

Conceptual Model of User Adaptive Enterprise Application

Inese Šūpulniece and Jānis Grabis

Institute of Information Technology, Riga Technical University, Latvia

Inese.Supulniece@rtu.lv, Grabis@rtu.lv

Abstract. The user adaptive enterprise application is a software system, which adapts its behavior to an individual user on the basis of nontrivial inferences from information about the user. The objective of this paper is to elaborate a conceptual model of the user adaptive enterprise applications. In order to conceptualize the user adaptive enterprise applications, their main characteristics are analyzed, the meta-model defining the key concepts relevant to these applications is developed, and the user adaptive enterprise application and its components are defined in terms of the meta-model. Modeling of the user adaptive enterprise application incorporates aspects of enterprise modeling, application modeling, and design of adaptive characteristics of the application. The end-user and her expectations are identified as two concepts of major importance not sufficiently explored in the existing research. Understanding these roles improves the adaptation result in the user adaptive applications.

Keywords: Enterprise application, modeling, adaptation.

1 Introduction

Various definitions of adaptation imply the notion of a changing object to meet some specific requirements or purpose [1]. The user adaptive application is a software system, which adapts its behavior to the individual user on the basis of nontrivial inferences from information about the user [2]. Research on the user adaptive applications is fragmented focusing on individual aspects of these applications, and in practice users often are not satisfied with the results of adaptation. Partially this is caused by the lack of generic theoretical analysis about the essence of adaptation and causes of adaptation success or failure.

This paper focuses on User Adaptive Enterprise Applications (UAEA), because adaptivity can be one of the solutions to addressing usability problems of large packaged applications. The main distinctive feature of adaptation of enterprise applications is a focus on improving business process execution efficiency according to business goals and needs of users. Regardless of the type, adaptive applications have a number of common features distinguishing them from non-adaptive applications. The common features of the adaptive systems are monitoring of changes, goal driven adjustment of the system, and a feedback loop measuring the success of the adjustment [3]. Self-adaptive systems recently have attracted the most attention in computer science. The self-adaptive systems are able to modify their behavior and/or structure in response to their perception of the environment and the system itself, and their goals [4]. Cheng et al. [5] have identified main research challenges associated with self-adaptive systems, namely, modeling, requirements specification, engineering with emphasis on representation of the feedback loops, and runtime assurances. These research challenges are further complemented

and extended in [4] by focusing on design space definition, development processes, decentralization of control loops, and practical run-time verification and validation.

Modeling of adaptive applications is researched in several domains, from various perspectives, and for different purposes. The cornerstone of self-adaptive systems is their conceptual architecture [6]. It has many similarities with the base structure of autonomic computing. It explicitly distinguishes controllable and uncontrollable (i.e., environment) parts, and shows that different type of adaptation logic can be used. The adaptation logic implements a control loop in line with the monitor-analyze-plan-execute (MAPE) loop of the autonomic computing. Salehie and Tahvildari [7] define a hierarchy of so called self-properties. They classify self-adaption according to the object of adaption, realization issues, temporal characteristics, and interaction concerns. Barth and Gomi [8] propose a meta-level architecture to separate the components responsible for the acquisition and manipulation of user modeling from other components of the adaptive system. Bielikova and Moravcik [9] present the method for modeling content of adaptive applications using ontology. Juan and Sterling [10] introduce the ROADMAP meta-model, designed to describe intelligent adaptive multi-agent systems in open environments. Weyns et al. [11] elaborate a formal reference model of self-adaptive systems. The model is represented using both UML diagrams and Z language that allows reasoning about behavior of adaptive systems. Multiple case studies are provided.

Context-sensitive applications are required to adapt dynamically to context of use situations unforeseeable at design time. This leads to the recent extension of modeling scope from design time to runtime. A group of researchers utilize models at runtime, e.g., Lehmann et al. [12] present meta-model of runtime models. Modeling dimensions, which are proposed by Andersson [6], are applicable for describing the adaptation process, however, relationships between business and adaptation goals are missing. Modeling of the business goals is an important part of enterprise modeling [16] and these models must be considered during the enterprise application development and implementation. The existing conceptualizations also mainly focus on technical systems with limited attention devoted to actual users of information systems.

A model of UA EA is a starting point of further design and implementation of these applications. However, majority of investigations developing adaptivity methods or algorithms focus on representation of these particular methods rather than on capturing common features of user adaptive applications. Therefore, it is proposed to develop a general meta-model for modeling UA EA, which provides a common general representation of these applications, and specific adaptive method can be detailed on the top of this model. The common high level representation is useful because adaptive mechanisms can change quickly, e.g., new information about users, environment and systems becomes available.

The objective of this paper is to describe a conceptual model of UA EA. In order to conceptualize the UA EA, its nature is analyzed and the meta-model defining the key concepts relevant to these applications is developed.

Traditional system modeling approaches have limitations to highlight the cause of adaptation, adaptation mechanisms, consequences, and user adaptive application specific factors. Therefore this paper presents the modeling approach that is demonstrated using the developed generic UA EA model.

