Replication Data Management: Needs and Solutions

An initial evaluation of conceptual approaches for integrating heterogeneous replication study data

Stefan Biffl, Estefanía Serral, Dietmar Winkler
Institute of Software Technology and Interactive Systems
CDL-Flex, Vienna University of Technology
Vienna, Austria
{first_name}.{last_name}@tuwien.ac.at

Oscar Dieste[1], Natalia Juristo[1, 2]
[1]Facultad de Informática
Universidad Politécnica de Madrid
Boadilla del Monte, Spain
{odieste, natalia}@fi.upm.es

Nelly Condori-Fernández
ProS Research Center, Universitat Politècnica de València
Valencia, Spain
nelly@pros.upv.es

[2]Department of Computer Science and Engineering
University of Oulu,
Oulu, Finland
natalia.juristo@oulu.fi

***Abstract* — [Context] Replication Data Management (RDM) aims at enabling the use of data collections from several iterations of an experiment. However, there are several major challenges to RDM from integrating data models and data from empirical study infrastructures that were not designed to cooperate, e.g., data model variation of local data sources. [Objective] In this paper we analyze RDM needs and evaluate conceptual RDM approaches to support replication researchers. [Method] We adapted the ATAM evaluation process to (a) analyze RDM use cases and needs of empirical replication study research groups and (b) compare three conceptual approaches to address these RDM needs: central data repositories with a fixed data model, heterogeneous local repositories, and an empirical ecosystem. [Results] While the central and local approaches have major issues that are hard to resolve in practice, the empirical ecosystem allows bridging current gaps in RDM from heterogeneous data sources. [Conclusions] The empirical ecosystem approach should be explored in diverse empirical environments.**

I. INTRODUCTION

In empirical software engineering, replication studies target several objectives. One of the most relevant is enabling data analysis across data collections of several iterations of an experiment, e.g., comparing the effectiveness of quality assurance methods [1] or agile techniques [2]. In practice, there are several researcher roles involved in planning, conducting, and analyzing replication experiments: (1) *Researchers focused on one experiment* and (2) *Researchers looking at several experiments*. *Researchers focused on one experiment* want to efficiently plan, conduct, analyze, and report selected aspects of this experiment, with consideration to previous related experiments. In this context there are roles, such as: (a) a research manager, in charge of a series of experiments; (b) an experiment manager, in charge of a specific experiment instance; (c) an operative senior experimenter, who is often supported by personnel and is in charge of practical aspects of the experiment; and (d) a meta analyst, in charge of reviewing the raw data and testing hypotheses. *Researchers looking at several experiments* want to efficiently collect and process data on aspects of several related experiments, typically from different research environments. For all researchers in the experiment, it is very important to agree on a data model to allow ensuring the correct measurement and analysis of the core experiment data during the experiment process and supporting queries on the status of the experiment as needed by each role. Replication Data Management (RDM) is concerned with providing both types of researchers with effective and efficient approaches on data model definition, integration, and querying.

Even though there are replication guidelines to support replication validity [3], these guidelines do not take into account the experiment in minute detail or prescribe the local representations of data model parts for RDM. As a consequence the local data models and storage solutions tend to vary widely and make the resolution of queries to replication experiment data (e.g., for experiment process quality management, for data mining or knowledge management), unnecessarily inefficient and error prone.

Key challenges for RDM come from integrating heterogeneous data models and data from empirical study infrastructures that were not designed to cooperate [4], e.g.: (a) data model variation of local data sources: experiment designs vary somewhat from previous experiments to look at new aspects, therefore, also the data model varies; (b) incomplete data semantics even if data models exist (e.g., in UML or EER): often the semantics of the data is agreed on only for the most basic data elements (e.g., role); or (c) the needs of researchers are not explicit. While there is some agreement on desirable results of empirical replication studies, there is, to our best knowledge, no consolidated set of requirements documented on schemas and functions for researchers to access, use, and share their data sets.

Based on initial discussions with replication researchers, we found as a common goal to use an effective and efficient infrastructure for conducting empirical studies. This infrastructure should provide replication researchers with a common data

[1] This research has received funding from the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria, from FiDiPro, Finland, and from TIN2011-23216, Spain.

model that collects data from heterogeneous data models and allows local researchers the freedom to work with their local data model. There have been several attempts to establish central data repositories in the empirical software engineering community, e.g., [5] [6]. However, researchers still tend to conduct local variations of experiments and not to follow a common data model [7] [8].

In this paper we analyze RDM needs and provide an initial evaluation of conceptual RDM approaches to support replication researchers. Following an adapted ATAM (Architecture Tradeoff Analysis Method) [9] evaluation process, we analyze: RDM stakeholders, use cases, example data models, and data operations; and elicit the key RDM needs of several empirical replication study research groups to derive criteria for evaluating RDM approaches. Then we evaluate three conceptual solution approaches to address these RDM needs: (a) *central data repositories* (CR) with a fixed data model, which are convenient for repository keepers and meta studies; (b) loosely connected heterogeneous *local repositories* (LR), which are convenient for experimenters and are in wide-spread use in the experimental software engineering community; and (c) an *empirical ecosystem* (EE) concept, similar to well-established software and systems engineering ecosystems [10], which promises to provide replication and meta study researchers with a common data model that collects data transparently from local heterogeneous data models, and also allows experimenters to manipulate their own local data models.

Major results of the initial and qualitative evaluation are: the approach for RDM comparison was found useful and usable by the researchers involved. While the central and local approaches have major issues that are hard to resolve in practice, the empirical ecosystem concept allows bridging current gaps in RDM and should be empirically explored. We report lessons learned and research issues for future evaluation for replication researchers and empirical data repository keepers.

This remainder of this paper is structured as follows: Section II provides an overview on empirical replication studies, replication experiment stakeholders, empirical repository approaches, and data integration approaches from heterogeneous data sources. Section III motivates the RDM research issues and approach. Section IV analyzes RDM needs, introduces a novel approach for RDM, and derives RDM evaluation criteria to evaluate the three conceptual RDM approaches. Section V discusses the evaluation results and Section VI concludes and suggests further research work.

II. Related Work

This section summarizes related work on empirical replication studies, repositories for empirical data, and data integration approaches from heterogeneous data sources.

