Towards a Model of Context-Aware Recommender System

Lauma Jokste

Information Technology Institute, Riga Technical University, Kalku 1, Riga, Latvia
lauma.jokste@rtu.lv

Abstract. Users often have difficulties to use large-scale information systems efficiently because of their complexity. Additionally, these systems might be context dependent. If these context dependencies are taken into account during the system's run-time phase, the most appropriate functionality might be provided to users in the form of recommendations for each context situation. The paper proposes to account for the context dependencies by using context aware recommendations and outlines an approach for modeling such recommendations. This approach is based on methodological foundations of Capability Driven Development. The paper discusses a motivational case for developing context aware recommendations and presents the initial method for recommendation modelling.

Keywords: Recommender Systems, Recommendation Modeling, Context-Aware recommendations, Capability Metamodel.

1 Introduction and Related Work

Large-scale information systems (IS) are subject to dynamically changing circumstances in the IS delivery phase. The current situation can be described by different context data defined as *"any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between the user and the application, including the user and the applications themselves"* [1]. The context data can be taken into account in IS delivery thus increasing the usability and user satisfaction.

Complex and extensive IS might reduce the user satisfaction and, if possible, users might choose alternative ways of completing their tasks. For example, citizens are avoiding using public e-services and are favoring physical services. Recommender systems are widely used in order to improve the usage of software and tools. They provide suggestions by recommending the items that users might likely be interested in [2]. Recommender systems are increasingly popular. Many recommender systems focus on improving and evaluating the collaborative-filtering technique [3]. They use internal information about historical user activity, user profile information and other to match users with recommended items [4].

Gradually, the recommender systems start to use more varied data and data sources, for example, social network data [5][6][7]. In some cases the context information might be relevant to calculate the most appropriate recommendations for users by using context-aware recommendation systems [8]. Recent investigations present location-based [9] and weather-dependent [10] recommendation algorithms and methods. Discovering suitable context data is still a challenge in recommender systems evaluation [11]. The proposed approach in this paper includes the usage of different context data that can be retrieved from both internal and external data sources. Context processing includes not only reading the context data, but also context data analysis which helps to predict the context data and user behavior.

The recommender systems are usually item oriented and suggest the items in which users would be potentially interested in [12]. For instance, e-commerce sites such as Amazon.com recommend items users would be likely to buy [13], while content based systems recommend *things* based on textual analysis, e.g. in research area, citations can be recommended for the research by analyzing words in research papers [14]. In order to improve the recommender algorithms, hybrid recommender systems are developed by combining different recommendation algorithms and methods into one information system [15]. The approach proposed in this paper assumes that recommendations could be any kind of software entity and examples of recommendations include suggestions to execute a function, procedure, workflow or to perform data processing operations.

This paper presents the recommendation system proposal which starts with recommendation modeling. Different kind of context information can be used as an input to the recommendation module. The modeling phase is considered as significant since models can help to deal with complexity and are easy perceptible for stakeholders without any specific IT skills [16]. Recommendation modeling includes specifying a set of business rules that help to define the software entities context dependencies and appoints which recommendations should be run in each contextual situation.

Nowadays variability in IS delivery becomes more and more important. When business processes change, the software supporting these processes should be adjusted accordingly in order to fulfill the organizations goals [17]. Competitive software should be able to deal with the variability with minimal efforts. Changes in business processes can be affected by internal and/or external context. Adjusting software to changes in form of user recommendations by integrating Recommendation module in existing software allows to deal with variability without an important effort because recommendations can be easily changed in recommendation module without changing underlying software.

This paper analyses the potential of using context-aware recommendations and proposes an approach for modeling context aware recommender systems. The modeling approach is based on the Capability Driven Development (CDD) method used in development of adaptive systems [18]. The potential of using context-aware recommender systems is analyzed by exploring a use case from the e-government domain described in section 2. Section 3 provides an overview of the recommendation modeling method including a small representative example of recommendation modeling. Section 4 presents the conclusions and future work.