The rest of the paper is structured as follows. Section 2 introduces the main concepts for modeling the user adaptive enterprise application. The meta-model is described in Section 3. A few UA EA models are presented in Section 4. The paper concludes with Section 5, where research results and further research are discussed.

2 User Perspective of Adaptive Application

In the adaptation process, there are changing and adapted objects [1]. A simplified input/output view of the core concepts of the adaptive application is presented in Figure 1.

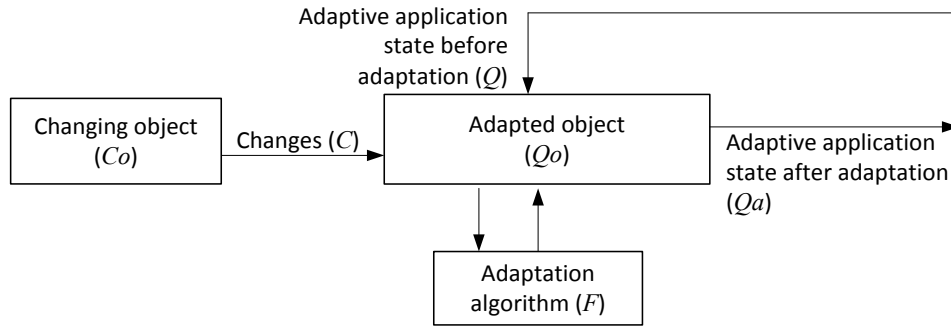


Figure 1. A simplified input/output view for the core concepts of the adaptive application

The adaptive application state after the adaptation (Q_a) depends on the state before the adaptation (Q), changes (C), and the adaptation algorithm (F) or

$$Q_a = F(C, Q) \quad (1)$$

Consequently the main core concepts of the adaptive applications are: changing object (C_o), adapted object (Q_o), and adaptation algorithm (F).

The adaptation is a purposeful process, so a stakeholder (H) defines the goals, which are realized by the adaptation process [13] (Figure 2). Another important actor is an end-user (U), who has expectations in her mind, what should be the state of the application after the adaptation process. The stakeholder and the end-user view the adaptation result on a different generalization level (as goals and expectations). The stakeholder and the end-user are roles, and one person might take both roles at the same time.

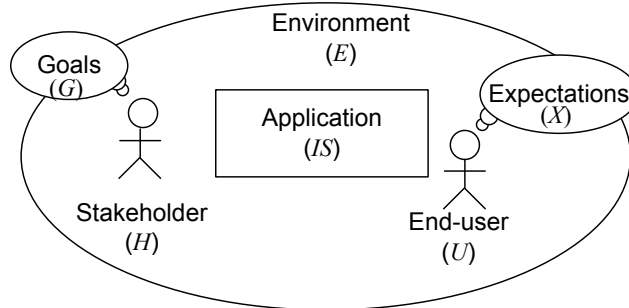


Figure 2. Stakeholder and End-user concepts in an adaptive application

The stakeholder benefits from the adaptive application, so she formulates the set of goals for the adapted object. The set of goals for the adaptive application is a union of all stakeholders' goals or

$$G = G_{H_1} \cup G_{H_2} \cup \dots \cup G_{H_k}, \quad (2)$$

where $G_{H_i} = \{G_{H_{i1}}, G_{H_{i2}}, \dots, G_{H_{iv}}\}$ is a set of goals defined by the i -th stakeholder and G is a set of all goals defined by the all stakeholders. The goals do not change in time.

However, the set of expectations towards the adaptation result is defined by the end-user. All end-users' expectations form the set of expectations for the adaptive application or

$$X = X_{u_1} \cup X_{u_2} \cup \dots \cup X_{u_{uk}}, \quad (3)$$

where $X_{u_w} = \{X_{y_1}, X_{y_2}, \dots, X_{y_{vx}}\}$ is a set of expectations towards the parameters characterizing the application state after the adaptation. The expectations differ per each individual end-user. These expectations also are time dependent (e.g., depend on the user's mood or the particular situation) even if a subject is the same.

The adapted object state after the adaptation is defined as:

$$Q_a = F(C, Q, G, W), \quad (4)$$

where W is a set of the calculated expectations (Figure 3).

The identified core concepts of the adaptation are used during the modeling and development of the UAEA. They can be used to illustrate the differences between similar adaptive applications, too.

The expected result (the adaptive application state after the adaptation) should be equal to the expectations set by the end-user or

$$Q_a = X_{u_w}, \quad (5)$$

where $Q_a = \{Q_{a_1}, Q_{a_2}, \dots, Q_{a_{vx}}\}$ is a set of the real adaptive application parameters after the adaptation. Practically, that is hardly achievable.