A. Replication Studies and Data Management Needs

This section provides a brief summary of empirical study research and practice from three work groups in three different universities: UP Madrid, UP Valencia, and TU Vienna.

UP Madrid studies. There are three main active lines of empirical research at UP Madrid: software testing, human factors, and requirements elicitation. The *software testing* line belongs to the family of experiments initiated by the pioneering work of Basili and Selby [11]. In particular, the experiments executed at UP Madrid are based on Kamsties and Lott's replication package [12]. The research goal is identifying the effectiveness of structural and functional unit testing techniques when applied to various defect types. Around 15 replications have been carried out since 2001 within and outside UP Madrid. Several changes in the design of the original experiment (e.g., cross-over versus parallel design, different types of defects) have been exercised for several reasons, either theoretical or simply practical (e.g., adapting the experiment to available lecture hours). The *human factors research* started in 2004, triggered by a previous research about the relationship between people characteristics and their ability to perform certain software activities (e.g., analysis and/or design) [13]. The type of empirical study used was a quasi-experiment and, although it has been replicated five times in three different sites, the original design remained quite stable. The *research in requirements elicitation* started in 2007 with the aim of independently verifying some unusual effects (e.g., experienced analysts do not clearly beat novices) in requirements acquisition experiments [14]. Quasi-experiments were run until 2012, when controlled experiments began to be run. The basic design remained quite stable through 4 replications, but changes to the context variables and population characteristics have been frequent and considerable.

Several aggregations have been performed [15][16][17] but none can be regarded as conclusive. The main obstacle has been the differences among replications that prevented a cohesive analysis. Other sources of problems are the experimental data itself and the various mechanisms to collect and store the experimental data. Data sharing among UP Madrid researchers is possible, but consultation is invariably required to clarify the semantics of the data sets.

UP Valencia studies. Software measurement and quality in a model-driven context has been investigated through various experimental studies carried out at the Technical University of Valencia. Main goal of these replications was to iteratively adjust the effectiveness of mapping rules that were defined for measuring functional size of artifacts obtained at different phases of a model-driven development process (i.e., requirements specifications [18][19] and conceptual models [20][21]). Beyond getting an effective measurement method, results of these studies have been used to define a theoretical model for evaluating the acceptance of model-based measurement tools [22][23]. For the experiments' design, UP Valencia used the framework proposed by Wohlin *et al.* [24].

Although part of the instrumentation was more easily reused because they were stored in a common repository, researchers at UP Valencia experienced a higher effort for data analysis and aggregation regarding experimental data manually collected by each experimenter and stored on different local machines with different formats, schemas, and time periods.

TU Vienna studies. Software Inspection (SI), a static quality assurance approach for identifying defects in requirements specifications and source code, has been in focus of various studies at TU Vienna. Main goal of these studies was the investigation of effects of different reading techniques

(i.e., guidelines for systematically reading specific documents under inspection) on defect detection performance, e.g., effectiveness, efficiency, team effects, and decision support. Based on a large-scale initial empirical study [25] including 200+ participants, the study was replicated, evaluated, and published [7]. Biffl *et al.* reported on this family of experiments with focus on groupware effects for SI [26] and Halling *et al.* reported on defect detection effects of tool support [27]. Results of these studies have been used to identify best-practice inspections in context of requirements inspections as candidate improvements for supporting pair programming or testing [8]. TU Vienna applied the process, presented in Wohlin *et al.* [24], for every study and captured the individual experimental data in local repositories.

For evaluation purposes collected data had to be combined manually with considerable effort for data aggregation and analysis. Because of the high effort for data collection, analysis, and aggregation (if experiment data are collected in local data bases) there is a strong need (a) to provide tool support for linking individual experiments and experiment data and (b) to provide a platform for experiment and replication support.

B. Repositories for Empirical Data

A key issue in empirical research is providing related experiment data for analysis and aggregation purposes where stakeholders (a) provide related data and (b) use, analyze and aggregate data. Typically, main architecture elements of such research environments are data repositories for (a) individual and (b) several experiments, and (c) appropriate interfaces to provide data and support efficient and effective data management. Data management includes mapping and transformation between repositories, data validation, and query generation for analyzing (aggregated) experiment data.

Efficient data management of experimental data is crucial for all RDM stakeholders. While senior experimenters typically focus on planning, executing, and analyzing a small set of experiments within their own scope (relatively homogenous data), and experiment managers as well as research managers and meta analysts aim at (a) defining overall research goals involving several individual experiments and (b) aggregating data based on individual experiments. Aggregation and synthesis of data strongly depend on the capability to exchange data in a distributed and heterogeneous experiment environment. Knowledge engineers aim supporting data management, mapping and transformation based on semantic technologies.

Discussions with empirical researchers coming from three different research groups revealed two main approaches to handle data management in large-scale research experiments even across research groups: (a) central repositories and (b) heterogeneous local repositories. Literature mainly reports on central repositories to support empirical software engineering data analysis and aggregation.

Central repository (CR) concepts assume a unified data model. Individual experimenters provide their data via interfaces according to a pre-defined data model of the CR. Figure 1 shows a schematic overview on the steps in the central experiment data repository approach. Basically, research managers provide underlying research strategies and define goals (1). Knowledge engineers provide the CR data model, queries, and interfaces to related stakeholders (2). Based on these requirements individual research groups plan, execute and provide their (replication) experiment data to the CR (3a, 3b). In the last step research managers derive aggregated results (based on defined queries) from the CR data base (4). However, the introduction of new data sets that do not fit to the central data model is challenging and can hinder collaboration in empirical software engineering.

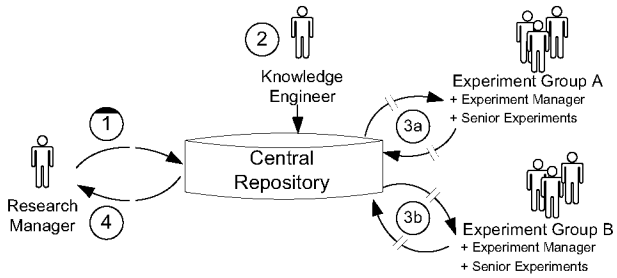

Figure 1: Central data repository with fixed data model.

There are reports on the following groups of CR approaches: (a) data/material repositories, (b) (replication) experiment management, and (c) knowledge and experience management.

