APEM journal
Advances in Production Engineering & Management 6 (2011) 3, 221-231
ISSN 1854-6250 Scientific paper

ROADMAP FOR REQUIREMENTS ENGINEERING PROCESS IMPROVEMENT USING BPM AND UML

Dragicevic, S.*; Celar, S.** & Novak, L.*
*Split Airport, Cesta dr. Franje Tuđmana 96, 21210 Kastela, Croatia
**University of Split, FESB, R. Boskovica 32, 21000 Split, Croatia
E-mail: srdjana.dragicevic@split-airport.hr, stipe.celar@fesb.hr, luksa.novak@split-airport.hr

Abstract:
The roadmap assists stakeholders in their daily requirements engineering work helping them to manage continuously changing and quickly aging requirements. The roadmap for software requirements elicitation that enables better communication between business stakeholders and software developers is described. The proposed roadmap recommends usage of Event-driven Process Chain (EPC) method for business process modelling and UML diagrams for software development. Four roadmap areas are described in the paper. This roadmap is designed to help stakeholders to deal efficiently ("doing things right") and effectively ("doing the right things") deal with requirements decisions. It is based on the four areas which are identified as the foundation of requirements engineering. The paper focuses on the development of customized software, but the same activities, only to a less extent, can be used to implement off-the-shelf software. A case study of roadmap's application at Split Airport (Croatia) is described.

Key Words: Design, Methodologies, Requirements Engineering, Business Process Management, UML diagrams, Event-driven Process Chain (EPC) method

1. INTRODUCTION

There is a high probability for software projects to be cancelled or to significantly exceed available resources. The Standish Group research of the success rate of application of development projects discovers that most software development projects are cancelled before they are completed [1]. It also finds that fewer than 20% of projects are completed on time and on budget. Of the projects that are delivered on time and budget, half fail to deliver the projected functionality. The Standish Group notes that a key cause of such failures is the lack of clarity in eliciting and communicating user requirements.

Leffingwell [2] finds that 40% of the total project budget is related to rework costs triggered by low-quality requirement documents. So, requirements engineering (RE) is considered as one of the most critical phases in software projects [3] and poorly implemented RE constitutes a major risk of projects failure.

In 21st-century business, most enterprises adopted a process approach and Business Process Management (BPM), because the very essence of business performance is based on the effectiveness of processes. A quote from IBM Systems Journal sums it up nicely: "BPM technology provides not only the tools and infrastructure to define, simulate, and analyse business process models, but also the tools to implement business processes in such a way that the execution of the resulting software artefacts can be managed from a business process perspective."[4] Thus, business process models, which are designed to help document, communicate, or improve business processes, are also used for requirements elicitation (capture) as part of software engineering processes. But, process approach and BPM do not guarantee better software requirements.

In reality, experience has shown that business processes and requirements are often misunderstood, or even entirely neglected, leaving resulting systems incomplete in meeting stakeholder requirements. This frequently happens because developers ask business stakeholders to understand requirements and process logic in technical terminology, rather than business logic itself. Another cause of the problem is insufficient attention that is given to defining requirements in complex projects, resulting in loss of quality, cost increases and extension of project deadlines. Time spent in defining requirements is a well spent time. If that time is not invested, then more time and money is spent in fixing problems, rewriting or maintaining erroneous code for missed or incorrectly elicited requirements [5].

The main goal of our research is to enable better communication between business stakeholders and developers. Therefore, we developed roadmap (concept) that includes process approach, BPM and UML models for eliciting and communicating software requirements. This roadmap assists stakeholders in their daily requirements engineering work, helping them to manage continuously changing and quickly aging requirements. The paper focuses on development of customized software, but the same activities, only to a less extent, can be used to implement off-the-shelf software.

The remainder of this paper is organised as follows: in Section 2 we discuss related work. In Sections 3-5 we introduce the mentioned roadmap, briefly describing its main characteristics. In Section 6, we show some extracts from the case study and, finally, we conclude the paper with Section 7.

2. RELATED WORKS

The field of research covers the different areas from an overview of the field of software system requirements engineering (RE) [6], through recommendation for further research in RE field [7], using the BSC method for the selection processes that will be automated [8], using BPM in defining Service Oriented Architecture (SOA) systems and web services [9], [4], [10], using BPM models for eliciting requirements of Model Driven Architecture (MDA) projects [5] to a description of non-functional requirements [11].