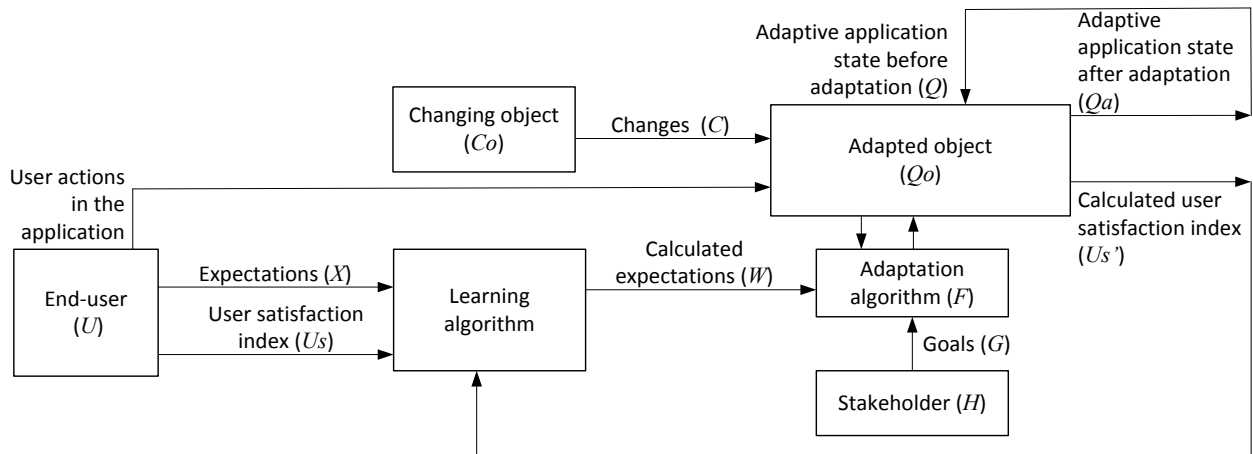


Figure 3. An extended input/output view for the core concepts of the adaptive application

The user satisfaction index shows the proportion of the user expectations fulfilled as the result of the adaptation:

$$US = \frac{\sum_{vix=1}^{vx} IX_{vix}}{vx}, \quad (6)$$

$$\text{where } IX_{vix} = \begin{cases} 1, & \text{if } X_{y_{vix}} = Q_{a_{vix}} \\ 0, & \text{if } X_{y_{vix}} \neq Q_{a_{vix}} \end{cases}$$

If $US = 1$ then the user expectations are equal to the application state after the adaptation, where the user is completely satisfied with the adaptation result (this is an ideal adaptation). If $US < 1$, then all user expectations are not met and the user is not satisfied. Thus, the main goal of the adaptation would be to minimize the difference between the adaptation result and the user expectations:

$$|QX| \rightarrow \max, \quad (7)$$

where $QX = Q_a \cap X_{u_w}$.

Even if it is unrealistic to fully capture all individual expectations, the application might know or predict some user expectations towards the adaptation, e.g., user preferences available in the application. These are referred to as calculated user expectations (QW). Consequently, the calculated user satisfaction index (Us') shows the proportion of the calculated user expectations fulfilled as the result of the adaptation.

Substituting X_{u_w} with W_{u_w} yields that the goal of the adaptive application is to minimize the difference between the application state after the adaptation and the calculated user expectations or

$$|QW| \rightarrow \max, \quad (8)$$

where $QW = Q_a \cap W_{u_w}$.

3 Meta-model

Systems analysis and systems modeling are widely used to analyze, conceptualize, and construct the application before actual development and implementation. Domain specific extensions of standard modeling languages are often used to represent specific modeling aspects. Different adaptive techniques are increasingly adopted in enterprise applications, and a general approach for modeling these adaptive mechanisms would be useful to capture specific aspects of modeling and developing adaptive applications.

A meta-model for modeling the UAEA provides a common general representation of this kind of applications, and the specific adaptive method can be developed on the basis of this model.

3.1 Overview

While analyzing conceptual aspects of the UAEA in Section 2, the key concepts were identified – stakeholder and end-user, goals and expectations, changing object, adapted object and adaptation algorithm. Thus modeling approach should highlight and emphasize them. The UAEA meta-model consists of a number of sub-models corresponding to these concepts (Figure 4):

- 1) Stakeholder and End-user Model (SEM) presents the structure of actors (human roles), which are related to the adaptive application.
- 2) Goals and Expectations Model (GEM) illustrates the structure of goals towards the adaptation and individual user's expectations behind them.
- 3) Changing Object Model (COM) is the structure of the application or the environment part, which is changing (triggers the adaptation).
- 4) Adapted Object Model (AOM) is the structure of the application, which is adapting to the change.
- 5) Adaptation Algorithm Model (AAM) describes the rules and the behavior of the particular adaptation algorithm.
- 6) System Model (SM) presents the structure of the application (e.g., the architecture).

SEM, GEM, COM, AOM and SM are structural diagrams, which present the main elements of the adaptive application and relationships between them. AAM is a behavioral diagram.

3.2 Goals and Expectations Sub-model

Figure 5 shows goals and expectations in the meta-model. There are three types of goals: business goals, operational goals, and technical goals. Here business goals and operational goals

are inspired by [14]. The technical goals are associated with expectations and adapted object. All types of goals are associated with stakeholders and expectations are associated with end users.