Data/Material Repositories. The *PROMISE* [28] *data repository*[2] is an initiative to collect local and individual data from real-world software engineering projects where researchers aim to provide their local data for reusing purposes. Another example in this group is the *Software-Artifact Infrastructure Repository (SIR)* [29]. However, the data storage approaches used have limitations in data aggregation in heterogeneous experiment environments.

(Replication) Experiment Management. Several approaches address the need for managing experiments and replications. The *Framework for Improving the Replication of Experiments* (FIRE) [30] addresses knowledge sharing issues for internal as well as external replications to (a) improve knowledge transfer, (b) enable high-quality replications, and (c) provide more generalizable results. Further examples in the group are the *Coding Contest Data Repository*[3] [31], the *Software Engineering Technology Testbed (SETT)* [32], [33], and the *Simula Experiment Support Environment (SESE)* [34]. These approaches are suitable within one family of experiments but are typically limited to selected application domains and/or experiment settings.

Knowledge and Experience Management. The *Experience Factory* [35] approach defines a framework for experience management with focus software engineering best-practices including procedures and tools that support managing the experience base. However, the tool support for building and accessing the experience base seems to be initial. Further examples in this group are the *eSEE system* [36] and the *EXPerience Repository (EXPRE)* [37] knowledge management tool. Other repositories focus on reconciling several phenomenological software models in a common framework

[2] Promise: http://promisedata.googlecode.com
[3] Catalyst Coding Contests: http://www.catalysts.cc

(e.g., *CeBASE* [38]) or inter-organizational learning (e.g., *ViSEK* [39]) but did not show significant progress since their introduction.

Local repositories (LR) aim at providing data in a small scope (e.g., within one work group) and enable local analysis and aggregation. Although this is the typical approach that is followed for RDM, data exchange and aggregation across research groups often become difficult, time-consuming, and error-prone because of the semantic heterogeneity of local data repositories. Figure 2**¡Error! No se encuentra el origen de la referencia.** illustrates the approach of heterogeneous local data repositories. Research managers provide strategies and direction directly to the responsible experiment teams (1) who execute the study and collect data in LRs, e.g., structured text files, simple data bases, or scripts (2a, 2b). For the aggregation and data analysis of combined data (from various LRs) an effort-consuming aggregation (3) step is necessary. Usually this aggregation step is executed manually by Knowledge Engineers and/or Research Managers and therefore is error prone and risky. Finally, research managers can derive aggregation results from defined queries (4).

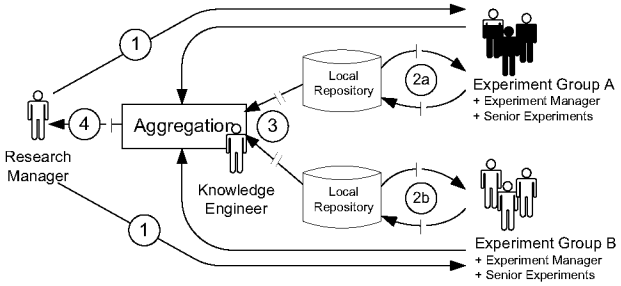

Figure 2: Aggregation of heterogeneous local repositories.

C. Data integration from heterogeneous data sources

The data models and instances are managed in most cases locally by the research groups in charge of the experiments. Thus, to allow running queries across different local data models, a key question is how to integrate the data models from heterogeneous local data sources. Data integration is formally defined as the solving of problems originating from the intent to share data across heterogeneous data sources [40]. The fundamental reason that makes heterogeneity hard to address is the independent origin of data sets using varying structures to represent the same or overlapping concepts [41].

According to Wiesner *et al.* [42] there are three approaches to achieve data integration in heterogeneous environment: (a) brute-force approach, (b) global standard approach, and (c) interchange standardization approach. The *brute-force approach* proposes to create a special software converter for each pair of models. Each converter translates the replication data from one model to other. This approach requires a lot of effort for coding and maintenance, which is time- consuming and costly. The *global standard approach* is based on the creation of a global standard: all experiment replicators should agree on a common format for the representation, processing, and storage of their data. Although the idea is good and such a standard would extremely facilitate managing the heterogeneous replication data, the feasibility of this approach in engineering practice is unlikely. One of the reasons is that usually empirical researchers use their own models and it is difficult to settle a simple one that is applied for all the researchers. Along with the technical aspects, the change of the standard will generally lead to a costly modernization or adjustment of their workflows and tools.

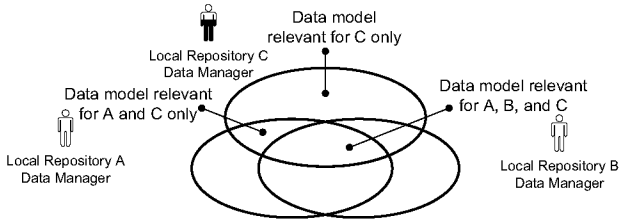

Figure 3: Interchange Standardization Approach.

The *interchange standardization approach* (see Figure 3), has stakeholders agree on a minimum common model for data exchange and, at the same time, allows researchers to continue using their familiar data models and formats and to preserve their habitual workflows and tools. This common model, which corresponds to the white area in Figure 3, represents the concepts that are in the overlap between two or more models and that are relevant for the experiment. This approach needs only converters between each data model and the common model. To be flexible enough for potential changes, the approaches placed in this option should not hard-code the semantics of the models and their relations in the converters, but instead these semantics should be specified explicitly using an appropriate machine-understandable syntax, e.g., ontologies.

Ontologies have been applied in several approaches for the semantic integration of heterogeneous systems. An ontology is a formal description of a domain of discourse [43]. Ontologies capture the semantics of data sources and allow to make the models and the relationships explicit and machine-understandable [44]. Some of the approaches successfully applied for semantic data integration, such as the Engineering Knowledge Base (EKB) approach for systems engineering [45], can be adapted for managing replication data.

III. Research Issues and Approach

The absence of a common data model and variations in the local data models of the researchers make it difficult to collect and analyze the data. Although there have been several attempts to establish central data repositories in the empirical software engineering community, researchers still tend to conduct local variations of experiments and not to strictly follow a common data model. In this context, we address the following two research issues.