It is important that software product helps to meet the enterprise strategic goals. Many enterprises adopted Balanced Scorecard (BSC) method, developed by Kaplan and Norton [12], to define their strategic goals and to select business processes which support these goals. The BSC suggests that we view the organization from four perspectives: financial, customer, internal business processes and learning and growth, and to develop metrics, collect data and analyse it in relation to each of these perspectives. BSC adds strategic non-financial performance measures to traditional financial metrics to give managers and executives a more 'balanced' view of organizational performance. Paper [8] describes a web-based business process evaluation model based on BSC and fuzzy AHP (Analytic Hierarchy Process). The decision hierarchy is structured by the four major perspectives of BSC, followed by criteria and detailed criteria. AHP is also used to generate the weighting of the four perspectives and the weighting of the evaluation criteria. Because human decision-making process usually contains vagueness, fuzzy theory is adopted to solve this problem.

MDA is an approach to software development in which application code can be automatically generated from design models. The Computation-Independent Model (CIM) should represent the definition of requirements and specification model that incorporates a true interface by taking elements in the real world, and realising them in the software modelling domain. The OMG describes the CIM as ‘‘the environment of the system, and the requirements for the system’’ [13]. However, the OMG fails to prescribe a clear definition of the CIM constitution and supports the use of BPM notations (BPMN). But the BPMN models fail to match the OMG definition of a CIM [5].

Many works describe benefits resulting from connection of BPM and SOA [9]. Service Oriented Architecture and Business Process Management in combination have been used for agility in services and dynamic process management for many years. SOA should not be

confused with web services. It is not a product, it is a management concept. Its system architecture breaks down business software into function units executed by small software modules. Although the SOA configuration process is technology driven, BPM helps SOA architect to answer the following questions: which services are demanded, in what sequence, what response times are needed, are there any critical services, how will business process changes impact on executed BPEL processes, and so on.

Information systems have been used to support business processes for a long time. Consequently, there has been an ongoing trend towards connecting and integrating the isolated IT solutions of enterprises into coherent systems in order to support the seamless execution of business processes electronically. Due to the often business critical nature of the processes such systems need to adhere to service level agreements that guarantee quality aspects such as a specific throughput or availability for a given process. However, the key weakness of existing development approaches of information systems is that they mostly focus on functional aspects of system development and lack an integrated view between different stakeholders and views in development, e.g. between business process management and software architecture [10].

3. ROADMAP AREAS

Organizational (Corporate) growth typically goes hand in hand with an increasingly complex IT environment. Aligning IT structures with an organization's objectives and business processes is the only way to achieve sustained improvements and significant reduction in cost of developing, maintaining, and upgrading of IT systems. Bringing all IT development under BPM platform confers the following benefits to the IT organization [14]:

- flexible, rapid development and deployment in one unified platform;
- reuse of existing IT assets;
- scalable, enterprise-class platform;
- change management;
- integration-based infrastructure that allows third-party applications to be leveraged;
- quick and easy meeting of business demands;
- self-service capability for business users (through governed configuration, customization, and composition);
- fast, iterative development, thereby decreasing project risk; and
- proof of the value of IT.

Business process management has been around over 20 years. Innovations in technology such as XML, Web services, component-based development, and message-oriented middleware have fuelled the current interest in BPM. One key factor reported for obtaining a high quality BPM is the active participation of all the stakeholders that ensures the development of a shared vision of the business processes. Unfortunately, literature shows that, in practice, organizations face various difficulties in achieving this state because of: a lack of truly cross-departmental BPM initiatives [15]; a lack of consensus on the BPM notations; and the growing complexity of these notations. Despite efforts made to overcome these difficulties, it is reported that the most popular BPM notations still lack the constructs to be easily used as the means of requirements elicitation in software engineering.

Software development is dependent on the quality of the requirements elicitation activities [16] and it is crucial that the business processes are adequately modelled. Software requirements express the requirements and constraints on a software product that contributes to the satisfaction of some 'need' in the real world. Ambiguity is a major problem in requirements specification.

- Individual as well as group decisions are made increasingly difficult due to the increasing size and complexity of software projects. Stakeholders engaged in RE processes typically have to deal with the following types of decisions [17]:

- Quality decisions, e.g.: Is the requirement non-redundant, concrete and understandable?
- Preference decisions, e.g.: Which requirements should be considered for the next release?
- Classification decisions, e.g.: Which topic, component, or team does this requirement belong to?
- Property decisions, e.g.: Is the effort estimation for this requirement realistic?

