

Using Goal Models to Visualize and Prioritize Requirements for Learning Management Systems

Viktor Lantz¹, Jennifer Horkoff^{1,2}, and Sara Alibrahim¹

¹ University of Gothenburg, ² Chalmers University of Technology, Sweden
 guslantvi@student.gu.se, jenho@chalmers.se, gusalibrsa@student.gu.se

Abstract. Learning Management Systems (LMSes) handle all aspects of the learning process, a crucial part of educational technology. This study elicits and models the functional and non-functional requirements of two academic and one industrial LMS. The overall purpose is to provide a general requirements model, grounded on existing systems and collected evidence, to aid future LMS development. Goal models, partially validated with interviews, are used to provide a visual presentation of LMS functional and non-functional requirements. A survey was used to prioritize these requirements, obtaining 63 responses from students and academics in Software Engineering. The prioritized requirements are used to create a general LMS requirements model.

1 Introduction

Educational technology is defined as “the study and ethical practice of facilitating learning and improving performance by creating, using, and managing appropriate technological processes and resources” [17]. Learning Management Systems (LMSes) are a core educational technology, used for delivering, tracking and managing courses or training programs in academic and industrial settings [8]. As LMSes become widespread, it is particularly important to understand what makes such systems successful. In systems and software engineering, functional requirements (FR) capture the functionality of the system, while non-functional requirements (NFRs) capture quality attributes [3]. In this work we aim to determine which FRs and NFRs make an LMS effective.

Conceptual modeling has long offered ways to capture and understand FRs and NFRs, particularly using goal modeling (e.g., [10, 5]). In this paper, we use goal modeling to analyze and understand LMSes. Specifically, we investigate which requirements can be extracted from current LMSes and prioritize them from a user perspective, helping to determine the most important aspects of the system. This process can be thought of as reverse engineering, the process of “extracting knowledge or design information from anything man-made” [7].

Existing work has looked at requirements for effective LMSes, e.g., communication, course content management, and evaluation [9]. However, this work has not made any attempt to prioritize possible LMS requirements based on user

feedback, or to provide visual summaries of these requirements. Furthermore, this study is unique in that it takes a “bottom-up” approach to LMS analysis, extracting requirements from existing academic and industrial systems.

The goal of this work is to produce a general model containing the most highly prioritized LMS requirements, helpful for development, extension or evaluation of new or existing LMSes. Instead of focusing on education for conceptual modeling, here we focus on the use of conceptual modeling for education. More generally, we develop and illustrate a method for how to procure and analyze prioritized requirements using reverse engineering and goal models.

This paper is organized as follows: Sec. 2 summarizes work understanding LMS requirements and goal modeling for pedagogy. Sec. 3 describes our study method, while Sec. 4 summarizes results, including models. We end with a discussion (Sec. 5), conclusions and a consideration of future work (Sec. 6).

2 Related work and Background

LMSes. Richardson describes how to use available resources to provide a high quality, flexible learning environment for staff and students [16]. Chan presents experiences in developing and evaluating a web-based learning product [14]. Design considerations focus on the lack of social interaction between students and teachers when using LMSes, lack of motivation and other pedagogical barriers. These works contribute to our elicitation of LMS requirements.

The thesis by Faxén [9] has two main objectives: the identification of technology that improves academic e-learning by comparing LMSes and the establishment of requirements for LMSes in an academic environment. Faxén establishes thirty requirements arranged in eleven functionality subsets for an academic LMS. These eleven categories were then ranked, with course content management, evaluation and communication determined to be the most important subsets. Faxén evaluates different LMSes to determine if they supported this functionality. In contrast, our study extracts requirements from existing LMSes, creates visualizations of these requirements, and ranks requirements based on user feedback. Our work focuses on both functional and non-functional requirements, whereas Faxén evaluates LMSes as per 12 functional requirements only.

Kunz [12] delves into the reasoning behind successful LMS design elements and also what a system needs to provide. The work investigates a development agenda for the next generation of LMSes which includes components that already exist, components that need to be improved and components that have to be developed. This work can supplement our findings in terms of potential LMS functionality, however, this functionality has not been modeled or prioritized.

Goal modeling and pedagogy. Goal modeling has a long history in requirements engineering, conceptual modeling, and other areas in capturing FRs and NFRs with a social perspective [10]. In this work, the syntax of the iStar 2.0 standard was generally followed [5], see Fig. 2 for an example iStar model and Sec. 4 for description of the contents.