2 Motivational case

In order to justify development of context aware recommender systems, context dependencies in large scale IS are analyzed by using the case of the public electronic finishing service E-Loms, part of the municipal e-service platform www.epakalpojumi.lv developed by the Latvian company ZZ Dats Ltd. Currently, E-Loms provides several functions: to check information about lakes and rivers; to buy fishing licences online; to register the catch etc. E-service users' statistics (see the following figures) show that the number of licences sold varies in a wide range meaning that the usage of this service is affected by different circumstances. Several context elements affecting the e-service usage have been retrieved during the e-service usage analysis.

Assumption 1: E-service usage depends on the current and predicted air precipitation level. Fig. 1 shows the number of licences sold in September 2014 and the level of air precipitation in September 2014. There have been two pronounced rain periods and during these periods the number of fishing licences sold has rapidly decreased, thus the service usage dependents upon current and predicted air precipitation level.

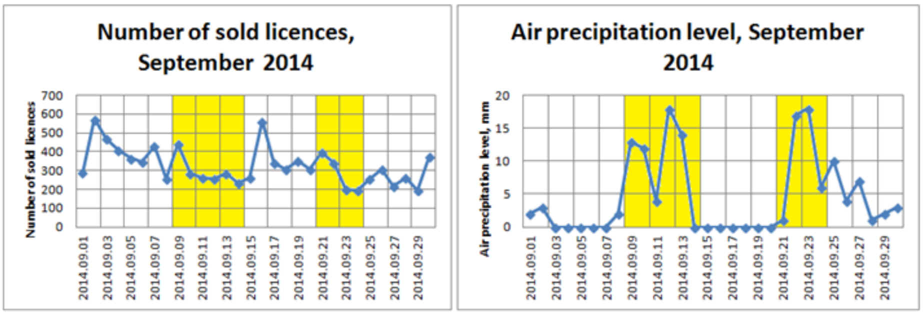

Fig. 1. E-service usage and air precipitation level dependencies

Assumption 2: E-service usage depends on the current and predicted air temperature. Fig. 2 shows the number of fishing licences sold and the average air temperature in February 2014. The same periods are highlighted in both charts. Periods with the lowest air temperature and the sharpest drops in temperature match with periods in which the largest numbers of fishing licences have been sold. In winter, ice fishing is very popular but it requires low air temperatures. This means e-service usage is highly dependent on the air temperature.

Assumption 3: E-service usage depends on calendar events. Analysis of e-service usage statistics in 2014 brings forward the following regularities:

- The beginning of January – many fishermen buy their yearly fishing licence;
- 1st of May – fishing season opening – in the end of April and the beginning of May the number of licences sold has rapidly increased;
- In the beginning of June the number of fishing licences sold has rapidly increased marking the beginning of the summer and vacations periods.
- The number of licenses sold is rapidly increasing on Fridays and Saturdays.

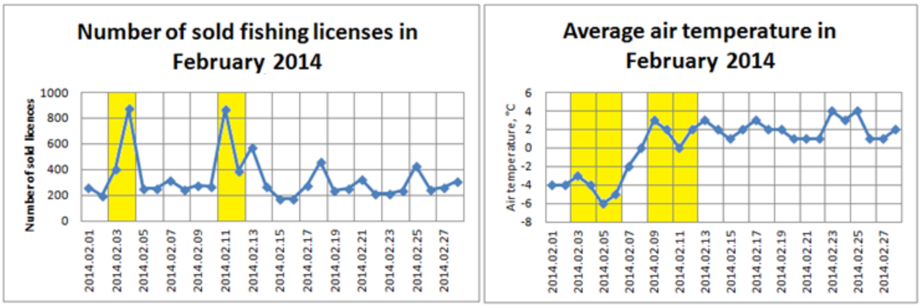

Fig. 2. Number of sold fishing licenses and average air temperatures in February 2014

The examples suggest that the usage of the E-Loms e-service is highly affected by different dynamical context data. When monitoring, predicting and taking into account the run-time context data, the e-service delivery might be improved by generating context aware recommendations for the service users.