This roadmap is designed to help stakeholders to deal efficiently ("doing things right") and effectively ("doing the right things") deal with these decisions. It is based on the four areas which are identified [18] as the foundation of requirements engineering:

- All terminology used in requirements engineering should be grounded in the reality of the environment for which it is to be built.
- It is not necessary or desirable to describe (however abstractly) the system to be built. Rather, the environment is described in two ways: as it would be without or in spite of the system and as we hope it will become because of the system.
- Assuming that formal descriptions focus on actions, it is essential to identify which actions are controlled by the environment, which actions are controlled by the system, and which actions of the environment are shared with the system. All types of actions are relevant to requirements engineering, and they might need to be described or constrained formally. If formal descriptions focus on states, then the same basic principles apply in a slightly different form.
- The primary role of domain knowledge in requirements engineering is in supporting refinement of requirements to implementable specifications. Correct specifications, in conjunction with appropriate domain knowledge, imply the satisfaction of the requirements.

These four areas (see Figure 1) explain the precise nature of requirements, specifications, and domain knowledge, as well as the precise nature of the relationships among them. They establish minimum standards for which information should be represented in a requirements language. They also make it possible to determine exactly what it means for requirements engineering to be successfully completed.

Many times developers completed a project confident that they have succeeded, only to realize that a key stakeholder had completely different expectations and objectives. A thorough understanding of project objectives is critical for project success, so these objectives must be identified. It is highly recommended to decompose these objectives and to connect them to particular business processes (functions). Objectives can be related to each other in a hierarchy and description can also indicate the critical factor a goal has.

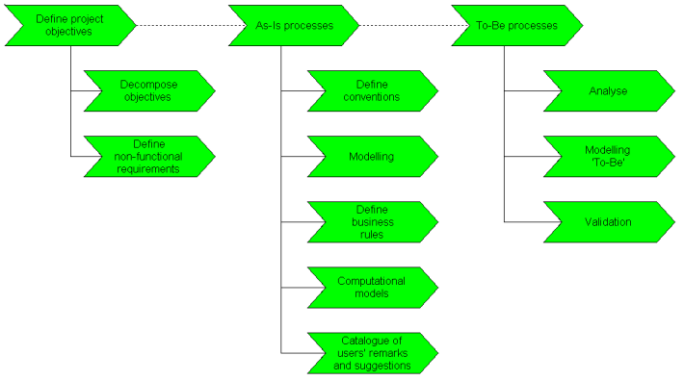

Figure 1: Roadmap.

The question is how stakeholders can measure accomplishment of these goals. Key Performance Indicators (KPIs) use a language that is easily understood: metrics. KPIs can help developers to know where to focus resources to deliver the greatest business benefit—

so they can better align IT investments with the business. Thus, KPI (one or more) must be defined for all project goals.

Generally, non-functional requirements apply to the system as a whole, so (some of) these requirements can be elicited from project goals, and it is a highly recommended practice. Non-functional requirements also require metrics.

4. AS-IS PROCESSES

Developers have long recognized that understanding of the business processes is the key to eliciting the needs of theirs users. The business process models serve as a basis for the requirement analysis. But developers use tools and methods which are incomprehensible to the business users. Business users are generally intimidated by code; they prefer a visual way to design and analyse processes, to determine inter-process dependencies, and to extract system requirements from processes. Business users should model business processes in some of business process modelling languages, which could then be transferable to those modelling techniques used by developers. Developers can add further detail and abstraction to suit the engineering need.

This roadmap recommends usage of Event-driven Process Chain (EPC) method for process modelling and UML diagrams for software development. EPC is one of the most commercial used languages for process modelling, and is supported by major ERP and BPM vendors, and UML is de facto standard for object-oriented software development.

After the project's objectives are defined, the as-is processes should be modelled. We recommend top-down approach: beginning with high-level processes and refining them to low-level processes. Models created here are independent of both the selected system architecture and the implementation platform.

4.1 Conventions

However, some conventions on project must be established first. These conventions include modelling conventions, graphics conventions, roles and privileges, naming conventions and procedures for configuration management:

- Modelling conventions include definition of levels of business process modelling, model and object types, properties and connections between them.
- Graphics conventions include visual presentation of models, object and connections, templates, font formats, print options, use of free-form graphics and texts.
- The modelling and graphics conventions define which items are mandatory and which are optional.
- Different users need to see different information, different levels, different perspectives (views), so role concept can be used to define privileges of individual users or groups.
- The naming conventions enable standardized and clear presentation of documentation. It is crucial to standardize use of performance names/activities, use of abbreviations or words that are usually abbreviated and to define company-specific glossary.
- Procedures for configuration management include procedures for updates, versions and backups of the requirements database(s).