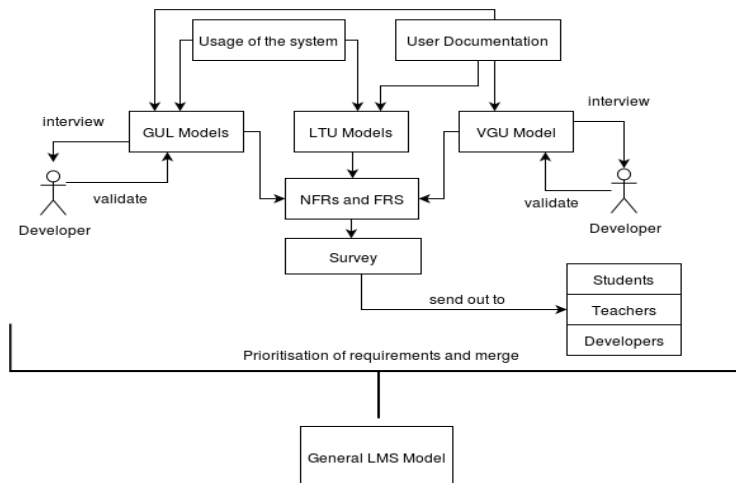


Fig. 1: The steps that lead to the creation of the general model

The general model produced in this work can be thought of as a pattern for LMS development. Previous work has used goal models to capture requirements patterns (e.g., [4]). To our knowledge, no LMS requirements pattern models exist.

Recent workshops and papers have focused on pedagogy for goal modeling, but very little work uses goal modeling for pedagogy. Work by Monsalve & Leite has used iStar for exposing the motivations behind teaching, providing educational transparency [15]. The LMS requirements models developed in our work could be used similarly, for transparency in terms of LMS goals and functionality.

3 Methodology

The individual LMS models were created through a combination of manual reverse engineering, looking at three existing systems, and requirements from the literature. The results were prioritized via survey to produce the general model. The entire process is summarized in Fig. 1.

3.1 Data collection

We examined three LMSes: Göteborgs Universitet Lärplattform (GUL) (a subset of the Ping-Pong platform), Luleå Tekniska Universitet (LTU) and Volvo Group University (VGU). Data collection began by collecting the product and user documentation of the three systems. In addition to the user manuals, we were able to access, use and evaluate the functionality of the GUL [2] and LTU [1] systems. The VGU Navigator was inaccessible to non-employees; however, a thorough demo was provided by the company. Once the documentation and systems were analyzed and the requirements extracted, the modelling began.

3.2 LMS goal modelling

The three systems were modeled, with the GUL and LTU systems broken down into comparable subsets of functional features. Specifically, we used the func-

tionality subsets as per [9] to organize found requirements: Communication, Course Content Management and Additional Services. Briefly, *communication* includes functionality and requirements which “Enables communication between administrators and learners” [11], “Search and identify learners and deliver targeted courses, news, references, and other information to continually engage them” [11]. *Course content management* includes functionality and requirements related to “Assignment upload, uploads of course assignments for the students” [9], “Target content to the correct individuals or groups” [11] and “Course object reuse, possible for the teacher to create courses from existing course objects” [9]. *Additional services* contain functionality and requirements which exists in the system but does not fall into the two aforementioned subsets. This includes “ePortfolios which could be used by students as a knowledge construction and reflection space” [12], “Manage user registrations and profiles” [11] and compatibility with third-party software [9].

The reverse-engineered requirements were divided into three models as per these categories, in order to increase model clarity and reduce the size of the overall models. The goal models were created interactively. The process started with the creation of the actors, then adding the functional requirements, and finally creating the dependencies between the actors. The next step was to create the soft goals which represent the non-functional requirements of the system. After modelling, the next steps were to validate content via interviews and create the survey to prioritize the elicited functional and non-functional requirements.

Semi-structured interviews were conducted to validate the model content and to discover missing requirements. Interviews were conducted in person with individuals that have developed and contributed to the VGU and GUL LMSes. One interview was carried out with a senior management consultant at Volvo, while another was carried out with a developer at Ping-Pong (framework for GUL). System developers for the LTU model were unavailable, so validation of this model is left for future work, ideally as part of the development of future LMSes. Overall, the interview findings did not result in changes to the models.

3.3 Survey

The created models helped to formulate the questions in the survey by providing a collection of the most commonly occurring FRs and NFRs in the three LMSes. Functionality which appeared in all three or two of the LMSes was included in the survey. Some further functionality, deemed important or interesting, was included, but survey space restriction did not allow all functionality to be added.