RI-1: Replication data management (RDM) needs. We investigate the needs of key stakeholders for RDM to derive criteria for evaluating conceptual RDM approaches. Starting from a literature survey on RDM needs and solution approaches, we collect additional RDM needs from empirical research groups in Valencia, Madrid and Vienna (see also Section II). We analyze RDM stakeholders, most relevant use cases and needs of several empirical replication study research groups to derive a set of evaluation criteria for RDM approaches.

RI-2: Evaluation of RDM approaches. As an alternative RDM approach we introduce the empirical ecosystem concept, adapted to RDM from well-established approaches for data integration in software and systems engineering [45]. Then we evaluate three conceptual RDM approaches based on the criteria identified in RI-1: central repositories (CR), loosely connected heterogeneous local repositories (LR), and the empirical ecosystem (EE). We follow a scientific approach in defining hypotheses for research and report in a qualitative analysis on how well the individual approaches aims at supporting RDM needs.

Since the goal of comparing empirical RDM approaches seems well related to the ATAM [9] goal of analyzing the risks and sensitivities of architecture approaches with use cases, to deal with these research issues, we have adapted and applied the ATAM steps (identify business drivers, identify architecture approaches, derive quality attributes, analyze architecture approaches, discuss weaknesses and risks).

IV. Replication Data management Needs and Solutions Analysis

This section reports on the RDM evaluation following the steps of ATAM adapted to the RDM context and introduces the Empirical Ecosystem, a novel approach for supporting RDM.

A. RDM stakeholders, requirements, and use cases

In order to identify needs from different perspectives (a wide coverage), first we identify the most relevant RDM stakeholders (similar to ATAM step „identify business drivers"). We observed the following stakeholders with partially overlapping responsibilities from the empirical work carried out by UP Madrid, UP Valencia, and TU Vienna (see also Section II.A):

The *Research Manager* is responsible for research directions, interpretation of experiment results and their transfer to practitioners, and publication. Main challenge is keeping an overview on a set of different experiments in context of the research strategy. Main use case is getting an overview, comparing and analyzing statistics on replicated experiments in one research group and/or across different groups. Typical queries that may be needed are: getting an overview of the experiment process state; getting the status of a specific experiment replication or the person that is in charge of it; getting which hypotheses, techniques (factor levels), or response variables were used in the replications of a certain experiment.

The *Experiment Manager* is responsible for individual experiment planning, design, custody of experimental objects, overseeing experiment tasks, and coordination and data analysis. Main challenge is managing a specific experiment or a small set of (replicated) experiments. Main use case is the monitoring, control and analysis of one experiment and/or a small subset of related experiments (e.g., replication studies). Typical queries that may be needed are: what replications of a certain experiment are planned; how many replications there are for a certain experiment; get results of the experiment as an integrated view of the results of its replications.

The *Senior Experimenter* is responsible for execution and data analysis of an individual experiment and could have several roles, such as trainer, analyst or supervisor. Main use case is getting important information of a replication and research issues to be answered based on the research strategy. Typical queries that may be needed are: which are the results of a certain replication, how many subjects participated in a replication; getting which techniques (factor levels), or response variables were used in a certain experiment replication.

The *Meta Analyst* is responsible for the synthesis of experiment results. Typical queries are: finding all experiments that involved certain levels of a factor; identifying contextual variables to carry out subgroup and sensitivity analysis; define new hypotheses based on the findings.

B. RDM stakeholder needs

From the use cases and requirements reported by the stakeholders, we derived the following areas of needs for an effective tool-supported method to integrate the data coming from different sources and models.

Common data model. The method must allow researchers to use their well-known experiment data models, but at the same time, should provide a common data model that allows sharing the information between the related replications (see also Figure 4). This common data model must support the required empirical study process steps, facilitate finding interconnections between replications of the same experiment, and facilitate querying across replications,as basis for statistical analysis of the obtained data.

Data storage. The method must provide a proper data storage to store the specific data of the experiments and their results, captured according to the data model. This data storage should be supported by appropriate tools for managing the collected data. Since replication researchers may want to store different versions of the data, like storing historical results, this storage should also provide data versioning.

Query support. Once the data is collected, replication researchers need query support for analyzing and managing both the experiments information and the data obtained when executing them via the common data model. Typical queries that should be supported have been described in the previous subsection. The method must provide appropriate query support.

Advanced process support. Beyond query support, the method must support more advanced processes like: change propagation between data models and change notification to the opportune researchers; checking the consistency of the data across the diverse empirical studies taking into account the design process and the researcher roles; or the execution of reasoning and analysis algorithms for extracting statistics and relevant knowledge from experiments' results.

C. The empirical ecosystem integration concept

Since the initial literature research and discussion with RDM stakeholders revealed shortcomings in the traditional RDM approaches of central repositories (CR) and local repositories (LR), we introduce an alternative RDM approach, the empirical ecosystem (EE), similar to the ATAM step „identify

architecture approaches". The EE is a novel RDM concept that applies an ontology-based interchange standardization data integration approach (see Section II.C). The main difference to CR is that stakeholders using the EE need to agree only on a minimum common model for data exchange instead of a full data model for the complete repository.

As stated in the previous subsection, every senior experimenter should be able to use his own local data models. However, at the same time, the others stakeholders need to have these local data models interlinked to be able to exchange data and to define queries across different (local) replication data repositories. As explained in Section II.C, the most common and suitable way to achieve these both requirements is to create a minimum common data model where it is defined the relevant information for interconnecting the different local data models, and the information that is overlapped and that should be integrated.

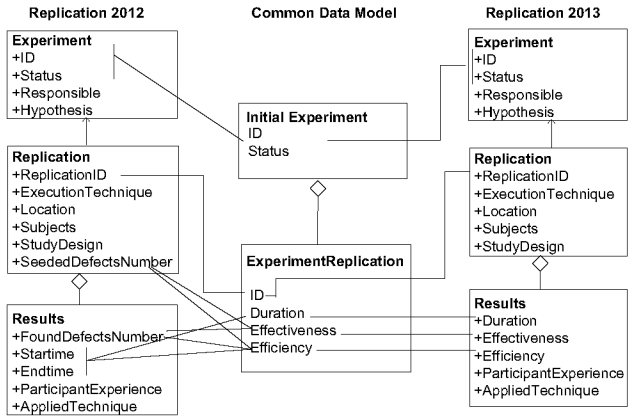

Figure 4: Example of Common Concepts applied to a real RDM use case.