These conventions are foundation for completeness check.

4.2 Business process modelling

Discovering how departments and employees work together in real life involves addressing the following aspects [19]:

- Organizational analysis:

Who works with whom, how, and in which order?

- Analysis of data and document relationships:
  Which data and documents are used in the process, and how?
- Analysis of system support:
  Which IT systems are used, and how?

Organizational diagram shows the structure of an organization and the relationships and relative ranks of its parts. For the purpose of RE, it is necessary to model the corporate organizational structure to the level of positions/jobs.

The workflow sequences are modelled using the EPC method: functions and events alternate possibly separated by connectors. The events in EPC are states with no duration; they are triggers for function or results of function. The individual function can be augmented with a description of the input and output data and documents, the operational organizational units (position), and the IT system which supports that function.

Data model diagrams, such as an entity relationship diagram, showing the structure of the inputs/outputs. We recommend that business users only collect necessary data and documents (templates, scans of document, screens etc.), and developers model this data in data model diagram with which they are familiar.

We highly recommend collecting of user suggestions and comments during the modelling processes and cataloguing them. In this way, it is possible to gather critical information on the actual efficiency of business processes.

4.3 Business rules

Business Process Management (BPM) and Business Rule Management (BRM) technologies are both used to improve the processes agility, flexibility and efficiency, but they do so at different levels. BPM is focused on defining, analysing and monitoring business processes while BRM is focused on defining, maintaining and executing decision logic that is used in individual process activities.

In many cases, changes in applications only affect decisions while the structure of business processes remains stable. It is highly recommended to separate description (modelling) of process logic from description of the actual decisions logic. To do this: first, decision points in process must be identified; second, all rules relevant to the decision must be assembled. Description of decision logic must be understandable to non-technical users because they must be able to validate and update these rules. It is convenient to use either Excel tables for rule definitions, or one of business rules applications which convert decision logic directly into the source code.

4.4 Computational models

UML has been widely accepted in the object-oriented software development, but a number of experts agree that UML lacked the vocabulary to express business processes in a natural and intuitive way. Business stakeholders can hardly understand requirements and process logic in technical terminology, so we recommend usage of EPC modelling language for business processes modelling. The question is how to integrate both modelling worlds and link them methodologically? There are commercial frameworks which enables a UML analysis model to be derived from the business process models. In this way, the elements that exist in business process models are re-used. But, the same principles can be used for manually transformation. Two complementary methods of deriving a UML analysis model from the business process model are used [20]:

- Use Case Model
  All functions to be implemented in a software system must be identified in the business processes. These functions are treated as use cases in the UML environment and copied to a UML use case diagram. All organizational units

assigned to the function are added to the use case diagram as actors. The use cases can be detailed by further diagrams.

- Analysis Class Model
  The input and output parameters of the EPC functions – the so-called technical terms – represent another point of contact for the requirement analysis. In the requirement analysis, these technical terms are examined to determine candidates which are suitable for technical classes and these which are more suitable as technical attributes. Technical classes and technical attributes determined in this way are positioned in the same diagram and linked with the respective technical term.

5. TO-BE PROCESSES

5.1 Analyse

Analysing project objectives, non-functional requirements, ‘as-is’ models (including BPM models, computational models, business rules) and catalogue of users suggestions and remarks, developers can define the requirements.

Requirements should be evaluated and analysed as they are being developed. 'As-Is' models can be used to analyse the requirements and clarify misunderstandings, and to provide feedback and guide the requirements elicitation process. Models are also used to understand aspects of the problem domain and test the feasibility of the requirements.

It is necessary to identify the risks due to various legal requirements or regulations on conduct.

Finally, the requirements need to be analyzed for potential ambiguities and conflicts. The results of analyse, identified ambiguities and conflicts should be resolved in consultation with the stakeholders.

5.2 To-be model

Requirements must clearly communicate the needs of the users and the constraints on the software. Requirements should be read, analysed, re-written and validated, so it is very important in which manner they are documented.

