AutoSys: Supporting Distributed Teams Performing Systematic Studies

Paolo Tell - Steven Jeuris
Software Systems Section
IT University of Copenhagen, Denmark
pate@itu.dk - sjeu@itu.dk

September 16, 2015

Version:

Version	Date	Note
0.1	07.aug.15	Input from M. Kuhrmann
0.2	30.aug.15	Adaptation, correction, and refinement.
0.3	01.sep.15	Refinement.
1.0	02.sep.15	Delivery.
1.1	08.sep.15	Included a BF related to caching.
2.0	16.sep.15	Included APIs.

Roles:

Role	Name
Client	Paolo Tell
Application domain expert	Paolo Tell
Chief software architect	Steven Jeuris
Solution domain consultants	Jacob Benjamin Cholewa Mikkel Bybjerg Christophersen Mikael Lindemann Jepsen

1 Introduction

The body of knowledge in research is continuously growing, and it is hard for researchers to determine the exact state-of-the-art and state-of-practice in a given domain. To structure a specific research area and get insights into existing contributions, mainly two approaches [4] are used: Systematic Literature Reviews (SLR) and Systematic Mapping Studies (SMS).

However, performing these type of studies is rather challenging as: (i) they tend to be significantly time consuming due to both their intrinsic systematic nature and the amount of data they are required to handle, (ii) they are highly susceptible to errors as currently almost entirely human dependent, and (iii) they require the cooperation of a team of researchers to ensure quality and avoid bias. For these reasons such methods require heavy tool support, which is currently non-existing to the best of the author's knowledge.

Therefore, the purpose of the system you have been hired to design and develop is to provide appropriate tool support for these methods.

2 The SMS and SLR Processes

The main difference between the two approaches lays in the scope of the sought research questions. While during the conduction of an SLR the aim is to precisely answer one or more focused research questions, during an SMS the objective is to provide a wide overview of a research area, to establish if research evidence exists on a topic and an indication of the quantity of the evidence [3]. Sample questions of an SLR are: "Which aspects of the UML have been most subjected to empirical study?" [1] or "Does the type of actual usage measure (subjective or objective) affect the accuracy of TAM predictions?" [6]. Whereas, in the case of an SMS: "What usability evaluation methods have been employed by researchers to evaluate Web artifacts, and how have these methods been used?" [2]) or "Which software tools (commercial, free or research based) are available to support Global Software Engineering?" [5].

Regardless of this difference, the processes followed to perform these two type of studies share most of the fundamental activities. Figure 1 depicts such activities and the expected outcomes of each of them. In the remainder of this section the main activities and outcomes will be described; however, if necessary, further more comprehensive details can be discussed with the application domain expert or found in [3], which describes the SLR approach in details and dedicates a section on the SMS differences.

Planning the review

During this phase the study is planned and a document containing all the information related to the details of the study is created. First, the research question(s) is specified. Second, the inclusion/exclusion criteria are identified; these are used during the study to choose whether a paper retrieved should be

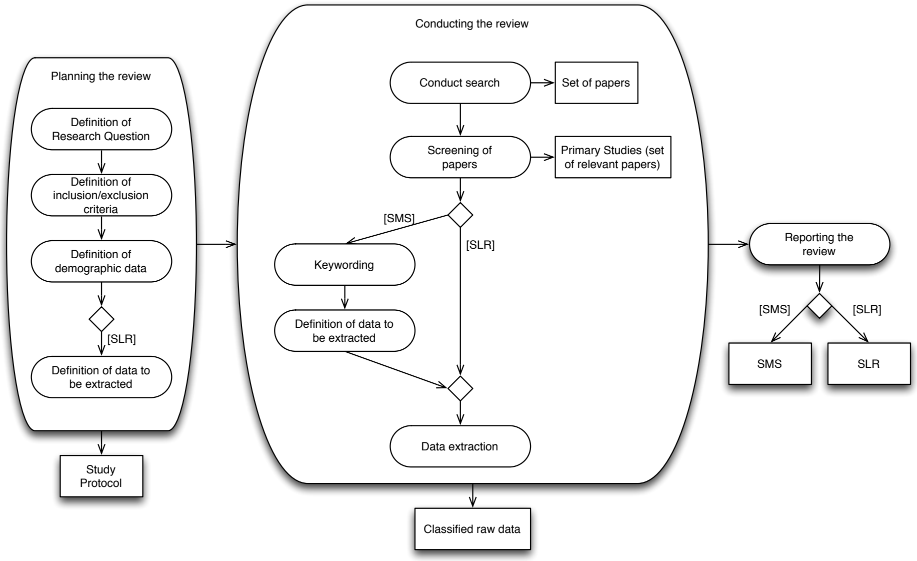
Planning the review
Definition of Research Question
Definition of inclusion/exclusion criteria
Definition of demographic data
[SLR]
Definition of data to be extracted
Study Protocol
Conducting the review
Conduct search
Set of papers
Screening of papers
Primary Studies (set of relevant papers)
[SMS]
[SLR]
Keywording
Definition of data to be extracted
Data extraction
Classified raw data
Reporting the review
[SMS]
[SLR]
SMS
SLR

