SLRTOOL: A Tool To Support Collaborative Systematic Literature Reviews

Abstract—*Background:* **Systematic Literature Reviews (SLRs) are used in a number of fields to produce unbiased accounts of specific research topics. The SLR process is particularly well documented and regulated in the medical field, where it is accepted as the standard mechanism to assess, for instance, the benefits of drugs and treatments. SLRs and meta-analysis techniques are increasingly being used in other fields as well, from Social Sciences to Software Engineering.**

Aim: **This paper presents the SLRTool tool - an open source, web-based, multi-user tool that supports the SLR process for a range of research areas. The tool is available at `http://www.slrtool.org` and is developed using a model-driven approach to enable its adaptation to different disciplines.**

Method: **The functionality of the tool is presented in the context of support for various phases of the SLR process. The use of the tool is illustrated by means of a simulated SLR aiming to map out existing research in the domain of Enterprise Architecture (EA).**

Results: **Details of the SLRTool tool are provided. The tool and its source code are available under an open source license. Commentary on the use of the tool and potential additional benefits are discussed, for example, the role of the tool in non-medical meta-studies.**

Conclusion: **The SLRTool tool supports all phases of the SLR process and lends itself to creating and supporting research communities geared to SLR oriented activities. In particular, the tool could be suitable for the novice researcher community.**

I. Introduction

The systematic literature review (SLR) is an increasingly important research strategy for extracting new knowledge from existing research data. The SLR process originated from the medical field but is now common across many disciplines, including Social Sciences and Computer Science. The SLR process is a well defined, methodical approach for identifying, assessing and evaluating existing published studies to support the exploration of one or more specific research questions. In the medical field, the Cochrane Collaboration[1] is a reference organisation that *"produces and disseminates systematic reviews of healthcare interventions"*. Cochrane reviews follow the guidelines provided in the Cochrane handbook. According to the handbook, "*a systematic review attempts to collate all empirical evidence that fits pre-specified eligibility criteria in order to answer a specific research question. It uses explicit, systematic methods that are selected with a view to minimizing bias, thus providing more reliable findings from which conclusions can be drawn and decisions made*".

The leading effort to formalize the SLR process has been carried out by Kitchenham et al for Software Engineering [**?**]. In their work they propose a set of guidelines which are obtained by synthesising the guidelines provided in other fields, including the Cochrane Handbook.

In Software Engineering and Information Systems, SLRs have been published on a range of topics including: open source software development [**?**], classifying requirements specification errors [**?**], motivations of developers [**?**] amongst others. The general usage of SLRs to investigate evidence base Software Engineering led to a meta study where Kitchenham et al assessed the impact of SLRs on SE practice [**?**]. Findings from this study suggest that the spread of topics covered by SLRs up to 2009 were fairly limited and main-stream Software Engineering topics were not well represented. Such areas could benefit from mapping studies of domains in the manner of Jorgensen and Shepherd [**?**].

SLRs are different from expert-led ad-hoc literature reviews by providing definitions and supporting documentation for review protocols prior to conducting the review such as: research questions; descriptions of the search strategy; descriptions of criteria to be used to assess each published study and procedures to be used to perform the review [**?**].

Experiences of SLRs reported in the SE domain suggest that in general, the SLR process is very resource intensive with particular problems associated with data extraction, management of large data and collaborative activities associated with conducting SLRs, with reviewers playing multiple roles during the SLR process. Further, the relative newness of the use of SLRs in the SE domain indicates that there also issues in provision of appropriate support for novice reviewers. As a consequence, it becomes essential that the SLR process can be easily validated and repeated, if research outputs from the SLR process are to carry an appropriate level of confidence. Validation and repeatability are guaranteed by the compliance with a certain protocol, which is clearly stated in medical SLR following the Cochrane method. A further requirement supporting the need for validation is the ability to demonstrate traceability between decisions.

A. Contribution of this paper

Currently, there is little evidence of using bespoke specialist tools to support SLRs in Software Engineering with respect to protocol definition, data collection and analysis, hence the opportunities for performing meta-level SLRs are consequently also limited. This paper proposes requirements for tools to support the SLR process based on existing SLRs performed within the SE domain and provides an assessment of existing options available. The main contribution is the introduction of the SLRtool that is specifically aimed at providing support for a collaborative SLR whilst addressing many of the key challenges currently prevalent around the SLR process. We suggest

[1] http://www.cochrane.org

that provision of tool support is paramount for embedding a particular practice such as SLR in the community.

The remainder of the paper is structured as follows: In the next section we describe the key challenges associated with the SLR process. Section 3 presents related work on building tools for supporting the SLR process and proposes a set of requirements for SLR tools across a range of possibilities. Section 4 provides a description of the key functionality of the SLRTool and illustrate its use for a small example mapping out the literature for Enterprise Architecture. In Section 5, the conclusion is presented along with remarks about the directions of future work.