The business users simply cannot, or at least it is very difficult for them to, understand software modelling tools. To facilitate the validation of requirements, it is necessary to model to-be processes. The 'to be' model is a proposed diagram of how we hope the future process to be. This is used to demonstrate, model and test the new process and then to implement it.

Models created here are independent of both the selected system architecture and the implementation platform.

5.3 Validation

The resulting system should meet the need for which it is designed. Requirements validation is the activity of checking correctness, completeness and consistency of the requirements. Validating the requirements is not an easy task especially when stakeholders have divergent goals. It can be a problem to maintain agreement with all stakeholders in the process of requirements eliciting and modelling. Therefore, it is necessary to analyse and resolve conflicting requirements, to support stakeholder negotiation, and to reason with models that contain inconsistencies.

6. CASE STUDY

In this section, we describe a practical application of this roadmap through a case study of an aircraft turn around automation.

Business specialists from Split Airport defined project objectives and non-functional requirements, modelled as-is processes, collected suggestions and validated to-be processes. Developers added further details in computational models, analysed requirements and (together with business specialists) modelled to-be processes.

The aircraft turn around is process of loading, unloading, and servicing of aircraft, and it is one of the most critical and complex parts of Split Airport's key process. This part of the process comprises the sequence of ground operations required to service an aircraft from landing to take-off. Some activities are carried out par parallelly, while the others are carried out in the specific order. Activities are defined by industry standards, but the duration of each activity is defined by agreement with airline.

The turn around process includes simultaneous activities of many organizational groups (teams) and third parties, and it is essential that they work together effectively to avoid delays. Turnaround coordinator monitors the flow of the process at the apron and manages teams work. Coordinator needs real-time information (from check-ins, gates, baggage area, about slots, etc.). They enable him to maximise the efficiency and effectiveness of the turn around process.

Before automation, turnaround coordinator used radio station for communication and filled the necessary papers manually. Paper based process is prone to error and difficult to analyse and search. Turnaround coordinator needs some handheld device to enter data during the turn around process and to send and receive necessary information to integrated airport software system.

Split Airport wished to proactively manage aircraft turn around and decided to enhance existing integrated airport software system with another module (see Figure 2).

Figure 2: Turn Around Monitoring (PDA Application).

6.1 Objectives and non-functional requirements

In the first phase objectives are set:

- early identification of problems;
- to make all necessary information available to coordinator;
- to capture accurate data for invoicing;
- to replace paper based process;
- to enable analysis and SLA/KPI measurement.

These objectives are connected to particular business functions. For each objective, one or more KPI are defined (see Figure 3), as well as critical factors.

As Split Airport uses ARIS tools group for business process managing for more than ten years now, we used ARIS Toolset for modelling of objectives (objective diagram), organizational and process structure. ARIS BSC is used to define KPIs.

The next non-functional requirements are defined:

- application should be robust;

- apron must be 100% signal covered, with high redundancy, because aircrafts are dynamic obstacles, metal reflective;
- handheld device must be lightweight, waterproof, with good sunlight visibility.

6.2 As-is processes

Turn around process is already modelled in ARIS Toolset, with appropriate level of details, so we used existing models and conventions. We checked and found that the models generally correspond to the actual state of processes, and needed only some minor changes.

During the verification of the model we have created a catalogue of users' suggestions and comments (see Figure 4).

We used ARIS UML Designer to generate computational models. The existing elements in the business process models are re-used. Functions are treated as use cases in the UML environment and copied to a UML use case diagram. All organizational units assigned to the function are added to the use case diagram as actors. The input and output parameters of the functions in the EPC models are automatically transferred in (initial) system analysis models, and those models were manually refined by developers.

6.3 To-be processes

When developers were sure to understand problem domain and when several ambiguities and conflicts were resolved, 'to-be' processes were modelled (see Figure 5). 'To-be' models described the environment, and expressed what the new system must achieve in that environment. Thus, business users could validate that given design would meet the objectives.

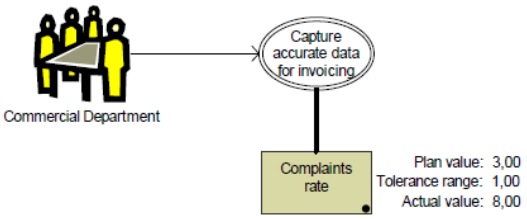

Figure 3: KPI Allocation Diagram for Objective.