Figure 1: Systematic study process (UML activity diagram).

included as relevant or not. The set of included relevant studies is also known as set of primary studies. Examples of these inclusion/exclusion criteria are: “the paper was published after 2010”, “the paper is longer than 6 pages”, “the paper is on software engineering”, “the paper reports a study in a company (not a laboratory experiment)”, etc. Third, interesting aspects of the papers that will be selected as primary studies are outlined. Usually, and especially in the case of an SMS, at this stage only the aspects related to the demographics are selected. Samples of these facets are: the publication year, the authors, the publication venue, etc. Commonly, these information are afterwards used to provide the demographic distributions of the primary studies over, for instance, the years or the publication venue they were published in. In the case of an SLR, the information sought by the review is already known; therefore, the contribution facets can also be defined. These represent the data that will have to be extracted from the paper to answer the research question(s). An example of these based on one of the previous research questions could be: the aspects of the UML that are subject to empirical study in the paper. In the case of an SMS, these facet are usually not yet clear as one of the purposes of the SMS itself is to define such facets; therefore, they will be defined at a later stage of the review.

Conducting the review

This activity represents the core of the study in which the relevant papers are identified and the data required to answer the research question(s) extracted. The smallest set of activities comprising this phase are: conducting the search, screening the publications retrieved from the search, and extract the data from the set of primary studies (i.e., the set of papers selected as relevant from the ones retrieved during the search). Each of these activities produces an artifact, namely: the set of retrieved papers, the set of primary studies, and the classified raw data respectively. In the case of an SMS, in which the contribution facets have not yet been defined, after the selection of the primary studies, these emerge in the form of keywords during the screening of the primary studies. As an example of this process, the research team could identify as keyword "the study type" (e.g., field study, laboratory experiment, etc.) or "the size of the teams reported in the study". For each of these topics, keywords are identified to be assigned to the primary studies and eventually detail a classification schema that will define additional data to be extracted from the primary studies.

Reporting the review

Finally, in the reporting phase, data is aggregated and presented in all sorts of different formats to be used in the actual review document. Bubble graphs, bar charts, tables, etc. are examples of these formats, and they are generated to be included in a comprehensive document meant to answer the research question(s) defined in the planning.

3 User Requirements

As it can be intuitively predicted, to perform a significant part of the activities described in the previous section tool support can—and should—be used. Moreover, in some cases the tool support could go as far as automating some aspects of these activities.

The intended software system shall provide (distributed) research teams with a sophisticated tool support to conduct comprehensive secondary studies (i.e., SMS or SLR). Notably, the intended software systems shall support the activities of planning the review and conducting review. The final activity of reporting the review is not part of the scope of this system; however, the generation of more elaborate outputs (e.g., graphs) could be supported.

As the research community asks for such tool support, the system shall be developed in a way that allows for easy deployment and dissemination. Among others, a goal of this system should be to be easy and quick to distribute. This would allow to include feedback mechanisms in the system to allow end-users to propose improvements.

Even though with significant limitations, some of the basic support that researchers performing systematic studies need can be provided by a spreadsheet software. Therefore, to ensure that the system will be chosen over the competition, the system will have to compare against such spreadsheet software (e.g., Microsoft Excel).

Minimum set of features.

In this section, an initial set of basic features (BF) requested by the client is outlined. This set is prone to modifications and extensions; however, for the purpose of the exam it can be considered as the minimum set required to pass. During the next classes a few additional mandatory requirements will be introduced.