II. CHALLENGES OF THE SLR PROCESS

One of the consequences of a growing number of SLRs conducted in Software Engineering and elsewhere is the need for consistency. This has parallels in the advent of methods engineering research in the software technology community. Following the development and use of methods such as Information Engineering [?], it was the advent of tool support (like the IEF) that finally embedded the use of such methods [?]. Other examples include the Rational Unified Process and its supporting tools such as Rational Rose. In 2007, Brereton et al. reported on lessons learnt from applying the SLR process within the Software Engineering domain [?], which can be consolidated into a set of requirements such as: support continuous revision of research questions and to keep track of changes; support multiple reviewers for the process and to track their usage; ability to tailor and re-configure the methodology for a particular review; support external validation of the review process; support multiple search strategies as different databases such as ACM and IEEE use different approaches; and keep detailed record of decisions. Elsewhere Riaz et al report on how formulation of research questions, quality assessment and the time consuming nature [?] of the SLR process create challenges [?]. The same study also commented on experiences of novice researchers who also struggled with formulation of research questions, with carrying out the reviews, and in particular with the time factors involved in carrying out the reviews and extracting data. Others have reported on the need to manage how specific roles are allocated and the activities associated with roles, for example the need to cross-check and validate criteria such as that for inclusion / exclusion or quality [?].

Staples and Niazi [?] reported on their experience of conducting an SLR using Kitchenham's guidelines. Of particular interest is their analysis in the potential for automation and support for SLRs. They used and benefitted from only a basic level of automation: simple tabular wordprocessing, simple spreadsheets and use of statistical analysis packages. Whist they viewed the potential prospects for tool support as *"very dim ...to support all phases of arbitrary systematic reviews"* they acknowledge that such mechanisms may provide "a nexus for the improvement of systematic review methodology" and contribute to a notion of a central index of systematic reviews for software engineering. This last point is particularly pertinent in the context of being able to perform meta-level systematic reviews accross sub-domains within software engineering.

While not directly concerned with the SLR process itself, the UBIRD study by Stelmaszewska et al [?] investigated the information seeking skills of students in the business domain. Findings from the study reported that students: used personal / social networks to refine search strategies; experienced significant difficulty in storing information collated from the searching activities (time being one such factor); and struggled with user interfaces of different digital repositories. The latter is consistent with Brereton et als' comment on the differences between the ACM and IEEE approaches.

Summarising, the current experience of using the SLR process suggests strongly that there is need for tool support in: configurable protocol definition while executing the SLR process by multiple stakeholders; reducing the time spent on managing the activities in carrying out the reviews and extracting the data for storage for subsequent review and analysis.

III. RELATED WORK IN TOOL BUILDING

Despite the increasing importance of SLRs to the Software Engineering and to other communities, there has been remarkably scant effort made to develop supporting tools for the SLR process.

Several tools have been built: for instance, the SM-VTM tool provides a text mining based approach to visualising the SLR clustering activities but does not provide support for an end to end SLR process [?]. The SLR tool developed by Fernandez-Saez et al. [?] focuses on the SLR process but some of the technical hurdles associated with the tool (such as installation and pre-configuration requirements, single user, and desktop-only version) limit its use [?]. Further, there has been no other evidence of its subsequent development. The StART tool reported by Hernandes et al [?] presents less of these technical limitations but remains limited to the Windows platform and does not readily lend itself to multiple users participating in the SLR process. Most closely related to the SLRTool is SLurP, [?] which has a functionality similar to our SLRTool, provides multiple user access and is a web based system implemented on top of the Linux distribution Debian and developed using Java (while we build upon the standard chain Apache+PHP+Mysql). The key difference is that our tool is fully open-source and implements a model-driven approach, in order to be adapted to a range of disciplines.

The medical domain, by virtue of the Cochrane organisation, has tool support available from *RevMan*[2]. Even in the case of RevMan, the tool only provides a stand-alone text editor enriched with functionalities that guarantee that all the steps in the review protocol are followed. Crucially, there is no support for collaborative work.

Related to tools supporting the SLR process are those which are essentially bibliography management systems. Some tools such as EndNote, Procite and Reference Manager focus on managing bibliographic databases and citation of references from such databases in the context of document preparation using desktop based applications. Other such as Zotero and Mendeley provide web based capability and in addition provide

[2] http://ims.cochrane.org/revman

some level of Web 2.0 collaborative activities including, for example, the sharing of bibliographic data between users. However, the key areas of collaboration, traceability of decisions, repeatability and ease of collection proposed by the functions of the tool described here are not collectively available in these technologies. Thus, generally the key weakness of these types of tools is the lack of support for the actual SLR process and there is only the possibility of manual record keeping of SLR data.

Both the lessons learnt from conducting SLRs as detailed in the Challenges section and the review of current tools leads to the following candidate set of requirements. Stage describes a grouping of activities associated a particular phase within the Kitchenham SLR process. Stages can be Planning, Conducting and Analysing an SLR.