7. CONCLUSION

The key cause of software project failures is the lack of clarity in eliciting and communicating user requirements. This is the result of misunderstanding and lack of communication between developers and business stakeholders. Generally, developers expect business stakeholders to understand requirements and process logic in technical terminology, which is usually not true. Developers and business stakeholders use different terminologies, different tools and different methodologies and have different perspectives.

This paper explores how to integrate both worlds to ensure the completeness, correctness and consistency of requirements. Integration of both worlds leads to adaptation, optimization and control of physical processes in software systems from a business perspective.

BPM models should be technology independent and without restrictions imposed by certain platforms. These models describe the environment rather than the system to be built. The environment is described in two ways: as it would be without the system and as we hope it will become because of the system.

Process flow is modelled at all levels of business, logic and implementation perspectives. Data models are modelled at the logical level and implementation level, although technical

information is identified and extracted from the business process model, and expanded with additional details in the computational models. Unified description language ensures consistency and reusability of data and models.

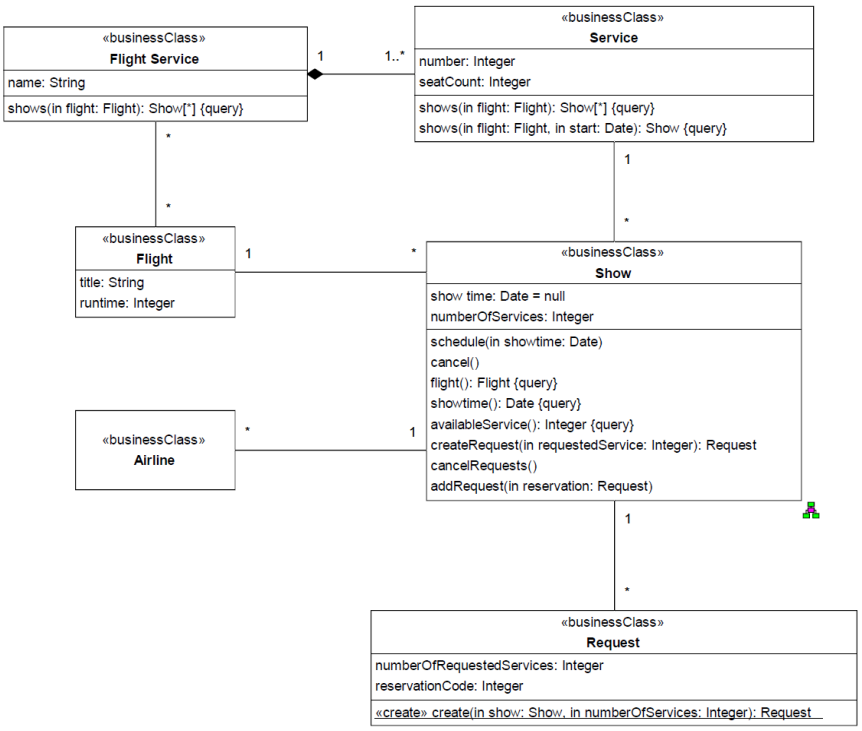

Figure 4: Service Request UML Class Diagram.

This paper highlights the importance of establishing of project goals and non-functional requirements, of collecting user's comments and suggestions and of assembling business rules. These activities are correlated with the success of project.

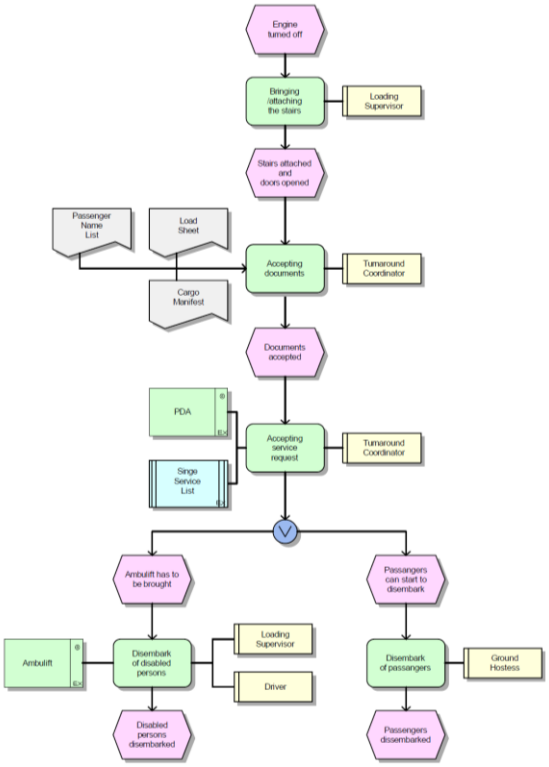

Figure 5: A Partial View of Aircraft Turn Around To-Be Process (EPC Model).