The survey started with basic questions about the respondent’s role and LMS use. It was then divided into four sections, three covering the functionality subsets as per [9], and one focusing on NFRs. Each section included questions asking about the importance of a requirement using a Likert scale. For example, in Course Content Management section, “How important is being able to obtain feedback from a teacher? Not important (1) to very important (5)”, mapping to the “Assignment feedback” FR coming from the goal model. The NFR section

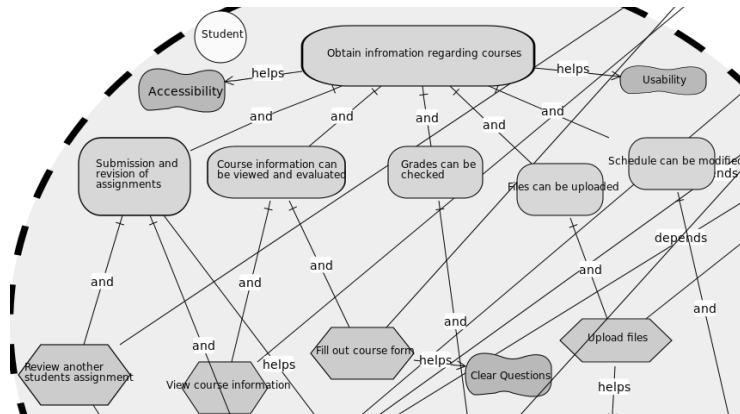


Fig. 2: A subset of the GUL course content management model

asked respondents to rate the importance of NFRs for LMS systems, providing an explanation of the NFR in that context. For example, “How important is the availability of the system? Availability is the proportion of time a system is in a functioning condition. So if the system is not available then it is not working as expected for a period of time.”, corresponding to the “Availability” NFR found through goal modeling. The full survey can be found in [13].

The Likert responses were evaluated by calculating a weighted average value, the median and the modal value. In the weighted average, the sum of the number of responses, multiplied by the Likert value, is divided by the total number of responses. Using these values, the importance of each requirement from the users’ perspective was evaluated, helping to decide which requirements were to be included in the general LMS model. Certain questions in the survey were open ended, asking users to provide important requirements which were missing from the questions: “What is the most important functionality to enhance (specific section) and why?”. The qualitative data entered for these questions was analyzed and taken into consideration when carrying out the prioritization.

The survey was piloted with three students and one teacher, with iterations made to improve clarity. It was distributed via e-mail and message boards to the Software Engineering Division within the Computer Science and Engineering Department of GU/Chalmers, as both students and professors there were likely to be experienced with these or similar software systems. Responses came from students (users), teachers (administrators) and developers (creators).

4 Results

4.1 LMS goal models

Our process produced seven models: communication, course content management, and additional services models for the GUL and LTU systems, and one VGU model. The VGU model focused only on the course content management section, as the other requirements categories did not appear in the system. The largest model contained roughly 100 elements. We illustrate details of a GUL model subset focusing on course content management (Fig. 2). A partial summary list of FRs and NFRs for course content management for all three systems

	GUL	LTU	VGU
Submission of assignments	x	x	x
Review of assignments	x	x	-
Assignment feedback	x	x	-
Grade checking	x	x	-
Personal file storage	x	x	-
Course evaluation	x	-	x
Quiz online	-	x	-
Exam online	-	x	x
Course online	-	-	x
Training record	-	-	x
View personal information	x	x	x
Search for training	-	-	x
Withdraw from training	-	-	x
Non-Functional requirements (NFRs)			
	GUL	LTU	VGU
Availability	x	x	x
Usability	x	x	x
Privacy	x	-	x
Performance	x	-	-
Accessibility	x	x	x
Reuseability	x	x	-
Interoperability	-	x	x
Certifiability	-	-	x
Maintainability	-	-	x
Responsiveness	-	-	x
Modifiability/Configurability	-	-	x

Table 1: Partial list of the requirements included in the Course Content Management subset of each LMS. (x = included) (- = not included)

can be seen in Table 1. These requirements map to elements in the goal models. Although the NFRs listed are generic, they can be contextualized to LMSes via their position and connections in the model.

The GUL course content model in Fig. 2 focuses on the student actor. The models provide a hierarchy of goals (orange ovals). The student’s highest goal is to “Obtain information regarding courses”, decomposed to sub-goals. The goals are decomposed to tasks that achieve them (green hexagon). For example, the goal “Course information can be viewed and evaluated” is decomposed to: “View course information” and “Fill out course form”. The dark orange shapes are the soft goals which represent NFRs. Tasks and goals help achieve soft goals. For example, “Fill out course form” helps to achieve the soft goal “Clear Questions”.