Level is a much more subjective treatment. Currently, it is clear that SLRs can be supported by just the definition of a process, or the definition can be augmented by the use of simple spreadsheets at one end. At the other end of the spectrum, we can envisage a system that can provide automatic summarisation, natural language parsing techniques, full configuration management, automated complex statistical analysis and other such functions to support meta-level SLR process activities. At an intermediate level there are range of functions such as efficient mechanisms for downloading and managing full text, BibTeX data and support for multiple reviewers and their reconciliation.

Table 1 provides an overview summary of requirements classified according to their level. These requirements are derived partly from the literature describing experiences of SLRs and partly from our efforts at developing tool support based on our experience. When making the assessments we have used the published descriptions of the tools. Where it is unclear if a particular requirement is supported, we have indicated that with a “?”.

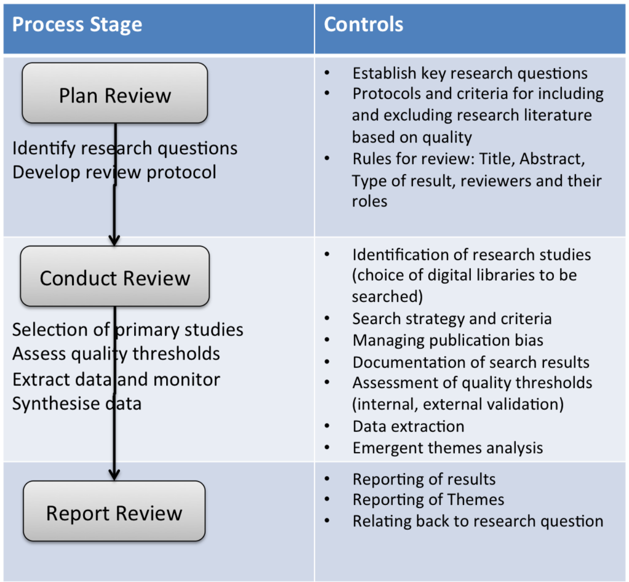
Based on these requirements, this paper introduces a software tool (SLRTool), with an open source implementation for configuring SLR processes and storing and managing data from an SLR process. The tool supports multiple users and is available in four guises: a generic web based version available at http://slrtool.org; a version specifically for reseachers at Middlesex University; a version for installation on local servers and an experimental version for the Raspberry Pi computer.

In designing the software, we have deliberately taken a model based language engineering approach in order to capitalise on potential meta analysis possibilities, as described below.

IV. SLRTool Functionality

The SLRTool is designed to support SLRs conducted using the guidelines described by Kitchenham et al. [?], [?], but is open to extensions, as described in Section IV-E. There are three key stages at the core of the guidelines: Plan Review; Conduct Review and Report Review. These are shown in Figure 1, along with some of the key activities within each stage.

In the following sections we introduce an illustrative case study of a SLR for Enterprise Architecture. The case study services to provide a demonstration of how the SLRTool tool supports the SLR process and some of the salient requirements.



Process Stage	Controls
Plan Review Identify research questions Develop review protocol	• Establish key research questions • Protocols and criteria for including and excluding research literature based on quality • Rules for review: Title, Abstract, Type of result, reviewers and their roles
Conduct Review Selection of primary studies Assess quality thresholds Extract data and monitor Synthesise data	• Identification of research studies (choice of digital libraries to be searched) • Search strategy and criteria • Managing publication bias • Documentation of search results • Assessment of quality thresholds (internal, external validation) • Data extraction • Emergent themes analysis
Report Review	• Reporting of results • Reporting of Themes • Relating back to research question

Figure 1. The SLR process

A. *Case Study*

We first introduce our SLR case study on Enterprise Architecture (EA). EA is the means by which the essentials of a business, its IT and its evolution, and analysis is performed. It is a coherent whole of principles, methods, and models that are used in the design and realisation of an enterprise’s organisational structure, business processes, information systems and infrastructure. To date, there has been no SLR of this topic. In particular, the type of mapping study in the manner of Jorgensen and Shepherd [?] would be particularly beneficial. Thus, the SLR study has the following research questions:

1) What are the main research topics being addressed by EA research?
2) What are the main methodological approaches being used in EA research?
3) What are the main publication outlets for research in EA?

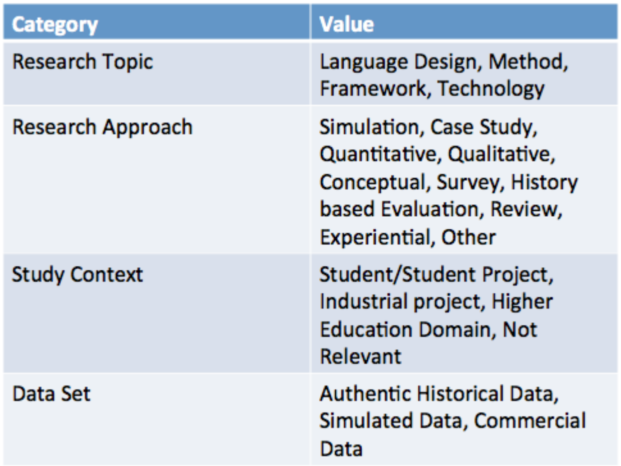
As the study is intended to focus on prevalent research activity in EA, resources such as literature opinion papers, panel discussions, commercial white papers are to be excluded. Categorization of resources is however, particularly important as the study aims to address the main research areas and the approaches taken, hence categories such as: Research Topic (values are: Language design, Method, Framework, Technology), Research Approach (values are: Simulation, Conceptual, Survey, Experiential and others) are needed. The full framework is shown in Figure 2. We can see how how research outputs can be classified using these categories to help contribute towards answering research questions 1 and 2.