BF.1 - The system shall support the management of research teams.

BF.2 - The system shall support distributed research teams to work on the same study.

BF.3 - The system shall support multiple studies concurrently.

BF.4 - The input data used when performing SLRs and SMSs are list of references. Several standards describing reference entries are available (e.g., .bib, .ris, etc). Among these, the most known and used is the bibtex format, which you are already familiar with (see last week exercise for an example). The system shall support at least importing bibtex files.

BF.5 - The system shall allow to review the status of the review at any point, also showing the rational behind some data (e.g., keep track of the reason behind the exclusion of a publication).

BF.6 - The system allows for exporting plain data sets for further processing in different formats. The minimum format is csv (comma separated values).

BF.7 - The system supports the creation and management of inclusion and exclusion criteria.

BF.8 - The system allows for the creation and management of classification criteria (i.e., research type facets and contribution type facets).

BF.9 - The system shall support the definition of the process to be followed during the various activities comprising the review. The system should for instance support the visualization of a selection of data from the publications to be reviewed by the researchers performing the study. Also, the system should allow for the assignment of specific sets of publications to specific members of the research team (e.g., 50% of publications to be reviewed by researcher A and 50% to be reviewed by researcher B).

BF.10 - The system shall support the visualization of statistics presenting the status of the review.

BF.11 - The system allows for generating the research protocol.

BF.12 - The system must be easy to deploy and install.

BF.13 - The system must be complemented with an installation manual.

BF.14 - The system must be complemented with a user manual.

BF.15 - The system should be publishable as Open Source Software. That is, that appropriate licenses must be evaluated. If 3rd party components are required, their used should be discussed with the client.

BF.16 - The system shall be developed using C# and WPF.

BF.17 - The system user interface must be replaceable by a newer version later.

BF.18 - The system shall support the manipulation of multiple primary studies simultaneously, which is primarily required for quick overviews.

BF.19 - The system shall allow users to work offline.

4 Proposed architecture

We propose to divide the architecture of the system into two main components. A 'Systematic Review Client', which is the user interface through which researchers participating in the SMS or SLR manipulate the study data. The client needs to be able to set up a team of researchers, as well as set up the specifications for multiple studies. These configurations are passed along to a 'Study Configuration Server', which is also responsible for handing out review tasks to the client while working on the study. A 'Study Configuration Server', which is the back-end responsible for storing study data, and providing the client with the appropriate requested data. It is up to the server to implement the logic for which review tasks need to be done. These two components can be developed by two teams in parallel. Initially, the client team will need to focus on the implementation of the user interface (UI), ensuring that an external UI component to configure a study can be embedded later in the project. The server team will first need to ensure the back-end is in place, allowing to run the study, and will then provide the UI component to configure a study to the client team.

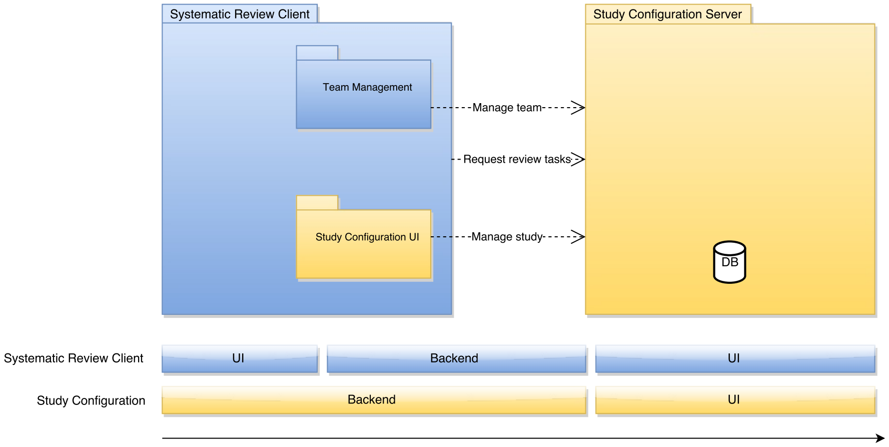

Figure 2: High-level presentation of the system architecture (UML inspired package diagram).