There are many other context data that can be obtained, stored and exploited in order to apply recommendations that help users to navigate in large service catalogs or recommend users potentially the most important functions in each context situation. This could take the recommendations to a new level where context dependencies could be defined for software entities and the execution of other software entities can be provided to users in the form of recommendations.

3 Context-aware Recommendation Modeling

Usually recommendations are item-oriented where items are suggested based on user profile and historical user activity information which is mined from usage logs [19]. The use case example demonstrated in Section 2 proves that not only items can be recommended, but also different software entities can be used as recommendations.

The conceptual basis of context-aware recommendation algorithm design and development is based on the Capability metamodel [18] which is developed within the FP7 project "CaaS: Capability as a Service" and is a part of the Capability Driven Development approach. Capability term is important in recommendations modeling because context processing and recommendations application relates to IS ability to deliver a certain value to a user in changing context circumstances. This section presents the Capability metamodel which is expanded in order to fulfill the needs for modeling context-aware recommendations.

The specific aspects relevant for context-aware recommendation modeling which are added to the Capability metamodel are marked in a darker color in Fig. 3.

From the recommendation's perspective, the Software Entity is the detailed (i.e., executable) level of Capability and each Software Entity has a Context Set which defines the context information that affects the Software Entity delivery. The term Software Entity describes different executable artefacts and can be either software as a whole or a single workflow, procedure, function, job etc.

Recommendations conform to the definition of patterns which are "*reusable solutions for reaching business Goals under specific situational contexts*." [18] They are reusable solutions that can be applied to the Software Entity in a certain run-time context situation with the purpose of improving the usage of the Software Entity. The recommendations are user-oriented and improve the user satisfaction with a software, service or software entity.

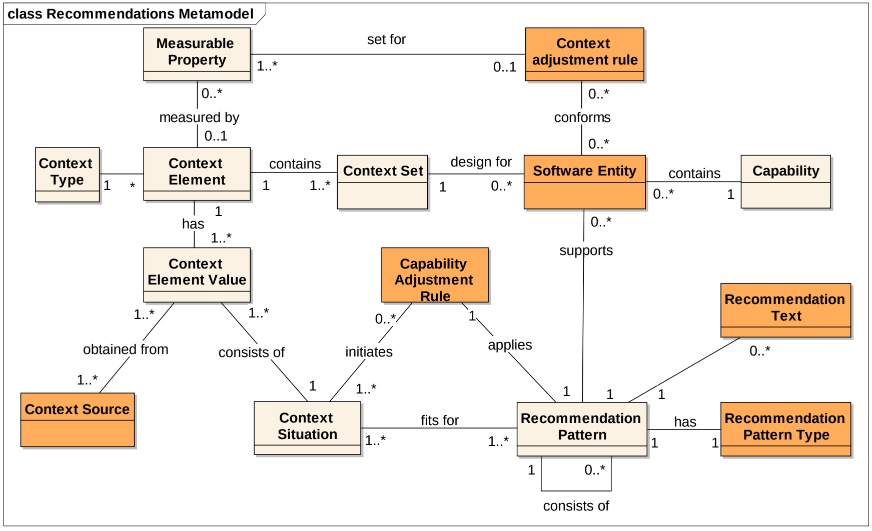

Fig. 3. Capability metamodel for modeling context-aware recommendations

A Recommendation Pattern can be applied automatically (e.g. automatic Software Entity highlighting) or recommended to a user with an informative notification which provides a choice to the user to run the recommended Recommendation Pattern. In the second case, the Recommendation Text should be defined as a separate class in the metamodel that marks separately modifiable text without changing the Recommendation Pattern itself. Each Recommendation Pattern can have multiple notification text values and the appropriate value for each context situation is defined in a Capability Adjustment Rule. In the example model (Fig. 6), the Recommendation Pattern is for instance a classificatory value ‘yearly license’ and the Recommendation Text recommends users to choose a licence type ‘yearly license’ instead of ‘one day license’.