B. *Plan Review Functionality*

The Plan Review stage aims to identify research questions (examples of research questions are listed in the previous

Requirement	Stage	Level	MS-Office	SM-VTM	SLR-tool (Fernandes et al)	StART	SLURP	SLRTool (Barn et al)
Ability to define Research Questions, Inclusion/Exclusion Criteria, Categories for refining research questions and the values such properties can take.	Plan Review	Advanced	Y	?	Y	Y	Y	Y
Ability to define dependencies and other relationships between criteria and categories.	Plan Review	Advanced	N	?	N	N	N	Planned
Ability to define search criteria independent of target resource database.	Plan Review	Intermediate	N	?	N	?	N	Y
Use of ontologies to provide a structured language for defining categories, criterias and their values.	Plan Review	Advanced	N	N	N	N	N	N
Ability to define roles to multiple users.	Plan Review	Intermediate	N	?	?	N	Y	Y
Automatic extraction of BiBTeX data of all resources located.	Conduct Review	Intermediate	N	N	N	Y (no export)	Y	Y
Automatic download and management of fulltext resources where licence permissions allow.	Conduct Review	Intermediate	N	N	N	N	?	Y
Ability to support multiple reviewers reporting on resources.	Conduct Review	Intermediate	N	N	Y	N	Y	Y
Difference analysis and Reconciliation services between users including traceability between decisions.	Conduct Review	Intermediate	N	N	N	N	Y	Planned
Ability to support summarisation and keyword extraction for automated classification.	Conduct Review	Advanced	Y	Y (partial)	N	Y	N	N
Ability to link classification decisions to specific sections in the resource documents.	Report Review	Advanced	N	N	N	N	N	N
Ability to support canned statistical analysis - such as cross tabs for criteria.	Report Review	Intermediate	Y (Excel)	?	N	N	N	Planned
Provision of descriptive statistics reporting tools.	Report Review	Intermediate	Y	Not Clear	?	N	Y	Y
Source Code available and extensible	All	Advanced	N	N	Y	?	N	Y
Multi-platform support	All	Advanced	Y	?	?	N	N	Y

Table I. SLR Requirements and current coverage



Category	Value
Research Topic	Language Design, Method, Framework, Technology
Research Approach	Simulation, Case Study, Quantitative, Qualitative, Conceptual, Survey, History based Evaluation, Review, Experiential, Other
Study Context	Student/Student Project, Industrial project, Higher Education Domain, Not Relevant
Data Set	Authentic Historical Data, Simulated Data, Commercial Data

Figure 2. Classification Framework for the Enterprise Architecture SLR project

section) and develop the review protocols such as definition of quality criteria, inclusion / exclusion of resources, reviewers and their roles and the definition of a classification framework. The SLRTool defines a review as a project and provides various dialogues for defining research questions, and the categories of resources that contribute to the classification framework, see Figure 3. The details of the protocol may differ in different disciplines: Section IV-E below describes how to accomodate this variability in the SLRTool. Repeating a particular SLR is then a matter of importing a particular protocol and reusing it. Currently the following requirements are relevant:

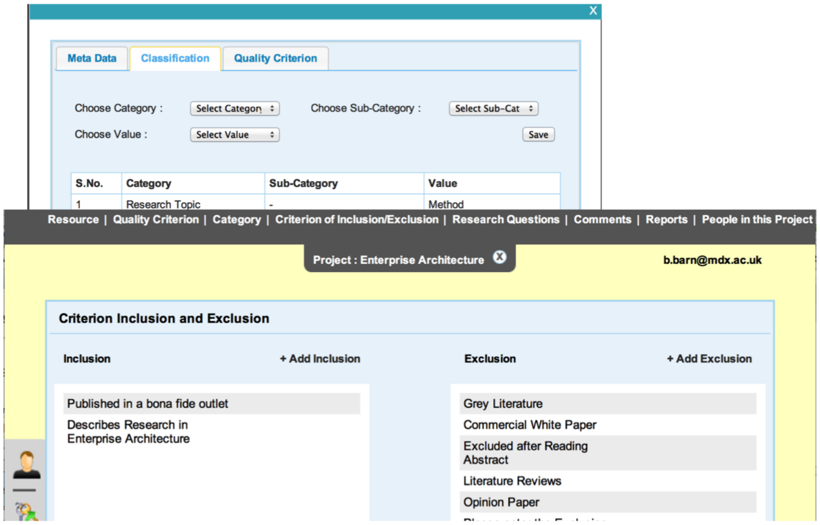
1) Ability to define Research Questions, Inclusion/Exclusion Criteria, Categories for refining research questions and the values such properties can take.
2) Ability to define dependencies and other relationships between criteria and categories.
3) Use of ontologies to provide a structured language for defining categories, criterias and their values.
4) Ability to define roles to multiple users.