5 APIs

In order for the Systematic Review Client (blue team) to be able to interface with the Study Configuration Server (yellow team), we define the following interface between the two.

Both the client and server are aware about users, teams, studies, and tasks, each uniquely identified by an ID.

The common understanding of a study is it is comprised of:

- several stages, which are executed in order.
- during each stage, users work on a set of tasks.

The intricate details of which tasks are handed out to which user, as part of which stage, are left to the server, and does not involve the client. However, a task is defined by:

- unique ID;
- a set of visible data fields shown as part of the task, which cannot be modified;
- a set of requested data fields shown as part of the task, which can be modified;
- a type: either (1) a request to fill out data field(s), (2) a request to handle conflicting data field(s), as filled out by separate users.

Data fields are defined by:

- the type of data field, either:
-
  - String;
  - Boolean;
  - Enumeration (select one out of a list of predefined values);
  - Flags (select multiple or none, out of a list of values already specified);
  - Resource (e.g., PDF file).
- its value. For resources: ID, size, and file type.

The Study Configuration server exposes the following operations:

- CRUD user/team operations. (CreateTeam, AddUserToTeam, . . .):
  allow retrieving, creating, removing, updating, and deleting users, teams, and their associations. A user can be part of multiple teams. Teams and users have a name.
- `GetStudies(userID):`
  retrieves all the studyIDs a user belongs to.
- `GetTasks(studyID, userID, count, type=any):`
  retrieves a certain amount of tasks which need to be performed by the specified user for the specified study. Tasks are returned in the order in which they are recommended to be performed.
- `GetStudyOverview(studyID):`
  reports the different stages in the study, and per stage, for each user the

amount of tasks done, out of all the known tasks to be done.

- `DeliverTask(studyID, userID, taskID, modifiedFields):`
  deliver a finished task, including the resulting modified fields. This should return whether or not the task was delivered (e.g., can still be delivered) successfully. This can be called several times for the same task, in which case the latest value is used (if the task is still editable, which is decided by the server).
- `GetResource(id):`
  retrieves a specified resource (e.g., PDF file).
- `GetReviewableTaskIDs(userID, studyID):`
  retrieves all task IDs of tasks which have already been delivered, and can still be edited. This can be used to clear the cache of tasks which can still be edited within the client.
- `GetReviewableTasks(userID, studyID):`
  retrieves all tasks which have already been delivered, and can still be edited. This allows editing already delivered tasks.

This interface is the bare minimum which needs to be provided to support basic SMS and SLR studies. A specific WebAPI interface for the server, adhering to this specification, will be provided later in the project.
To set up the study, the server will need to expose additional interfaces, which can be freely decided by the Study Configuration Server (yellow) team, since the Systematic Review Client (blue) team does not rely on them. Within the user interface, internal communication between the main client, and the user interface to specify the study, relies on passing user, team, and study IDs.

6 Glossary

SMS: systematic mapping study. A type of secondary study aimed at providing a wide overview of a research area, at establishing if research evidence exists on a topic and an indication of the quantity of the evidence [3].

SLR: systematic literature review. A type of secondary study aimed at precisely answering one or more focused research questions [3].

References

[1] D. Budgen, A. J. Burn, O. Brereton, B. A. Kitchenham, and R. Pretorius. Empirical evidence about the UML: a systematic literature review. *Software: Practice and Experience*, 41(4):363–392, 2011.

[2] A. Fernandez, E. Insfran, and S. Abrahão. Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, 53(8):789–817, 2011.

[3] B. A. Kitchenham. Guidelines for performing systematic literature reviews in software engineering (Version 2.3). Technical report, 2007.

[4] B. A. Kitchenham, T. Dybå, and M. Jorgensen. Evidence-Based Software Engineering. In *Proceedings of the 26th International Conference on Software Engineering*, pages 273–281, Washington, DC, USA, 2004. IEEE Computer Society.

[5] J. Portillo-Rodriguez, A. Vizcaíno, M. Piattini, and S. Beecham. Tools used in Global Software Engineering: A systematic mapping review. *Information and Software Technology*, 54(7):663–685, 2012.

[6] M. Turner, B. Kitchenham, P. Brereton, S. Charters, and D. Budgen. Does the technology acceptance model predict actual use? A systematic literature review. *Information and Software Technology*, 52(5):463–479, 2010.