naples: IEEE.
- [19] White, S. *Introduction to BPMN*. Retrieved from <https://www.google.com.au/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKewiCnObhpLbMAhVEkpQKHUoZAmUQFggkMAE&url=https%3A%2F%2Fwww.ibm.com%2Fdeveloperworks%2Fcommunity%2Ffiles%2Fbasic%2Fanonymous%2Fapi%2Flibrary%2F7624eb5a-089a-41bf-9b71>
- [20] Arfan, M. (2015). *Fuzzy Based Evaluation of Software Quality Using*.
- [21] Annie, I. & Antón, J. (2000). *Deriving Goals from a Use Case Based Requirements Specification for an Electronic Commerce System*.
- [22] Raian, A. (2010). A Goal-Based Framework for Contextual Requirements Modeling and Analysis.

- [23] Babar, A., Wong, B., & Gill, A. (2011). *An Evaluation of the Goal-Oriented Approaches for Modelling Strategic Alignment*. Gosier: IEEE.
- [24] Chambers. Goal. Retrieved from <http://www.chambers.com.au/glossary/goal.php>
- [25] Regev, G., & Wegmann, A. (2005). Where do goals come from: The underlying principles of goal-oriented requirements engineering. *IEEE*.
- [26] Cheng, D. (2006). Requirements Engineering. Retrieved from <http://www.cse.msu.edu/~chengb/RE-491/Lectures/06-goal-modeling.pdf>
- [27] Lapouchnian, A. (2005). *Goal-Oriented Requirements Engineering: An Overview of the Current Research*.
- [28] Anton, A. (1996). *Goal-Based Requirements Analysis*. Colorado Springs.

**Usman Zafar** graduated from Charles Sturt University, Australia with masters of information technology. He is currently working as a software engineer in IT Industry. His research interests include requirements engineering, conceptual modeling and software design.

**Moshiur Bhuiyan** is an experienced IT management consultant, who possesses extensive expertise in management consulting, business analysis, BPM, change management and enterprise architecture. He has significant passion in research and teaching. His research areas include but are not limited to business process discovery & modelling, process rules and policy integration, process execution, process reengineering and optimization, process lifecycle management, change management, software requirement engineering, cloud computing, ICT governance & architecture. He has published his works in reputed international conferences and journals. He has served as program committee member and reviewer in several conferences and workshops. He is also the founder member of a technology entrepreneurship company named enterprise cloud systems ([www.ecloudsys.com](http://www.ecloudsys.com)) which develops innovative cloud applications.

**P. W. C. Prasad** is an adjunct associate professor with the School of Computing and Mathematics at Charles Sturt University, Australia. Prior to this, he was a lecturer at the United Arab Emirates University in UAE, Multimedia University in Malaysia and also the Informatics Institute of Technology (IIT), Sri Lanka. He gained his undergraduate and postgraduate degrees from St Petersburg State Electrotechnical University in the early 90s and completed his PhD studies at the Multimedia University in Malaysia. He is an active researcher in the areas of computer architecture, digital systems, modelling and simulation. He has published more than 100 research articles in computing and engineering journals and conferences proceedings. He has co-authored two books entitled 'digital systems fundamentals' and 'computer systems organization and architecture' published by Prentice Hall. He is a senior member of the IEEE computer society.

**Farzana Haque** holds a bachelor of computer science and engineering degree and is currently working as a consultant in IT industry. She has a sharp analytical mind, with a powerful leaning towards complex problem solving and always willing to innovate the new things which can improve the existing business and technical processes. Her research areas include conceptual models, business process management, data mining, risk management, and fuzzy logic.