Figure 4 illustrates the application of the EE approach to a recent replication use case carried out at TU Vienna (see Section II.A). This example shows two simplified local data models based on two replications of the same experiment: the data model of the Replication 2012 on the left, and the data model of the Replication 2013 on the right. These data models collect information about the experiment, the replication and the results of its execution. Note that some of the collected information is represented in a different way in these local data models. For instance, Replication 2012 stores the number of seeded and found defects, while Replication 2013 stores the efficiency and the effectiveness. To be able to query these data in a consistent way, the local models are mapped to a common data model (see a simplified version in the middle of Figure 4). This model represents the information that is necessary for connecting the local models (e.g., experiment ID and replication ID) and provides a common representation for integrating the data that the local data models represent in different ways (e.g., duration, effectiveness, efficiency). Based on the common model and its mappings to the local data models, queries across different replications can be efficiently executed.

An implementation of the EE can be based on the *Engineering Knowledge Base* (EKB) [45]. The EKB is a layered semantic model, which has successfully been applied for integrating heterogeneous data models in multidisciplinary systems engineering projects, which is structurally similar to the RDM challenge illustrated in Figure 4. The EKB holds an ontology-based common model that defines the relevant concepts for interconnecting and integrating the data model of every local repository (see Figure 5). In addition, the EKB specifies each local data model in an ontology and represents the mappings between each local data model and the common model using a machine-understandable ontology syntax. In this way, both, the semantics of each data model and the semantics of their relationships with the common model are explicitly represented.

This allows senior experimenters to work with their own local repositories, but also allows semantic querying across different local data models through the common model (see Figure 4). Thus, research managers can also derive results or aggregation of data based on queries via the common model. The EKB also provides services and interfaces to access, query, and analyze the data stored in the local repositories, such as the use of dashboards to enable interaction with heterogeneous data from the local repositories.

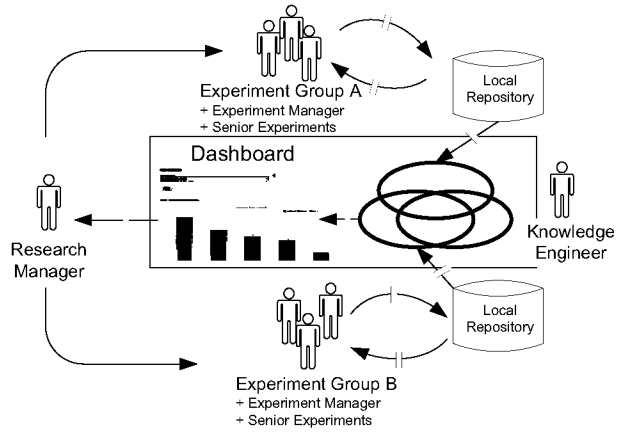

Figure 5: Empirical Ecosystem to support collaboration within (groups of) empirical experiment data repositories.

D. RDM evaluation critera

Based on identified needs and similar to the ATAM step "derive quality attributes", we have derived evaluation criteria for comparing RDM approaches. The criteria cover the four areas of RDM stakeholder needs: the application of data models, data storage and availability of data, queries to study data for evaluation purposes, and advance study process support.

Data Model Criteria. *A1. Study process support* measures capabilities for monitoring and control of experiments and the study progress based on the current state of the experiment. *A2. Representation of roles/responsibilities* rates the ability to clearly assign tasks to individual roles and people based on the study setting. *A3. State of studies and data versions* (e.g., planned, raw, and refined data) evaluates the capability to process monitoring and control of evaluation steps. *A4. Researchers can keep using their well-known tools and data models* rates whether users can work without changing their well-established experiment environment.

Data Storage Criteria. *B1. Tool support* rates the capability for managing the collected data effectively and efficiently. *B2. Versioning of study data* for raw and analyzed data rates the capability for tracing back study data to previous evaluation steps and data versions, relevant if there are several branches of data analysis steps.

Queries to Empirical Study Data Criteria. *C1. Overview for project management* rates the capability for observation and control across different experiments and experiment groups. *C2. Query to data on several studies via common/local data model* rates the capability to define and answer queries across studies based on the common data model and on local data models.

Advanced Empirical Study Process Support Criteria. *D1. Change propagation between local data models* rates the capability to notify related roles in case of data changes to the experiment setting or data. *D2. Consistency checks across, empirical studies and roles* evaluates the capability for sufficient checks to find data inconsistencies early. *D3. Reasoning, run algorithms* for analyzing experiment results rates the capability to apply sophisticated algorithms to the repository content. *D4. Include new repositories of experimental data* from different experiments, i.e., enabling an aggregated analysis of experiment data collected at different research organizations, rates the capability to integrate new data models that do not exactly follow the existing data model.

E. Comparison of RDM Approaches

Similar to the ATAM step "analyze architecture approaches" we evaluated three established and/or promising conceptual RDM approaches: the *central repository (CR) concept* (see Section II.B, Figure 3), the *local repository (LR) concept* (see Section II.B, Figure 4), and the *empirical ecosystem (EE) concept* (see Section IV.C, Figure 5). Based on the key criteria (see Section IV.D), we analyzed the architecture of the solution concept variants to determine how well these variants fulfill each of the criteria.

Table 1 summarizes the results of the qualitative analysis of the strengths and weaknesses of the RDM approach variants regarding the RDM needs elicited from the research groups' use cases (see Sections IV.A and IV.B). The color codes of the cells in Table 1 suggest how well a concept supports a given criterion: Green means good capability, yellow means limited capability, and red means insufficient capability.

The *central repository* (CR) concept can be classified as a global standard data integration approach since the CR uses a fixed and unified global standard model. Overall the CR concept provides good or limited support for replication researchers. Regarding the data model criteria, CRs tend to prescribe the parameters for the empirical study setting. Only some CRs support the work with versioned empirical data. The major limitation of CRs is the low flexibility to adapt the central data model to work with local tools and data models. While overview for project management is a feature supported by all CRs, most criteria regarding data storage, queries, and advanced process support are available only within the predefined central database. A major limitation of CRs is the high effort to adapt the central data model to work with new data models to accommodate new empirical study setups.