Roadmap presented in this paper is used at Split Airport for requirements definition in software projects. We describe a practical application of the roadmap through a case study in the automation of aircraft turn around process.

REFERENCES

[1] The Standish Group. The Chaos Report (1994). from http://www.standishgroup.com/sample_research/chaos_1994_1.php, accessed on 16-08-2011

[2] Leffingwell, D. (1997). Calculating the return on investment from more effective requirements management, American Programmer, Vol. 10, No. 4, 13-16

[3] Pohl, K., Process-Centered Requirements Engineering, John Wiley and Sons, 1996

[4] Leyman, F.; Roller, D., and Schmidt, M.-T. (2002). Web services and business process management. IBM Systems Journal, Vol 41, No. 2, 198

[5] Fouad, A.; Phalp, K.; Kanyaru, J.M.; Jeary, S. (2011). Embedding requirements within Model-Driven Architecture, Software Quality Journal, Volume 19, No. 2, 411–430, doi: 10.1007/s11219-010-9122-7

[6] Nuseibeh, B.; Easterbrook, S. (2010). Requirements engineering: a roadmap, ICSE '00: Proceedings of the Conference on The Future of Software Engineering, 35-46

[7] Cheng, B.H.C.; Atlee J.M. (2007). Research Directions in Requirements Engineering, FOSE '07: 2007 Future of Software Engineering, 285-303, doi: 10.1109/FOSE.2007.17

[8] Cho, C.; Lee, S. (2011). A study on process evaluation and selection model for business process management, Expert Systems with Applications, Vol. 38, 6339–6350

[9] Klueckmann, J. (2007). 10 Steps to Business-Driven SOA, ARIS Expert Paper Book, 30-36

[10] Hummel, O.; Momm, C.; Hickl., S. (2010). Towards Quality-Aware Development and Evolution of Enterprise Information Systems, SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing, 137-144

[11] Pavlovski,C.J.; Zou, .J. (2008). Non-functional requirements in business process modelling, APCCM '08: Proceedings of the fifth Asia-Pacific conference on Conceptual Modelling - Volume 79, 103-112

[12] Kaplan, R. S.; Norton, D. P. (1992) The Balanced Scorecard – Measures That Drive Performance, Harvard Business Review

[13] OMG. (2003).MDA guide version 1.0.1. Object Management Group, from http://www.omg.org/docs/omg/03-06-01.pdf, accessed on 16-08-2011

[14] Software AG (2007). Getting Started with Business Process Management, Business White Paper, from http://www.softwareag.com/corporate/images/sec_BPM_Get_Started_WP_tcm16-48780.pdf, accessed on 14-08-2011

[15] Monsalve, C.; April, A.; Abran, A. (2011). Requirements Elicitation Using BPM Notations: Focusing on the Strategic Level Representation, ACACOS'11: Proceedings of the 10th WSEAS international conference on Applied computer and applied computational science, 235-241

[16] Abran, A.; Moore, J.; Bourque, P.; Dupuis, R., SWEBOK: Guide to the Software Engineering Body of Knowledge 2004 Version. IEEE Computer Society, Los Alamitos, California

[17] Felfernig, A.; Schubert, M.; Mandl, M.; Ricci, F.; Maalej, W. (2010). Recommendation and decision technologies for requirements engineering, RSSE '10: Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering, 11-15

[18] Zave, P.; Jackson, M. (1997). Four dark corners of requirements engineering, Transactions on Software Engineering and Methodology (TOSEM), Volume 6, Issue 1, 340-343

[19] Blickle, T.; Hess, H. (2010). Automatic Process Discovery with ARIS Process Performance Manager (ARIS PPM), Business White Paper, from http://www.softwareag.com/corporate/images/sec_SAG-IDS_Automatic_PD_ARIS_PPM_WP_Oct10-web_tcm16-78674.pdf

[20] Andres, T. (2010). White Paper: From Business Process to Application (Model-driven Development of Business Management Software), Business White Paper, from http://www.softwareag.com/corporate/images/sec_SAG-IDS_BusProc-Appl_WP_Oct10-web_tcm16-80004.pdf, accessed on 14-08-2011