Tool capability that can provide a structured language for describing the various categories, criteria and values in the form of an ontology is an advanced capability that is curently planned for the SLR Tool but is not yet available.

C. Conduct Review

This is the most time consuming section of the SLR process. Activities include: defining the sources of data (the digital resources, journals etc.); the search strategies; data extraction; assessing the quality of resources; performing classifications based on the reading of the papers (resources) and of course storing and managing the data. The SLRTool provides tool functionality for these activities which are optimised for reducing the time resource spent on this. Some of the key functions (shown in figure 3.) include:

1) Ability to define search criteria independent of target resource database with interfaces to bibliographics support tools such as Google Scholar.
2) Automatic extraction of BiBTex data of all resources located, along with automatic download of full text PDFs subject to permissions for the host institution.



S.No.	Category	Sub-Category	Value
1	Research Topic	-	Method

Criterion Inclusion and Exclusion

Inclusion

- Published in a bona fide outlet
- Describes Research in Enterprise Architecture

Exclusion

- Grey Literature
- Commercial White Paper
- Excluded after Reading Abstract
- Literature Reviews
- Opinion Paper

Figure 3. Classification of resources and Inclusion/Exclusion criteria

These are done as batch downloads on selected resources.

3) Ability to include / exclude resources and to record reasons based on previous defined criteria.
4) Ability to categorise resources according to the review protocols defined earlier.
5) Support for multiple reviewers on a SLR project.
6) Ability for reviewers to comment on decisions made and to assess reviewer actions.

We have planned development activity to increase support for this phase of the SLR process to include requirements for delivering further collaborative actions such as:

1) Difference analysis and Reconciliation services between users including traceability between decisions.
2) Ability to support summarisation and keyword extraction for automated classification.

One of the features of the SLRTool is the facility to experiment with search strings and results from databases before defining the SLR protocols in detail: this is exemplified in the screenshot of Figure 4.

D. Report Review

This stage of the SLR process includes both descriptive statistics and more detailed analysis linking back to the themes and research questions intended to be explored in the review. The SLRTool provides functions to help in these activities as follows:

1) Report summaries of publisher details;
2) Report summaries of publication dates;
3) Reporting on Inclusion and Quality criteria;
4) Exporting of data to external systems.

As the tool is still being developed, other reports planned include reporting on the classification framework and the relationships linked to the research questions. Other scheduled activity includes the ability to link classification decisions to specific sections in the resource documentsband provision of canned analysis techniques.

Reports from the case study used in this paper illustrate the ease by which these can be produced: Figure 5 illustrates the distribution of publishers and the distribution of quality criteria for the papers identified in our study.