The *local repository* (LR) concept can be classified as a brute-force data integration approach since data exchange and aggregation are manually performed. The LR concept has the most severe limitations of the concepts studied as the support for coupling several local repositories is needed by the replication researchers but not provided by the local repository concept. Therefore, the support for local empirical processes and analysis for individual researchers is well solved, however, the support for a research group and, in particular, distributed research groups is only weak. The architecture and semantics of local data models makes the integration of data models effort intensive and error prone.

The *empirical ecosystem* (EE) concept seems well suited for integrating local repositories to combine the flexibility of local repositories with the need of data analysis across several data sources. The efficient mapping and automated transformation of data between common concepts and local representations supports both individual researchers and distributed work groups. However, the EE concept needs to be investigated to collect evidence on its merits in practical RDM use.

V. DISCUSSION

This section discusses the results regarding the research issues, reports lessons learned on success and risk factors, and discusses threats to validity of the study results.

RI-1: Replication data management (RDM) needs. During this study we identified four roles involved in empirical replication studies, who have needs for a RDM approach: the research manager, the experiment manager, the senior experimenter, and the meta analyst. An interesting outcome of our investigation was that the researchers could not conduct their use cases easily due to inflexible data models. The main causes of this problem were (a) that the use cases and needs of these roles change and grow over time; (b) the choice of data models and data storage technology are often left to individual researchers; and (c) data models and technology are found to be hard to adapt as more comprehensive use cases come up on the research agenda.

Individual researchers often tend to choose a variety of low-tech solutions such as Excel sheets, scripts, and simple databases to address their short-term needs. Unfortunately, the information encoded in these solutions is hard to reuse and scale up efficiently for analyses across researchers and research groups. On the other hand the providers of data repositories, experiment management systems, and knowledge and experience management systems (see the overview in Section II.B) focus on the role of the meta analyst and the other roles only to a limited extent.

Therefore, an important question coming from this research is: what kind and how much interaction between local individual researchers, research groups, and coordinating/mediating repository providers is sufficient to facilitate the effective and efficient exchange of data models and empirical data in a RDM context in order to support significant research progress in an empirical research field?

RI-2: Evaluation of RDM solution design approaches. The comparison of three conceptual RDM approaches assumed as background an empirical research field with data models that change over time. Against this background, CR and LR concepts were found to have major issues that are hard to resolve in practice, while the EE concept allows bridging current gaps in replication data management.

Table 1. Comparison of Replication Data Management Approaches.

	Data Management Approaches		
	Central repository (CR)	***Local repositories (LRs)***	***Empirical Ecosystem (EE)***
Data Model Criteria			
A1. Study process support for empirical study process steps.	Limited to one common process.	Limited to local processes.	Support of common and local processes.
A2. Representation of roles/responsibilities.	Centrally defined roles/responsibilities.	Individual local roles/responsibilities.	Flexible local implementations.
A3. States of studies and data versions, e.g., planned; raw, refined data.	Supported by some CRs.	Usually, no.	Data versioning available.
A4. Researchers can keep their well-known tools and data models.	No.	Yes, but not connected.	Yes, including data exchange support.
Data Storage Criteria			
B1. Tool support for data management.	Within the common database.	No.	Yes.
B2. Versioned of study data.	Supported by some CRs.	Usually, no.	Yes.
Queries to Empirical Study Data Criteria			
C1. Overview for project management.	Yes.	Limited to individual studies.	Yes.
C2. Query to data on several studies via common/local data model.	Yes, if included in the central database.	No.	Yes: common data model; limited for local data models.
Advanced Empirical Study Process Support Criteria			
D1. Change propagation between local data models	Yes, if included in the central database.	No.	Yes.
D2. Consistency checks across, empirical studies and roles.	Yes, if included in the central database.	No.	Yes.
D3. Reasoning, run analysis algorithms.	Yes, if included in the central database.	In local repositories.	Yes.
D4. Include new repositories of experimental data.	Yes, high effort for data integration.	No.	Yes, low effort for mapping.

The strength of the *CR approach* is its convenience for repository keepers and meta study researchers. However, a major risk is often the extra effort for local researchers to document and convert their local data sources, in particular, if the CR data model was not available when designing the local data models.

The strength of the *LR approach* is its convenience for local researchers. However, for sufficiently complex empirical data models the LR concept does not support efficient data integration across sites and often the knowledge on data is not documented in sufficient detail to use the data without the expert who designed the (informal) data model.

The strength of the *EE concept* is its promise to provide replication researchers with a common data model that collect data from heterogeneous and distributed data models and to allow local researchers the freedom to work with their local data model. Note, that EE roles are not fundamentally different to the CR roles. However, knowledge engineers are needed on global and local levels for the definition of the data model, data transformation, and data validation. Only little extra effort is needed for the "mapping" of data models on global and local levels for data transformation, most effort tends to go into data validation if mapping and transformation are not done well. Nevertheless, common data models can facilitate the sharing of experimental data, enabling independent analysis (/re-analysis) by third parties, experimental aggregation and eventually providing the building blocks for e-science initiatives (e.g., myExperiments.org). However, while the EE concept has been successfully used in systems engineering environments, its application to empirical environments needs to be prototyped and evaluated.

Lessons learned. During this study we found the following success/risk factors for CR/LR approaches, which merit further investigation.

Researchers' needs not defined explicitly. While there is some agreement on desirable results of empirical replication studies, there is, to our best knowledge, no consolidated set of requirements documented on data and functions for researchers to provide sufficient access to accumulated data.

Data model variation (local). Experiment designs always vary somewhat from previous experiments to look at new aspects; therefore, the data model also varies.

(Global) Data model missing. Even if there is a published basic experiment design, there is, in general, no experiment data model agreed across experiments.

Data semantics unclear. Even if data models exist, e.g., as UML or EER models, the semantics of the data is agreed only for the most basic data elements, if at all.

Data availability. Some relevant data is encoded, e.g., into filenames, data field names, and therefore difficult to extract automatically with high precision. Further challenges focus on heterogeneous and changing local sources and knowledge available only locally and diminishing over time.

Effort to collect and analyze data. Effort to collect and verify data is significant for both replication researchers and local researchers, which may prevent interesting research.