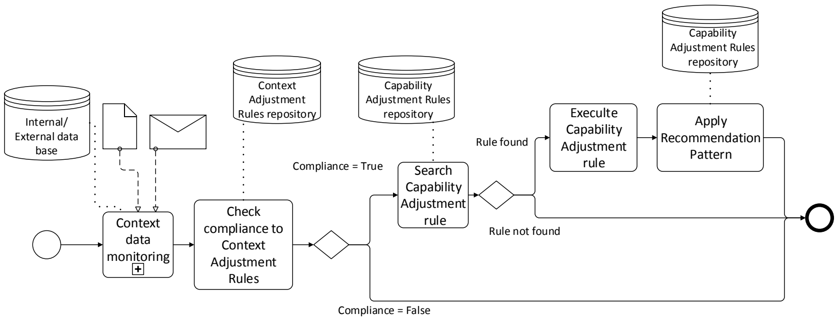
Capability Adjustment Rules repository
Context Adjustment Rules repository
Capability Adjustment Rules repository
Internal/ External data base
Executle Capability Adjustment rule
Apply Recommendation Pattern
Rule found
Compliance = True
Search Capability Adjustment rule
Context data monitoring
Check compliance to Context Adjustment Rules
Rule not found
Compliance = False

Fig. 4. **General recommendation algorithm execution business process model**

Context Adjustment Rules define the context ranges which affect the Software Entity while Capability Adjustment Rules initiate the recommendation which is appropriate for the certain run-time Context Situation. The form of Capability Adjustment Rules should be decided, for instance rules can be written as IF/THEN or Event-Condition-Action rules. The Recommendation Pattern Type specifies the type of recommendation, e.g. a function, procedure, textual notification, etc. The Context Source serves as a reference to the source from which the context data is obtained.

Based on a Recommendation model which is modelled according to proposed metamodel, a Recommendation module can be built. In Fig. 4 a general recommendation module execution process is given. The process includes concepts from the Capability metamodel. Context data monitoring is represented as a subprocess which will be specified in further research by taking into account different context sources that can be used to obtain context information relevant to IS delivery. In Fig. 5 and 6 an example of a recommendation model is given.

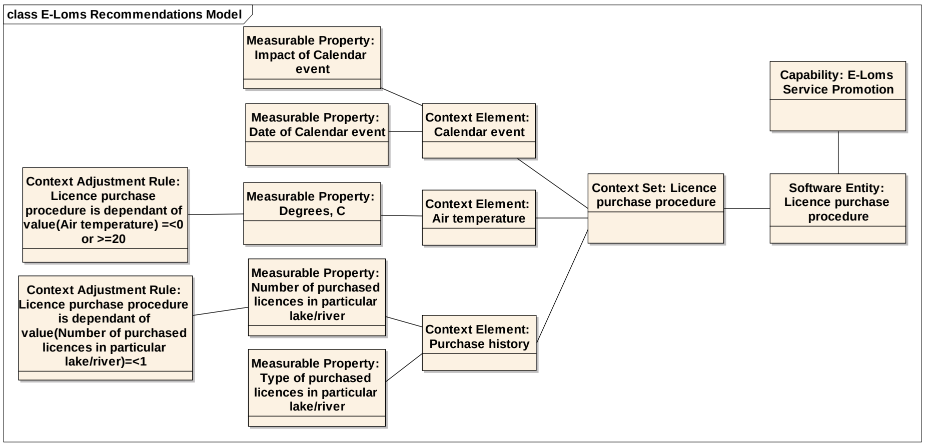
class E-Loms Recommendations Model
Measurable Property: Impact of Calendar event
Measurable Property: Date of Calendar event
Context Element: Calendar event
Capability: E-Loms Service Promotion
Context Adjustment Rule: Licence purchase procedure is dependant of value(Air temperature) =<0 or >=20
Measurable Property: Degrees, C
Context Element: Air temperature
Context Set: Licence purchase procedure
Software Entity: Licence purchase procedure
Context Adjustment Rule: Licence purchase procedure is dependant of value(Number of purchased licences in particular lake/river)=<1
Measurable Property: Number of purchased licences in particular lake/river
Context Element: Purchase history
Measurable Property: Type of purchased licences in particular lake/river