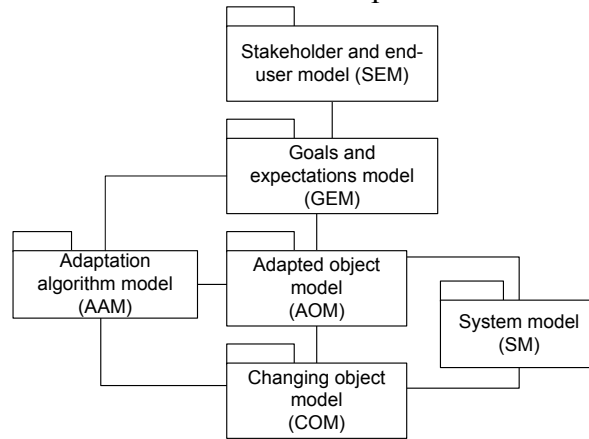


Figure 4. The model of the user adaptive enterprise application – a high level abstraction

Business goals (similar to Officials goals defined in [14]) are formally stated goals of an organization and often they are part of organization’s scorecard or annual report. These are high level goals and cannot be reached only with application. Consequently, the success/failure of these goals cannot be measured only with technical calculations.

Operational goals (similar to Operative goals defined in [14]) are the outcomes that the organization actually seeks to attain through its operating policies and activities; they define performance objectives. These goals are often measurable and part of the measures can be derived from applications.

Technical goals may be evaluated using technical measurement. All types of goals from Goals model can have hierarchy and relationships presented in the model. For modeling adaptive application only technical goals are explored further, but linkage to business and operational goals is the advantage as it clearly shows business benefits of adaptive application.

There are available various modeling languages and techniques for goals modeling, e.g., i* modeling language [15], goals model from 4EM method [16], and goals diagram from KAOS methodology [17].

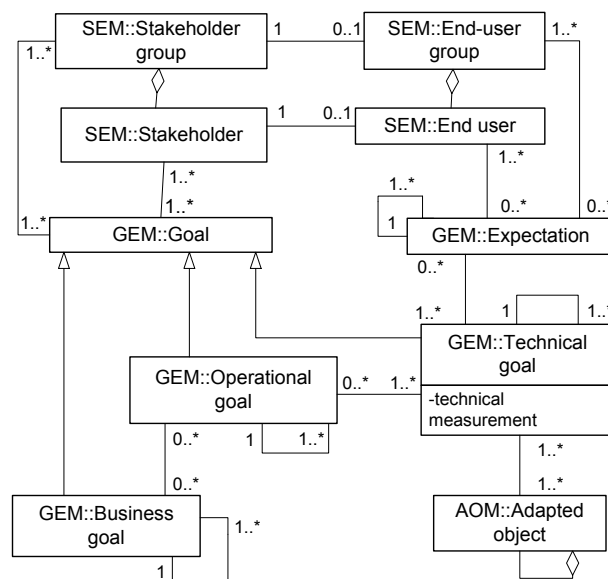


Figure 5. Goals and expectations in the meta-model

3.3 Other Sub-models

System sub-model describes the application, where adaptation is performed, e.g., system architecture or conceptual model of the application with highlighted changing and adapted parts. Adapted Object Model (see meta-model in Figure 6) is a part of system’s model, which specifies adaptive components and explores in details adaptive part of the application.

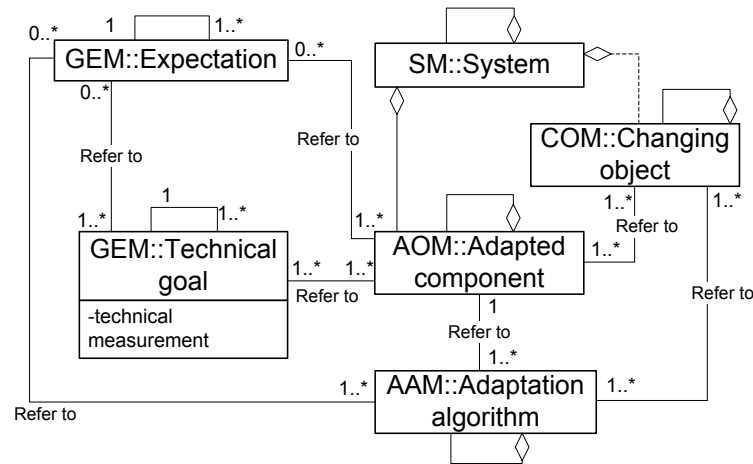


Figure 6. Adapted component’s concept in the meta-model

Adapted components are related to technical goals to illustrate the benefits, thus allowing selecting the best sub-set of adaptive components. Adaptive components are also related to expectations to illustrate concepts, which are used in adaptive components to achieve better adaptation result. Adaptation algorithm is linked to Adapted component as it explores adaptation logics for Adapted component. The relationship between Adapted component and Changing object presents triggers for starting the processes in Adapted component.

Changing Object Model defines the structure and interaction of those concepts, which cause the change or trigger the adaptation process. The change can happen within the application or outside of it. Thus Changing Object Model can be part of System model, which specify changing components and explore in details the changing part of the application. Or Changing Object might have just input/output link to System Model. Changing object is related to Adapted object and adaptation algorithm to illustrate, which Adapted component needs to be executed in particular moment of the concrete change.

Adaptation Algorithm Model describes behavior of each adaptive component. User expectations impact the result of adaptation algorithm and changing object triggers execution of the adaptation algorithm or particular activities within this algorithm.

The main elements defined in the meta-model have their special graphical representations, for creating visual models of user adaptive enterprise applications (Table 1).