Conflicts of interest. Local researchers want to keep their way of working, in particular, their data models; while replication researchers want to work with a common data model including relevant aspects from different local sources.

Threats to validity. As in all empirical work there are threats to validity that need to be addressed.

Internal validity. The quality of elicitation and definition of the RDM constructs such as use cases, needs, and evaluation criteria may suffer from flaws in the research method. To address this threat we started from the empirical process model steps for new and replicated experiments (see Section II. A), and worked with experienced persons in experiment roles from research groups active in replication studies.

External validity. The small number of research groups involved and the overlap of study authors and participating experts in the evaluation study may threaten external validity. Nevertheless, even the limited scope of the study helped to identify important RDM needs and options for improvement. Note that the core contribution of this work focuses on analyzing RDM needs. In addition we provide an initial and qualitative evaluation of RDM solutions which might include some limitation. Thus, external validity will be strengthened in using and extending the evaluation framework in future work with independent research groups.

VI. Summary and Future Work

An important objective of replication studies is to enable data analysis across data collections of several iterations of an experiment. Unfortunately, the absence of a common data model and variations in the local data models of the researchers make it difficult to collect and analyze the data. Despite attempts to establish central data repositories in the empirical software engineering community, researchers still tend to conduct local variations of experiments and do not follow a common data model.

In this paper, we adapted the ATAM method to analyze the replication data management (RDM) needs and use cases of several empirical replication study research groups, and to compare three conceptual approaches regarding the detected needs: central data repository with fixed data model, heterogeneous local repositories, and an empirical ecosystem. Major results are: the approach for RDM was found useful and usable by the researchers involved. While the local and central approaches have major issues that are hard to resolve in practice, the empirical ecosystem allows bridging current gaps in replication data management.

For future work we propose to adapt the empirical ecosystem (EE) concept from software and systems engineering to provide an infrastructure for empirical experimentation. This will allow (a) synchronizing several local data models in order to establish a common data model for data analysis across heterogeneous local data models, while (b) allowing the researchers to keep working with their well-known views on their research data.

To better understand the diversity of RDM needs, we propose to analyze the team structure and the common and local data models of diverse empirical research groups. This study focused on primary stakeholders, i.e., researchers that conduct, analyze, and put together experimental replications in software engineering. Secondary stakeholders may be other parties interested in RDM-based tools, such as journal editors (e.g., to provide permalinks to datasets), reviewers and readers of experimental publications (e.g., re-analysis) or other researchers (e.g., repository mining). The needs of secondary stakeholders were not in the scope of the study reported in this paper but should be explored in future work.

An example need of empirical researchers that has been identified during the development of this work is the *need for a repository on empirical studies on testing*. This goes beyond the RDM needs, but would greatly help to efficiently find experiments that are related and relevant for a planned experiment. Precision (and recall) of existing search engines are very low, and, although some attempts for optimizing a search strategy has been proposed, it is still not enough. In the future work we plan to propose an integration approach that allows creating such repositories. It should provide a balance between generality and precise metadata for efficiently interconnect the created empirical studies in a particular research area. For creating this repository, we plan to study the body of knowledge of empirical research to represent the main concepts and relationships used for storing the experiment data. Using this repository, empirical researchers could gather important knowledge for their experiments like: which existing projects are similar to mine (e.g., context, size, or type of problem); get the 10 best testing strategies, their effectiveness, and efficiency; get related studies that have been developed in one session; get data and materials that have been used in similar experiments.

References

[1] M. Ciolkowski, "What do we know about perspective-based reading? An approach for quantitative aggregation in software engineering," in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009, pp. 133–144.

[2] J. E. Hannay, T. Dybå, E. Arisholm, and D. I. K. Sjøberg, "The effectiveness of pair programming: A meta-analysis," *Information and Software Technology*, vol. 51, no. 7, pp. 1110–1122, 2009.

[3] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, "Experimentation in Software Engineering," *Springer*, 2012.

[4] N. Paton, "Managing and sharing experimental data: standards, tools and pitfalls," *Biochemical Society Transactions*, vol. 36, pp. 33–36, 2008.

[5] H. Do, S. Elbaum, and G. Rothermel, "Supporting controlled experimentation with testing techniques: An infrastructure and its

potential impact," *Empirical Software Engineering*, vol. 10, no. 4, pp. 405–435, 2005.

[6] E. Arisholm, D. I. K. Sjøberg, G. J. Carelius, and Y. Lindsjørn, "SESE an experiment support environment for evaluating software engineering technologies," in *NW-PER2002 (Tenth Nordic Workshop on Programming and Software Development Tools and Techniques). Copenhagen, Denmark*, 2002, pp. 81–98.

[7] S. Biffl, M. Ciolkowski, and F. Shull, "A Family of Experiments to Investigate the Influence of Context on the Effect of Inspection Techniques," in *Proceedings of the Empirical Assessment in Software Engineering*, IEE, 2002.

[8] D. Winkler, R. Varvaroi, G. Goluch, and S. Biffl, "An Empirical Study On Integrating Analytical Quality Assurance Into Pair Programming," in *5th ACM-IEEE International Symposium on Empirical Software Engineering*, 2006, pp. 21–23.

[9] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, and J. Carriere, "The architecture tradeoff analysis method," in *Fourth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS '98)*, 1998, pp. 68–78.

[10] D. Dhungana, I. Groher, E. Schludermann, and S. Biffl, "Software Ecosystems vs. Natural Ecosystems: Learning from the Ingenious Mind of Nature," in *Proceedings of the Second Workshop on Software Ecosystems*, 2010, pp. 96–102.

[11] V. R. Basili and R. W. Selby, "Comparing the effectiveness of software testing strategies," *Software Engineering, IEEE Transactions on*, no. 12, pp. 1278–1296, 1987.

[12] E. Kamsties and C. Lott, "An empirical evaluation of three defect-detection techniques," in *Proceedings of the Fifth European Software Engineering Conference*, 1995, pp. 362–383.

[13] S. T. Acuña and N. Juristo, "Assigning people to roles in software projects," *Software: Practice and Experience*, vol. 34, no. 7, pp. 675–696, 2004.

[14] G. M. Marakas and J. J. Elam, "Semantic structuring in analyst acquisition and representation of facts in requirements analysis," *Information Systems Research*, vol. 9, no. 1, pp. 37–63, 1998.