Fig. 5. Context modeling part of the E-Loms recommendation model example

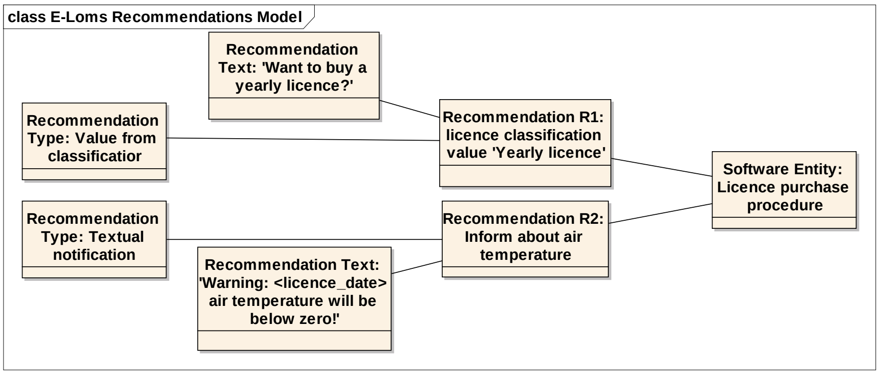
class E-Loms Recommendations Model
Recommendation Text: 'Want to buy a yearly licence?'
Recommendation Type: Value from classificatior
Recommendation R1: licence classification value 'Yearly licence'
Software Entity: Licence purchase procedure
Recommendation Type: Textual notification
Recommendation R2: Inform about air temperature
Recommendation Text: 'Warning: <licence_date> air temperature will be below zero!'

Fig. 6. Recommendation modeling part of the E-Loms recommendation model example

The Context Situation is a run-time element and cannot be modeled during the design phase. For instance, if a user is buying a one-day license in a lake where he already has bought one-day licenses, then another value from the classificatory can be recommended – 'a yearly license'. Recommendations can be run by a Capability Adjustment Rule, e.g. *If number of bought licenses in lake X >=1, THEN run recommendation R1.*

4 Conclusion and Future work

In this paper we discuss the usage of the Capability metamodel in the design of context-aware recommendations for software entities in order to increase user satisfaction. The example demonstrated in the paper proves the potential of generating recommendations based on run-time context data.

Current recommendation algorithms and methods are mostly item-oriented and based on a limited amount of data (user profile data, user activity, similarities between users and items etc.). The delivery of many software and software entities is affected by different context information. The achievement of business goals can be improved by applying context aware recommendations in software delivery.

The recommendation modeling method would allow designing modifiable recommendations more efficiently. In the Recommendation module, recommendations can be modified by defining or changing Capability Adjustment and Context Adjustment Rules, recommendation values, or modifying recommendations in the recommendation repository. Once the module is built, stakeholders without specific IT skills should be able to manage and maintain the recommendations.

This paper presents the initial proposal for context-aware recommendations modeling. The context-aware recommendation modeling and designing method should be improved and developed in detail in following aspects:

- The form for defining the Context Adjustment Rules and the Capability Adjustment Rules should be specified;
- The context processing subprocess should be specified;
- The technical solution for developing and implementing the recommendation module should be developed;
- The recommendation modeling method and recommendation module should be validated.