4 Generic Model

The developed meta-model is applied for describing a generic UAEA. The generic user adaptive application aims at incorporating wider range of different adaptive features, and specific UAEA also can be modeled using the meta-model and inspirations drawn from the generic model.

The enterprise applications are used to execute business processes. Usually these are packaged applications providing standardized implementations of business processes. These standardized implementations can be customized and modified to some extent as the result of implementation activities. Users of enterprise applications either use predefined workflows or rely on the user documentation and best practices to execute their business processes [18]. Besides these standard

capabilities, in many cases users also can use other functions provided by enterprise applications subjected to their access rights. That means that users have possibilities to introduce their own variations in process execution. Given that enterprise applications are mainly used for repetitive tasks [19], by considering these variations, users may come up with more efficient ways of executing business processes [20]. If an enterprise application supports users in identification of more efficient variations of business process execution and enables continuous execution refinement it is referred as to UAEA.

Table 1. Sub-models and their basic elements

Sub-model	Basic elements
Stakeholder and End-User model	
Goals and Expectations model	
Model of Adapted Object	
Model of Changing Object	

The overall goal of the UAEA is to improve efficiency of the existing enterprise application. This is a business goal. However, technically this can be completed by improving the application usability – decreasing time for routine activities, avoiding mistakes, and helping users in ad-hoc situations.

The UAEA is developed to solve usability issues in the currently available commercial enterprise applications while taking into account the spatial visualization ability of their users.

Thus the UAEA complements the existing (non-adaptive) enterprise application but does not change it. The UAEA consists of six independent components:

- 1) Adaptive process execution overview shows a full process or the part of the process, the current activity, and possible paths to finish the process.
- 2) Adaptive navigation support presents a quick link of next recommended activity, mandatory activities, prohibited activities, and already executed activities.
- 3) Adaptive information support recommends related documents, applications, or data based on a personal or a global user experience.
- 4) Adaptive decision support recommends possible decisions based on a personal or a global user experience.
- 5) Adaptive problem preventing presents most common problems and solutions related to the current activity. It prevents possible mistakes for non-routine activities or new users.
- 6) Adaptive error and exception handling notifies user about an incompleteness in process execution, e.g., a missed activity or unfinished process.

The idea of UAEA lies in the following observation [21]: users use the enterprise application to accomplish their tasks usually consisting of multiple steps; each user or user group has a preferred sequence of the steps (business process execution patterns). UAEA attempts to exploit such usage patterns with the aim to improve performance.

The model elaborated in the next sub-section is the basis for actual implementation of UAEA.

4.1 Model

The overall goal of UAEA is to identify possibilities of improving performance of existing enterprise applications. The main stakeholder is an abstract object named as Management of organization (Figure 7), however, management is not an end-user if we assume that management representatives are not real users of this application. New employee is the stakeholder (because she benefits from the application) and also end-user, because we assume that the new employee uses the application to execute a business process.

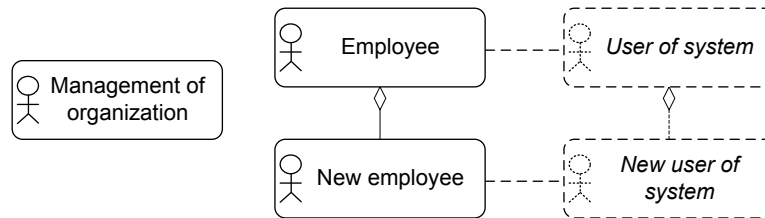


Figure 7. UAEA stakeholder and end-user model

In the proposed model we include only those business and operational goals, which can be related to technical goals. Technical goals are set towards the application, thus they can be evaluated purely in the application, where all measurements are available. Technical measurements and hierarchy of Goals, Stakeholders and End-users are hidden in Figure 8 to keep the view of the model readable. But it is possible to see these relationships, if another view perspective is selected.

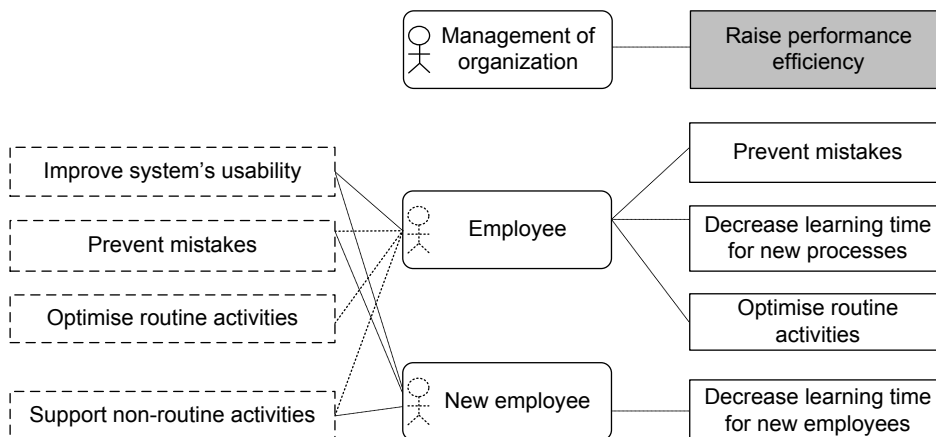


Figure 8. Relationships between UAEA stakeholder/end-user and goals

Adapted Object Model for the user adaptive enterprise application is presented in Figure 9. There are six independent components each realizing a specific adaptation algorithm.