4.2 Survey Results

The survey received 63 responses: 43 students, 18 teachers and 2 developers. The full survey responses including responses for the open ended questions can be viewed in [13]. Participants found the communication aspect of an LMS beneficial. The course content management aspects were also deemed important by the respondents, while the additional services aspect of an LMS were generally deemed less important. Most non-functional requirements in the survey were deemed as important. Overall, the responses solidified the need to include certain requirements in the general model.

4.3 General LMS Model

Using our prioritization results, we created two versions of a general LMS model: the high-priority and the lower-priority model. Both models can be split into the

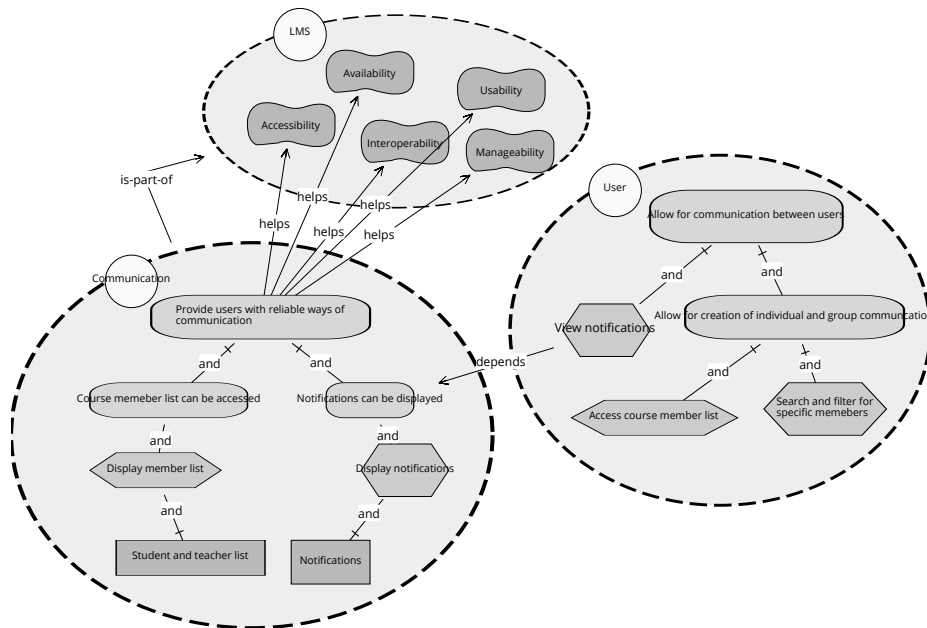


Fig. 4: Communication High Priority General Model

Modeling experiences. We found that having all of the functionality in one model led to a lot of disorganization, therefore we divided the functionality into three subcategories and views for readability. The first and last authors had to familiarize ourselves with the iStar framework and how to correctly represent elicited requirements. The FRs of the LMSes were easy to identify from user documentation, they were also simple to capture within the models. Drawing dependencies seemed logical and did not cause any issues.

On the other hand, the NFRs were harder to identify. These requirements are not obvious when analyzing a system and are often not stated in the user documentation. Further effort was needed to identify these NFRs; first, to identify which requirements were relevant; second, to identify which of the systems and sub-models they applied to; and third, to identify which specific goals that were ‘helped’ by these requirements. This was the most difficult modeling task.

Threats to Validity. We follow criteria for threats to validity as per Easterbrook et al. [6]. *Construct Validity.* The use of goal models offers a possible construct validity threat, as the models may not have been used as their underlying theory intended. However, we mitigated this threat by following iStar literature and involving a goal modeling expert (one of the authors). The subjects interviewed had software engineering backgrounds and an understanding of modeling languages, however, they were unfamiliar with iStar. To mitigate this threat, a short introduction about the framework was given. Still, it is difficult to judge how well the interviewees understood the models.

Internal Validity. The design of the survey may have been misleading, too long, or incomplete. To mitigate some of these threats, we conducted four pilot tests. *External Validity.* The analysis was carried out on three LMSes. Ideally,

more systems would be analyzed; however, we believe these three systems covered much of the possible LMS functionality and covered both industrial and academic scenarios. Furthermore, all LMSes studied were deployed in Sweden. Future studies should compare our findings to LMSes used in other countries.