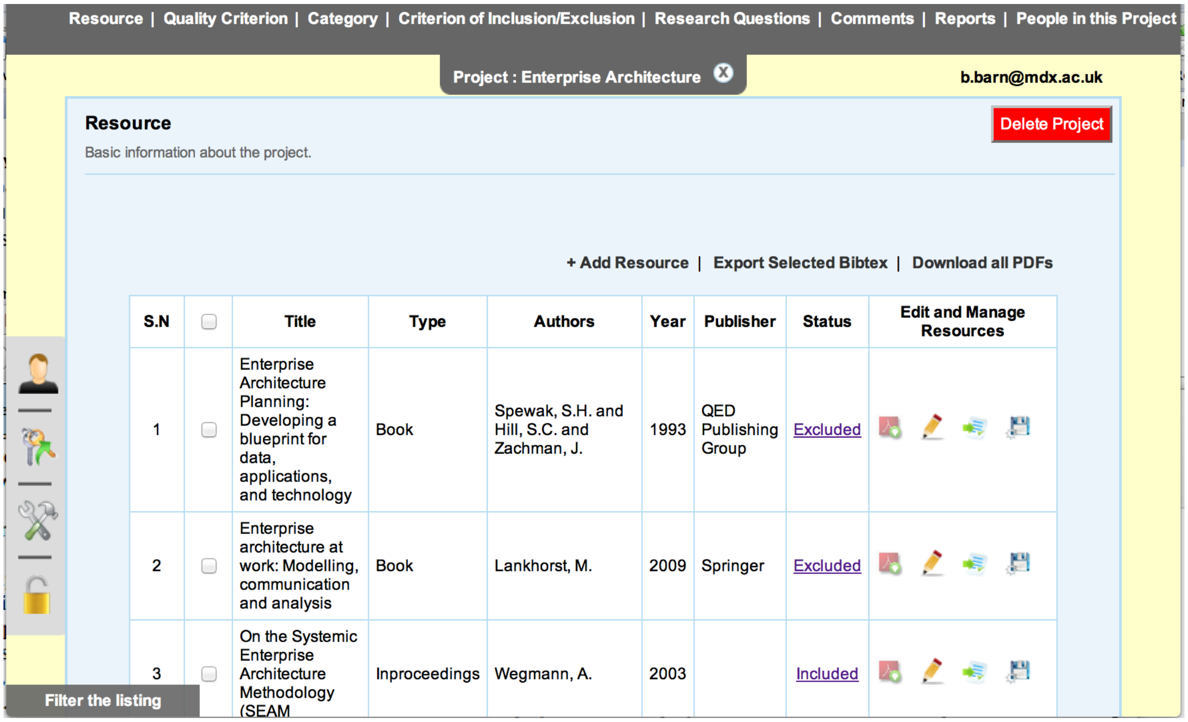
E. Technical aspects of the tool

In our analysis of existing technologies in this area, all the existing tools had some specific technical limitations around platform dependence and lack of multi-user access. This was influential in the approach that we took towards the selection of the technical architecture of the tool.

The SLRTool uses a standard open source web development platform comprising the Apache web server, MySQL and PHP. We have successfully installed it on a number of operating systems, including Linux, Mac, Windows, and even on a Raspberry Pi: the latter installation has the benefit of being mobile so that the *whole* set of repositories can be moved between institutions, if necessary.

Another key element in our approach was an early realisation that the potential for substantial automated analysis of reviews could be made possible if we took a model based approach to tool development. Thus we see the concepts underpinning the tool as an example of a domain specific language. In our project we have first of all defined a model for SLR. This is summarised in the figures reported below. Intuitively, as described in Figure 6, each Resource (a journal article, a study, etc.) has an associated Source (usually a database such as Google Scholar or Scopus). Resources can be a single paper or a collection of various other resources. A paper has a structure, which usually includes a Title, a set of authors Author, and a set of Sections. Each resource, in turn, is associated with a set of Properties that can be numeric, a range, Boolean values, etc.

The overall SLR process is defined in a similar way, and all the code is generated from this model. As a result, we are able to accomodate changes in the protocol and in other parts of the SLR process just by modifying the model, and then generating (and re-using, where available) code to support different disciplines.

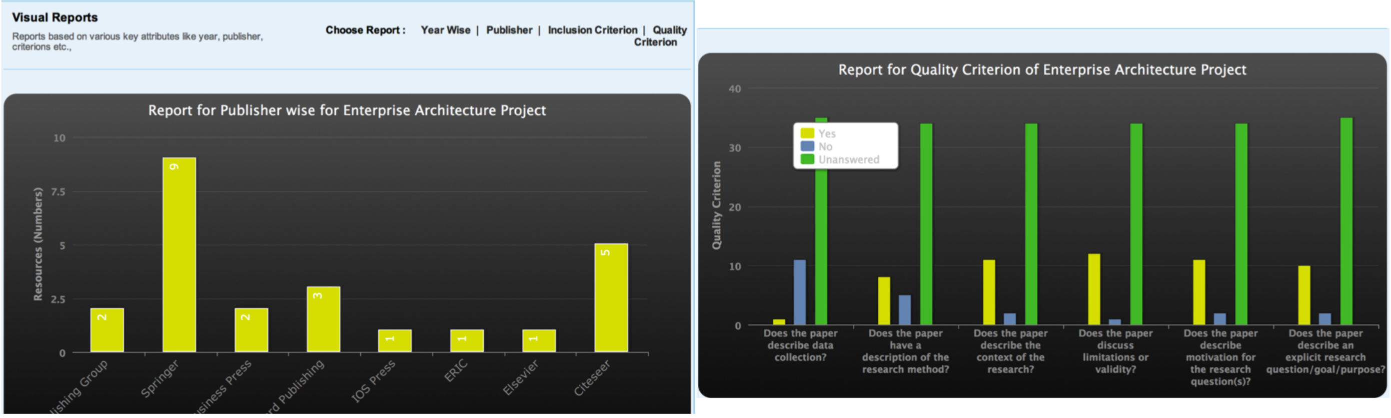

Resource | Quality Criterion | Category | Criterion of Inclusion/Exclusion | Research Questions | Comments | Reports | People in this Project
Project : Enterprise Architecture
b.barn@mdx.ac.uk
Delete Project
+ Add Resource | Export Selected Bibtex | Download all PDFs
Filter the listing

Resource

Basic information about the project.