The adapted objects are linked to the technical goals (Figure 10) to illustrate the benefits, thus allowing to select the best sub-set of adaptive components for particular user, user group, process, or application module. Implementation of all adaptive components at the same time would confuse the user and decrease application's performance because of computational overload.

Each adapted object in UAEA has one or more adaptive algorithms performing the adaptation. Various algorithms can be used for these purposes. Therefore, Table 2 identifies only general inputs and outputs as well as adaptation foals common for the adapted object. The adaptation algorithm for the adaptation navigation support and general overview of other algorithms is provided in Section 4.2.

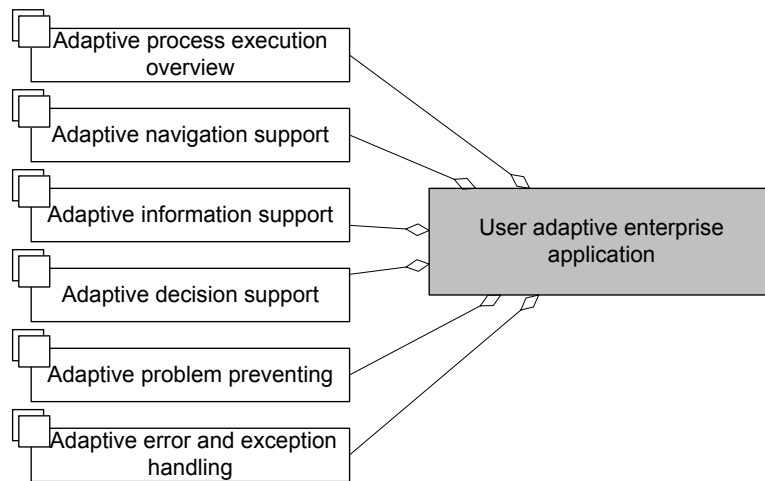


Figure 9. Model of UAEA adapted objects

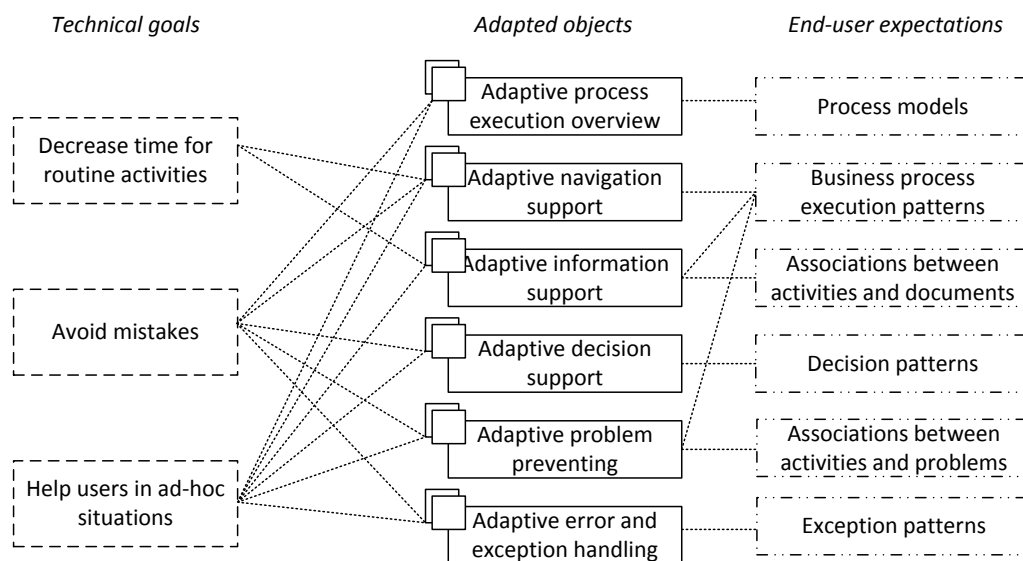


Figure 10. Relationships between UAEA technical goals, adapted objects and end-user expectations

Table 2. Adapted objects inputs/outputs

Adapted Object	Adaptation Goal	Inputs	Outputs
Adaptive navigation support	Minimize difference between executed activities and business process execution patterns.	Executed activity, business process patterns and business process execution restrictions.	Next step recommendation, mandatory and prohibited activities.
Adaptive process execution overview	Minimize difference between executed activities and process models.	Executed activity and process models.	Business process visualization.
Adaptive information support	Minimize difference between viewed documents and document-activity relationships.	Executed activity, viewed documents, document-activity relationships and business process execution patterns.	Related document recommendation.
Adaptive decision support	Minimize the difference between executed activities and decision patterns.	Executed activity and decision patterns.	Following activity recommendation (after the decision).
Adaptive problem preventing	Minimize the difference between viewed problem solutions and problem-activity relationships.	Executed activity; problem-activity relationships and business process execution patterns.	Possible problem solution recommendation.
Adaptive error and exception handling	Minimize the difference between executed activities and exception patterns.	Executed activity and exception patterns.	Error notification.