The survey focused on students and professors who are active within the field of Computer Science and Software Engineering (SE). This could limit the results gathered since SE students may not be representative of the entire LMS user population. Furthermore, there were only two subjects for the interviews; however, each subject represented their respective company.

Reliability. Our extraction of system functionality when reading documentation or using the systems could have been biased by our experiences. This threat was mitigated via interviews validating the resulting models. One of our authors is an expert in goal modeling, and she provided guidance on the model creation. This situation may not be reliably recreated in further studies; however, material like the iStar 2.0 Standard is intended to make such modeling more accessible [5]. If the survey were to be distributed to a different audience, the prioritization of requirements might be different, resulting in a different general model.

6 Conclusion and Future Work

This paper explores the FRs and NFRs which are present in three LMSes. The requirements were extracted through the use of manual reverse engineering, including analysis of user documentation, the systems, and examining the content of relevant literature. The requirements were modeled using the iStar modeling language in order to create overviews of LMS functionality.

The initial models provided the necessary information required to construct a survey which allowed user validation of requirement prioritization. Finally, a general model was created from the prioritized requirements, displayed in smaller views, potentially useful for the creation or evaluation of further LMSes. Our experiences lead us to believe that the process of requirements engineering modelling is an effective way to capture the FRs and NFRs of an LMS. We believe that our process (Fig. 1) could also be applied when attempting to capture the requirements of other software systems.

Future work should examine a larger, more international sample of LMSes in order to validate the FRs and NFRs. Furthermore, this work builds on the responses of students and professors within SE. Future work can be carried out to investigate if similar functionality is considered as important within other fields. Further work should validate the general model by creating or modifying an LMS based on the identified requirements.

Acknowledgments

We thank our contacts at Ping-Pong and Volvo for their participation.

References

1. Canvas Student Guide, <https://sv.guides.instructure.com/m/38870>

2. Ping Pong 6.0 User Guide, [http://www.uadm.uu.se/upi/resurser /ping-pong/userguideline.pdf](http://www.uadm.uu.se/upi/resurser/ping-pong/userguideline.pdf)
3. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering
4. Chung, L., et al.: Capturing and reusing functional and non-functional requirements knowledge: A goal-object pattern approach. In: Information Reuse and Integration, 2006 IEEE International Conference on. pp. 539–544. IEEE (2006)
5. Dalpiaz, F., Franch, X., Horkoff, J.: istar 2.0 language guide. arXiv preprint arXiv:1605.07767 (2016)
6. Easterbrook, S., Singer, J., Storey, M.A., Damian, D.: Selecting empirical methods for software engineering research. Guide to advanced empirical software engineering pp. 285–311 (2008)
7. Eilam, E.: Reversing: secrets of reverse engineering. John Wiley and Sons (2005)
8. Ellis, R.K.: Field guide to learning management systems. Learning Circuits (2009)
9. Faxén, T.: Improving the outcome of e-learning using new technologies in LMS systems and establishing the requirements for an lms system in an academic environment. Masters of Software Engineering and Management. University of Gothenburg (2011)
10. Horkoff, J., Aydemir, F.B., Cardoso, E., Li, T., Maté, A., Paja, E., Salnitri, M., Mylopoulos, J., Giorgini, P.: Goal-oriented requirements engineering: a systematic literature map. In: Requirements Engineering Conference (RE), 2016 IEEE 24th International. pp. 106–115. IEEE (2016)
11. Kamel, M.: Learning management systems: Practical considerations for the selection and implementation of an e-learning platform for the navy (2008), <https://calhoun.nps.edu/handle/10945/33783>
12. Kunz, P.: The next generation of learning management system (LMS): Requirements from a constructivist perspective. In: EdMedia: World Conference on Educational Media and Technology. pp. 300–307. (AACE) (2004)
13. Lantz, V., Alibrahim, S.: Using goal models to understand and prioritize requirements for e-learning management systems (bachelor thesis) (2017), <https://tinyurl.com/y7df8xlv>
14. Lau, R.W., Li, F.W.: Advances in web-based learning. ICWL 2006 pp. 165–175 (2006)
15. Monsalve, E.S., do Prado Leite, J.C.S.: Using i* for transparent pedagogy. In: iStar. pp. 25–30 (2013)
16. Richardson, J., Swan, K.: Examining social presence in online courses in relation to students' perceived learning and satisfaction (2003)
17. Richey, R.C.: Reflections on the 2008 AECT definitions of the field. TechTrends pp. 24–25 (2008)