[15] N. Juristo, A. M. Moreno, and S. Vegas, "Reviewing 25 Years of Testing Technique Experiments," *Empirical Software Engineering*, vol. 9, no. 1/2, pp. 7–44, Mar. 2004.

[16] N. Juristo and S. Vegas, "Using differences among replications of software engineering experiments to gain knowledge," in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009, pp. 356–366.

[17] O. Dieste and N. Juristo, "Systematic review and aggregation of empirical studies on elicitation techniques," *IEEE Transactions on Software Engineering*, vol. 37, no. 2, pp. 283–304, Mar. 2011.

[18] N. Condori-Fernández and O. Pastor, "Re-Assessing the Intention to Use a Measurement Procedure based on COSMIC-FFP," *ENSUR A*, p. 63, 2006.

[19] N. Condori-Fernández, S. Abrahão, and O. Pastor, "On the estimation of the functional size of software from requirements specifications," *Journal of Computer Science and Technology*, vol. 22, no. 3, pp. 358–370, 2007.

[20] B. Marín, O. Pastor, and A. Abran, "Towards an accurate functional size measurement procedure for conceptual models in an MDA environment," *Data & Knowledge Engineering*, vol. 69, no. 5, pp. 472–490, 2010.

[21] S. Abrahão and G. Poels, "A family of experiments to evaluate a functional size measurement procedure for Web applications," *Journal of Systems and Software*, vol. 82, no. 2, pp. 253–269, 2009.

[22] N. Condori-Fernández and O. Pastor, "Analyzing the Applicability of a Theoretical Model in the Evaluation of Functional Size Measurement Procedures," in *SEKE*, 2007, pp. 736–739.

[23] N. Condori-Fernández and O. Pastor, "Towards a Theoretical Model for Evaluating the Acceptance of Model-driven Measurement Procedures," in *SEKE*, 2008, pp. 22–25.

[24] S. Biffl and M. Halling, "Managing Software Inspection Knowledge for Decision Support of Inspection Planing," in *Managing Software Engineering Knowledge*, Springer, 2003, pp. 231–249.

[25] S. Biffl and W. Grossmann, "Evaluating the accuracy of defect estimation models based on inspection data from two inspection cycles," pp. 145–154, Jul. 2001.

[26] S. Biffl, P. Grünbacher, and M. Halling, "A Family of Experiments to Investigate the Effects of Groupware for Software Inspection," *Journal of Automated Software Engineering*, vol. 13, no. 3, pp. 373–394, 2006.

[27] M. Halling, S. Biffl, and P. Grunbacher, "An experiment family to investigate the defect detection effect of tool-support for requirements inspection," *Proceedings 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry IEEE Cat No03EX717*, pp. 278–285, 2003.

[28] T. Menzies, B. Caglayan, E. Kocaguneli, J. Krall, F. Peters, and B. Turhan, "The PROMISE Repository of empirical software engineering data," *West Virginia University, Department of Computer Science, 2012*. [Online]. Available: http://promisedata.googlecode.com.

[29] H. Do, S. Elbaum, and G. Rothermel, "Software-artifact infrastructure repository." [Online]. Available: http://sir.unl.edu/content/sir.php. [Accessed: 12-Jan-2012].

[30] M. G. Mendonca, J. C. Maldonado, M. C. F. de Oliveira, J. Carver, S. C. P. F. Fabbri, F. Shull, G. H. Travassos, E. N. Hohn, and V. R. Basili, "A framework for software engineering experimental replications," in *13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2008)*, 2008, pp. 203–212.

[31] W. D., K. M., S. C., and B. S., "Investigating the Impact of Experience and Solo/Pair Programming on Coding Efficiency: Results and Experiences from Coding Contests," *14th XP Conf. 2013*, 2013.

[32] A. Lam and B. Boehm, "Experiences in developing and applying a software engineering technology testbed," *Empirical Software Engineering*, vol. 14, no. 5, pp. 579–601, 2008.

[33] L. Hochstein, T. Nakamura, and F. Shull, "An Environment of Conducting Families of Software Engineering Experiments," *Software Development*, vol. 74, no. 8, pp. 175–200, 2007.

[34] E. Arisholm and D. Sjøberg, "A web-based support environment for software engineering experiments," *Nordic Journal of Computing*, vol. 9, no. 3, pp. 231–247, 2002.

[35] V. Basili, G. Caldiera, and H. Rombach, "Experience factory," *Encyclopedia of software*, vol. 1, pp. 469–476, 1994.

[36] J. Travassos, G. H., dos Santos, P. S. M., Neto, P. G. M., & Biolchini, "An environment to support large scale experimentation in software engineering," *In 13th IEEE International Conference on Engineering of Complex Computer Systems, 2008. ICECCS 2008.*, pp. 193–202, 2008.

[37] D. Wang, W. Zhang, J. Chen, Y. Yang, and Q. Wang, "EXPRE," in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '10*, 2010, p. 1.

[38] B. Boehm and V. Basili, "The cebase framework for strategic software development and evolution," in *Third International Workshop on Economics-Driven Software Engineering Research (EDSER-3 2001)*, 2001.

[39] B. Hofmann and V. Wulf, "Building Communities among software engineers: the VISEK approach to intra-and inter-organizational learning," *Advances in Learning Software Organizations*, pp. 25–33, 2003.

[40] A. Halevy, "Why Your Data Won't Mix: Semantic Heterogeneity," *Queue*, vol. 3, no. 8, pp. 50–58, 2005.

[41] S. Bergamaschi, S. Castano, and M. Vincini, "Semantic Integration of Semistructured and Structured Data Sources," *SIGMOD Record*, vol. 28, no. 1, pp. 54–59, 1999.

[42] A. Wiesner, J. Morbach, and W. Marquardt, "Information integration in chemical process engineering based on semantic technologies," *Computers & Chemical Engineering*, vol. 35, no. 4, pp. 708–692, 2011.

[43] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, vol. 5, no. 2, p. 199{220.

[44] H. Wache, T. Voegele, and U. Visser, "Ontology-based integration of information-a survey of existing approaches," *Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing*, pp. 108–117, 2001.

[45] T. Moser and S. Biffl, "Semantic Integration of Software and Systems Engineering Environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 38–50, Jan. 2012.