4.2 Adaptation Algorithms

In Adaptive Navigation Support (ANS), a user can find a shortcut to the next activity, mandatory and prohibited activities (Figure 11). It is a recommendation block in addition to the standard enterprise application. In the case of ANS, the calculated expectations are personalized business process execution patterns, i.e., it is assumed that the users would like to follow their personalized business process execution patterns. Consequently, the ANS adaptation goal is to maximize intersection between the sequence of activities executed so far and the personalized pattern or

$$|IP| \rightarrow \max, \quad (9)$$

where $IP = I(u_{ui}) \cap P_{pk}(u_{ui})$ and $I(u_{ui})$ is a sequence of executed activities; $P_{pk}(u_{ui})$ is the personalized pattern.

The ANS component utilizes: 1) business process execution restrictions to control business process execution rules; and 2) business process execution patterns to manage user oriented business process execution variations.

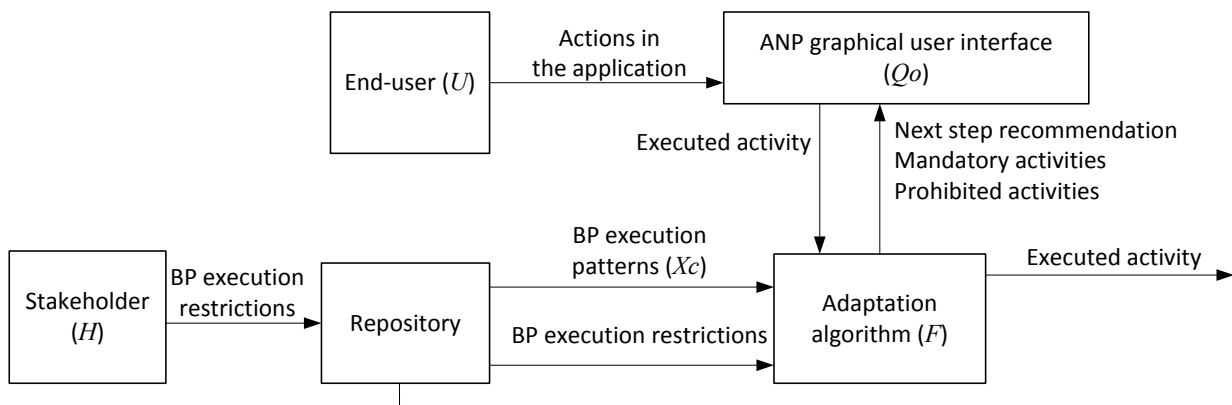


Figure 11. The adaptation algorithm's inputs and outputs

In Adaptive process execution overview object, a user can see currently executed business process visualization or a part of it. Prior activities and future activities are displayed to successfully complete the process.

In Adaptive information support object, a user can access documents needed for the particular activity. User works with several documents while executing a business process. These documents might be a part of the enterprise application or exist apart of it. Only enterprise application's internal documents are in the scope of this component. Document-activity relationships can be defined centrally or individually. The sequence of executed activities is compared with business process execution patterns and the result is the input for user group identification. If there is a relationship between the document and the particular activity, which is created or used by identified user group, then the adapted component presents a shortcut to this document.

In Adaptive decision support component, a user can see recommendations for the following process flow if decision point is reached. Decision patterns are derived from business process execution patterns.

In Adaptive problem preventing component, a user can see a note of possible problem during the execution of the particular activity and get additional information regarding the problem.

There are two note types: (1) problems created by the particular user and (2) problems created by other users. The first type notes are helpful to save personal instructions for future non-standard activities. Additional information about the problem includes a problem name, a description and a solution. A sequence of executed activities is compared with business process execution patterns to identify the user group. The second type notes include problems created by the members of identified user group. The user can also view the problem and mark it as important, - then it is added to personal notes.

In Adaptive error and exception handling component, a user is warned about made errors. The user can view the possible solution. Exception patterns are searched within user's executed activities. Exceptions and errors within this component are derived from the incorrect business process execution flow.

5 Conclusions and Future Work

A user adaptive enterprise application needs to be designed before development of its prototype. There is a wide spectrum of modeling languages, techniques and approaches for system modeling, however adjustments are needed for modeling specific aspects of adaptive applications, because adaptive applications are perceived differently than non-adaptive applications [11]. For designing an application, traditional modeling languages or techniques can be applied to model basic functionalities. Proposed meta-model should be applied additionally to traditional models as complementary modeling dimensions.

The UAEA meta-model provides the basis for the UAEA modeling method and its further elaboration would require definition of appropriate model checking and analysis mechanism.

User adaptive enterprise application modeling incorporates results of enterprise modeling (as goals and stakeholders); application modeling (as application components) and adaptive characteristics of the application (as changing and adapted object).

The end-user and her expectations are identified as two concepts of major importance not sufficiently explored in the existing research. Understanding these roles improves the adaptation result in the user adaptive application.

Sometimes adaptive part of the application has been created integrated with other application elements (e.g., user modeling components as cited in [8]) without a specific component responsible for it. The proposed meta-model can be applied to describe a detached adaptive component and also to identify adaptive characteristics for bounded adaptive functionalities.