S.N	□	Title	Type	Authors	Year	Publisher	Status	Edit and Manage Resources
1	□	Enterprise Architecture Planning: Developing a blueprint for data, applications, and technology	Book	Spewak, S.H. and Hill, S.C. and Zachman, J.	1993	QED Publishing Group	Excluded	
2	□	Enterprise architecture at work: Modelling, communication and analysis	Book	Lankhorst, M.	2009	Springer	Excluded	
3	□	On the Systemic Enterprise Architecture Methodology (SEAM	Inproceedings	Wegmann, A.	2003		Included	

Figure 4. Resources



Visual Reports

Reports based on various key attributes like year, publisher, criterions etc.,

Figure 5. Reporting

F. *Current Limitations*

We consider the tool as a prototype: the version available on-line implements a connector with Google Scholar, but we are planning to include other databases (subject to institutional agreements etc.). Google Scholar was given a dominant role as the study on information seeking behaviours of students provided empirical evidence of its widespread use in HEIs [**?**]. The current operation of the use of the connector to Google Scholar is available on the opening project screen. Users are able to search for a variety of search parameters and note results arising from the searches. By limiting searching to just Google Scholar, we recognise that papers will be missed because of issues such as how different digital libraries configure their searching facilities. One side effect though, is that it does provide a consistency in the protocols across SLRs. Examples of searching are shown in Figure 7.

We envisage SLR tool development activity to continue to progress along the spectrum we outlined earlier where there is increasing sophisticated support for the activities that we consider to be key: Management of resources (especially auto-upload); Recording evidence (the classification of resources and linking them to specific evidence within a resource) and Collaboration (how better to support team and inter-team working). These activities are essential if there is to be standardisation across SLR cases.

Other areas of the tool that we intend to develop further include more reporting options of the data collected, in particular the classification frameworks used. An embedded help guide can be particularly useful to novice researchers; this help feature can direct users towards the correct use of features of the tool in the context of the SLR process. We are currently developing these on-line help facilities.

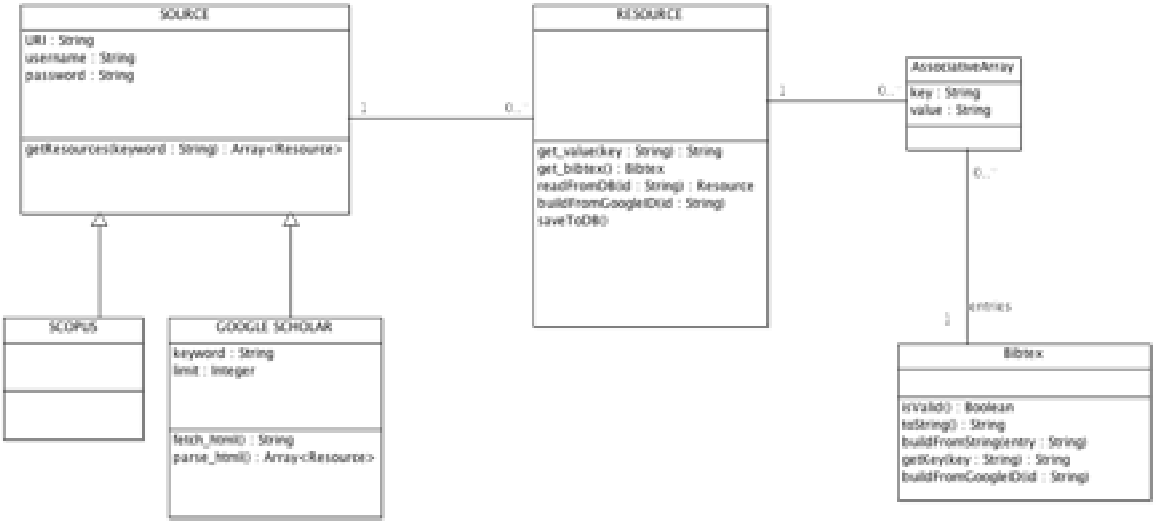


Figure 6. Conceptual model for the SLRtool.

Staples and Niazi in their report on their experiences of performing SLRs made specific reference to the notion that software tools can provide a nexus for the improvement of systematic review methodology. We have interpreted this beyond their original intention and propose that model driven engineering practice may provide valuable inroads in developing functionality that can specifically address functions that allow configurability of SLR processes. By designing the SLRtool using model driven principles we are aiming to support specification of an SLR process for a particular domain such that the specification can derive an instance of the tool for use for that domain.

Finally, we recognise that the tool has only been used in a small scale, non-independent manner. It is our intention to carry out evaluation studies with the tool both to refine the tool and to examine how the SLR process is possibly modified as a result of using tools.