References

1. Dey, A.K. Understanding and Using Context. In: Personal Ubiquitous Computing, vol 5(1), pp. 4–7 (2001)
2. Rubens, N., Kaplan, D. and Sugiyama, M. Recommender Systems Handbook: Active Learning in Recommender Systems (eds. P.B. Kantor, F. Ricci, L. Rokach, B. Shapira). Springer (2011)

3. Hang, S., Hui, P., Kulkarni, S. R. And Cuff, P.W. Wisdom of the Crowd: Incorporating Social Influence in Recommendation Models. Proc. ICPADS ’11, IEEE, pp. 835-840 (2012)
4. Burke, R. Hybrid Recommender Systems: Survey and Experiments. In: User Modeling and User-Adapter Interaction, Vol 12, Issue 4, pp 331-370 (2002)
5. Aranda J., Givoni I., Handcock, and Tarlow, D. An Online Social Network-based Recommendation System, Technical report (2010)
6. He J. and Chu, W.W. A Social Network Based Recommender System, Annals of Information Systems: Special Issue on Data Mining for Social Network Data (AIS-DMSND) (2010)
7. Shang, S., Hui, P., Kulkarni, S.R. and Cuff, P.W. Wisdom of the crowd: Incorporating Social Influence in Recommendation Models. In: Proceedings of the IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS '11), pp. 835-840 (2011)
8. Adomavicius, G. and Tuzhilin, A.Context-Aware Recommender Systems. In: Recommender Systems Handbook, pp. 217-253 (2011)
9. Savage, N.S., Baranski, M., Chavez, N.E. and Höllerer, T. I’m feeling LoCo: A Location Based Context Aware Recommendation System. Proceedings of the 8th International Symposium on Location-Based Services (2011)
10. Braunhofer, M., Elahi, M., Ge, M., Ricci, F. and Schievenin, T. STS: Design of Weather-Aware Mobile Recommender Systems in Tourism. In: Carolis B.N., Carolis E. (eds.) Artificial Intelligence meets Human Computer Interaction, CEUR Workshop proceedings, vol 1125 (2013)
11. Yujie Z. and Licai W. Some challenges for context-aware recommender systems. In: Proceedings of the 5th International Conference on Computer Science and Education (ICCSE), pp. 362 – 365 (2010)
12. Linden, G., Smith, B. and York, J. Amazon.com Recommendations. Industry report. Published by the IEE Computer Society (2003) URL: http://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf Last accessed: March 26, 2015.
13. Duma, D. and Klein, E. Citation Resolution: A method for evaluating context-based citation recommendation systems. In: Proceedings of the 52nd Annual Meeting in the Association for Computational Linguistics, Volume 2, pp. 358-363 (2014)
14. Hussein, T., Linder, T., Gaulke, W. and Ziegler, J. Hybreed: A software framework for developing context-aware hybrid recommender systems. In: User Modeling and User-Adapted Interaction, Vol 24. Issue 1-2, pp. 121-174 (2014)
15. Ricci, F., et al. (eds.). Recommender Systems Handbook, Springer Science + Business Media (2011)
16. Carvalho, J. A. Strategies to Deal with Complexity in Information Systems Development. URL: http://www3.dsi.uminho.pt/jac/SI/zdocumentos/complexity.pdf Last accessed: March 26, 2015
17. Soffer, P. Analyzing the Scope of a Change in a Business Process Model, Proceedings of Int’l Conference on Advanced Information Systems Engineering Workshops (2004)
18. Stirna, J., Grabis, J., Henkel, M. and Zdarvkovic, J. Capability Driven Development – and Approach to Support Evolving Organizations. In: The Practice of Enterprise Modeling: PoEM, Germany, Rostok, 7-8 November, 2012. Berlin: Springer Berlin Heidelberg, pp.117-131 (2012)
19. Koenigstein, N. and Koren, Y. Towars scalable and accurate item-oriented recommendations. In: Proceedings of the 7th ACM conference on Recommender systems (2013)