The meta-model is applied to model main components of UAEA. The developed models are used to build a prototype of the application [22].

References

- [1] V. M. García-Barrios, "Personalization in adaptive e-learning systems - a service-oriented solution approach for multi-purpose user modelling systems," Ph.D. dissertation, Institute of Information Systems and Computer Media, Faculty of Computer Science, Graz University of Technology, 2007.
- [2] A. Jameson. "Systems that adapt to their users: An integrative overview," Tutorial presented at 9th International Conference on User Modeling, Johnstown, USA, 2003.
- [3] F. Heylighen. (2004). Web Dictionary of Cybernetics and Systems [Online]. Available: <http://pespmc1.vub.ac.be> (accessed 27.12.2013)
- [4] R. De Lemos, H. Giese, H. A. Müller, M. Shaw, J. Andersson, et al. "Software engineering for self-adaptive systems: A second research roadmap," in Software Engineering for Self-Adaptive Systems II, Lecture Notes in Computer Science, 7475, 2013, pp.1-32. http://dx.doi.org/10.1007/978-3-642-35813-5_1

- [5] B. H. C. Cheng, P. Sawyer, N. Bencomo, and J. Whittle. "A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty," in Proc. 12th International Conference on Model Driven Engineering Languages and Systems, 2009, pp. 468 - 483. http://dx.doi.org/10.1007/978-3-642-04425-0_36
- [6] J. Andersson, R. De Lemos, S. Malek, and D. Weyns, "Modeling dimensions of self-adaptive software systems," in Software Engineering for Self-Adaptive Systems, B. H. C. Cheng, et al., Eds., LNCS, vol. 5525, pp. 27-47, 2009. http://dx.doi.org/10.1007/978-3-642-02161-9_2
- [7] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," in ACM Trans. Auton. Adapt. Syst. 4, 2, 2009. <http://dx.doi.org/10.1145/1516533.1516538>
- [8] F. J. Barth and E. S. Gomi. "A meta-level architecture for adaptive Applications," in Proc. International Conference on Adaptive and Natural Computing Algorithms, 2005, pp. 329-332. http://dx.doi.org/10.1007/3-211-27389-1_79
- [9] M. Bielikova and M. Moravcik, "Modeling the reusable content of adaptive web-based applications using an ontology," in Advances in Semantic Media Adaptation and Personalization, 2008, pp. 307-327. http://dx.doi.org/10.1007/978-3-540-76361_15
- [10] T. Juan and L. Sterling. "A Meta-model for Intelligent Adaptive Multi-Agent Systems in Open Environments," in Proc. Second International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 03, 2003, pp.1024-1025. <http://dx.doi.org/10.1145/860575.860777>
- [11] D. Weyns, S. Malek, and J. Andersson, "FORMS: Unifying reference model for formal specification of distributed self-adaptive systems," in ACM Transactions on Autonomous and Adaptive Systems, TAAS, vol. 7(1), 2012. <http://dx.doi.org/10.1145/2168260.2168268>
- [12] G. Lehmann, M. Blumendorf, F. Trollmann, and S. Albayrak. "Meta-modeling runtime models," in Proc. MoDELS Workshops, 2010, pp. 209-223. http://dx.doi.org/10.1007/978-3-642-21210-9_21
- [13] L. A. Rastrigin, "Adaptation of complex systems. Methods and applications," (Adaptatsiya slozhnykh sistem. Metody i prilozheniya), (Russian), Riga: "Zinatne", 1981.
- [14] C. Perrow, "Organizational prestige, some functions and dysfunctions," in American Journal of Sociology, vol. 66, pp. 335-341, 1961. <http://dx.doi.org/10.1086/222899>
- [15] G. Grau, J. Horkoff, and E. Yu. (2006). IStarQuickGuide [Online]. Available: <http://istar.rwth-aachen.de/tiki-index.php?page=iStarQuickGuide> .
- [16] K. Sandkuhl, J. Stirna, A. Persson, and M. Wisotzki, "Enterprise modeling. Tackling business challenges with the 4EM method," in the Enterprise Engineering Series, Springer, 2014.
- [17] Respect IT, "A KAOS tutorial, objectiver," 2007. Available: <http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf>
- [18] A. Magal and J. Word, "Integrated business processes with ERP systems," Preliminary Edition. Hoboken, NJ: John Wiley and Sons, 2010.
- [19] H. Klaus, M. Rosemann, and G. G. Gable, "What is ERP?," Information Systems Frontiers, vol. 2(2), pp. 141-162, 2000. <http://dx.doi.org/10.1023/A:1026543906354>
- [20] H. Topi, W. Lucas, and T. Babaian. "Identifying usability issues with an ERP implementation," in Proc. ICEIS 2005, 2005, pp. 128-133.
- [21] I. Supulniece and J. Grabis. "Discovery of personalized information systems usage patterns," in Proc. ICIST 2010, 2010, pp. 25-32.
- [22] I. Supulniece, "Personalization of business process execution in enterprise applications," Summary of Doctoral Thesis, RTU Press, 2014.