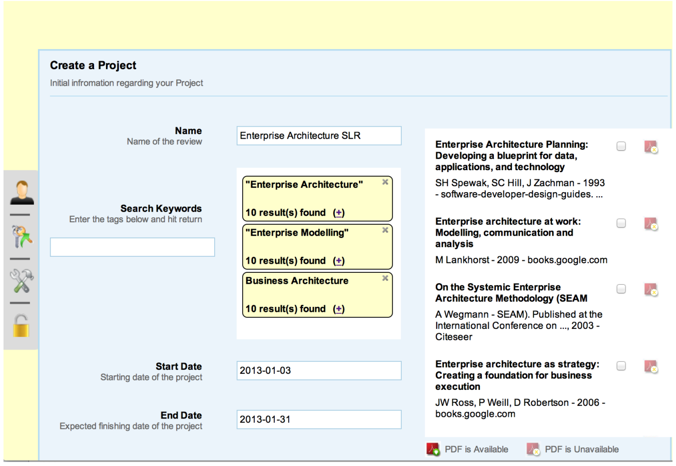
V. Final Remarks

The current literature indicates that a well developed SLR that provides reliable outputs is a difficult and time consuming task, mainly because of the management and administration of large complex volumes of data. When this is coupled with multiple users as part of a team activity then the problem increases. In this paper we have sought to describe the development of the SLRTool, which aims to address these issues and to enhance the capabilities required in performing an SLR.

While we have been motivated by our our experience as researchers struggling with bottlenecks and inefficiencies in attempting to carry out SLRs, we think that expert reviews typically found in research papers in general could also benefit from such a tool. Given this, we anticipate building a community around our approach, and future versions of this tool will have an appeal to many researchers in a range of research areas. We have noted that Brereton et al reported that the Simula Lab was the leading research lab on conducting SLRs because they had developed very effective research procedures which were embedded in the organisation. Such procedures require tool support and the provision of the SLRTool and its usage in the research community could help developing capability similar to the Simula Lab elsewhere. In particular we are keen to embed the tool as part of the research training undertaken by doctoral students at our institution and elsewhere.

It is likely that SLRs will become more pervasive and be relevant to most (if not all) disciplines, hence software tools that can support the SLR process will be a necessary requirement. Further, as different disciplines approach the SLR process definition differently (the protocols and activities for example), an important challenge that arises is the need to support different flavours of SLR methodologies. The project reported in this paper has concentrated on a particular style of SLR methodology; however, the fundamental design principles that underpin the software are based on model driven software engineering practice, and future versions of the software will be able to support user configurable SLR methodologies. Related to this, the setting up of an SLR protocol, the criteria for inclusion/exclusion, classifiers and so on can be re-used across specific sub-domains, leading to systematic meta studies of SLRs. For example, it is not inconceivable that SLRs aimed at mapping out sub-domains such as Component Based Design, Service-Oriented Architecture, Event driven Architecture (all broadly related to software architecture) could utilise the same SLR protocol enabling a much richer meta-analysis of the set of SLRs. We are progressing further development of the SLRTool to address these proposals and the limitations and requirements discussed earlier.

The model driven approach we have deployed in the tool may introduce the possibility for new insights to be generated from the act of performing the SLR process. For example, implicit relationships between classification frameworks and quality criteria may become more apparent. In the case study used to illustrate this paper, we observed that because of the explicit SLR processes that are supported by the tool. Conversely, analysis that exposes inconsistencies both in the primary data and in the data arising from the analysis undergone as part



Create a Project

Initial infromation regarding your Project

Name
Name of the review
Enterprise Architecture SLR

Search Keywords
Enter the tags below and hit return

"Enterprise Architecture" ×
10 result(s) found (+)

"Enterprise Modelling" ×
10 result(s) found (+)

Business Architecture ×
10 result(s) found (+)

Start Date
Starting date of the project
2013-01-03

End Date
Expected finishing date of the project
2013-01-31

Enterprise Architecture Planning: Developing a blueprint for data, applications, and technology □
SH Spewak, SC Hill, J Zachman - 1993 - software-developer-design-guides. ...

Enterprise architecture at work: Modelling, communication and analysis □
M Lankhorst - 2009 - books.google.com

On the Systemic Enterprise Architecture Methodology (SEAM □
A Wegmann - SEAM). Published at the International Conference on ..., 2003 - Citeseer

Enterprise architecture as strategy: Creating a foundation for business execution □
JW Ross, P Weill, D Robertson - 2006 - books.google.com

PDF is Available PDF is Unavailable

Figure 7. Searching Google Scholar

of the SLR process may also be possible. As we continue to work on the development of the tool, we anticipate that new syntheses of research literature will reveal additional meta-properties that could guide the production of more informed research output.

We note that one area of concern remains regardless of any tool implementation. There is a policy issue at stake. For a tool like this to be truly embedded in the research practice of academics and students (at all levels), the tool will need to be part of the armoury of utilities resourced by managers of institutional libraries. Ease of use of the tool, particularly access to full text resources, is dependent on permissions available at the home library. Often, access to digital resources is determined by various identity management systems such as Athens[3] and Shibboleth[4] in the UK. Such systems will need to be integrated the SLRTool tool for embedding to take place.

Acknowledgements: We thank the UK's Joint Information Systems Committee (JISC) which supported this research at Middlesex University.

[3] http://www.eduserv.org.uk/identity-access

[4] http://shibboleth.net