Expert Systems With Applications 96 (2018) 330–346

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Double-loop learning in project environments: An implementation approach

Benjamin Matthies*, André Coners

South Westphalia University of Applied Sciences, Haldener Strasse 182, Hagen 58095, Germany

ARTICLE INFO

Article history:
Received 8 December 2016
Revised 30 October 2017
Accepted 6 December 2017
Available online 7 December 2017

Keywords:
Project management
Project knowledge
Double-loop learning
Latent semantic analysis
Analytic network process

ABSTRACT

Project-based learning is based on the idea of iteratively learning for future projects from the successes and failures of past projects. This paper proposes a semi-automated implementation approach for double-loop learning in project environments. A combined application of two complementary methods is suggested for this purpose: Latent Semantic Analysis (LSA) and Analytic Network Process (ANP). By this means, the approach addresses two problems of the project management practice. First, the information overload in project environments, whereby the LSA is used for the semi-automated extraction of lessons learned from large collections of textual project documentation. Second, the lack of procedures and methods for the practical implementation of available project knowledge, whereby the ANP is used for the systematic modeling of extracted lessons learned and their integration into the evaluation of project concepts and current project management routines. Thus, the proposed implementation approach improves the ability of project-based organizations to consequently learn from past failures or successes. From a practical perspective, evident shortcomings of existing computerized double-loop learning approaches are addressed. The proposed approach contributes to the project management practice not only by demonstrating a solution for the exploration of representative and potentially new lessons from multiple combined experience reports, but also by presenting a solution for the systematic assessment of such project-governing variables and their mutual relationships as part of the decision-making in new projects. From a theoretical perspective, specific research avenues for further development of the double-loop learning concept by means of expert and intelligent systems are provided.

1. Introduction

Double-Loop Learning (DLL) is an organizational learning concept and includes the idea that an organization continuously learns and evolves as a whole, if collective experiences (*e.g.*, failures and successes) are used for evolving the organizational reference framework through consistent correction of its basic governing variables (Argyris, 1977; Argyris, 1999; Argyris & Schön, 1978). In more specific terms, this means that learned lessons are not only used for future problem-solving in similar situations, but for the reevaluation and modification of the underlying standards, policies, procedures, and objectives of a whole organization as well (Kantamara & Vathanophas, 2014).

Transferred to project-based organizations, DLL means that lessons learned from past projects are used in order to develop the basic project management framework for the design and implementation of future projects (see, *e.g.*, Ayas & Zeniuk, 2001; Brady & Davies, 2004; Holt, Love, & Li, 2000; von Zedtwitz, 2002; Wong, Cheung, & Fan, 2009). An important source of knowledge for such learning is project documentation, which codifies the experiences of completed projects (*e.g.*, problems and their solutions) and makes it available for reuse in project-based organizations (von Zedtwitz, 2002). The analysis of such codified project experiences supports future projects in scrutinizing ingrained routines and can ultimately prevent previous errors from being repeated (Koners & Goffin, 2007). Consequently, such repositories of codified project knowledge are an important asset of a continuously learning organization (Disterer, 2002; Parnell, Von Bergen, & Soper, 2005).

In the project management practice, however, the actual use of codified project knowledge falls far short of the corresponding ideal of project-based learning (Almeida & Soares, 2014; Barclay & Osei-Bryson, 2010). Project-based organizations are confronted by two central problems when reusing historical project documentation: the information overload due to often very extensive document collections as well as a lack of systematic procedures for the operational implementation of project knowledge in new projects.

* Corresponding author.
E-mail addresses: matthies.benjamin@fh-swf.de (B. Matthies), coners.andre@fh-swf.de (A. Coners).

https://doi.org/10.1016/j.eswa.2017.12.012
0957-4174/

Both problems are addressed in this article and are described below in greater detail.

The problem of information overload (see Eppler & Mengis, 2004, for a theoretical definition) describes the fact that project managers are often confronted by extensive, heterogeneous, and mostly textual repositories of project documentation (Caniëls & Bakens, 2012; Haksever, 2000; Strait, 2006). They are faced with the challenge of identifying useful historical lessons learned in a reasonable effort and ultimately to synthesize relevant project knowledge for learning in future projects. Haksever and Fisher (1996, p. 1) aptly emphasize this problem: "... unless it is understood and managed well, information overload can be a critical information problem which prevents project managers from performing their tasks effectively". To address this problem, methods are needed that can not only analyze the volume, but also the complexity of the mostly textual experience reports (Carrillo, Harding, & Choudhary, 2011; Choudhary, Oluikpe, Harding, & Carrillo, 2009). In summary, the following research question (*RQ*) can be extrapolated from the problem described:

RQ1: How can relevant project knowledge be extracted from extensive textual project documentation in a reasonable effort?

The second problem addressed is the lack of systematic procedures and effective methods for the operational implementation of existing project knowledge in projects. Here Barclay and Osei-Bryson (2010) criticize the fact that even if relevant lessons learned from past projects are available, they are incorporated in an unstructured way into the design of new projects. Instead, available project knowledge is incorporated based on the intuitive and subjective discretion of the individual project manager. As a result, this lack of systematic procedures means that even if the knowledge potential exists for project-based learning, it is often not implemented in a consistent learning process. The learning potential is therefore not only lost for individual projects, but also for the development of the entire organization. This problem is also addressed in this paper and can be summarized in the following research question:

RQ2: How can extracted project knowledge be systematically implemented in the continuous learning process of a project-based organization?

Despite the long history of incorporating expert and intelligent systems into organizational learning (Chou, 2003) and knowledge management processes (Liebowitz, 2001), comparably little attention has been paid to the development of systems and tools for the specific purpose of operationalizing DLL in organizations (Jaaron & Backhouse, 2017). However, such systems embedded into the DLL process have great potential to facilitate learning by, for example, improving the accessibility of large knowledge databases, supporting the task of model-building, expanding the computerization of decision-making processes, and by integrating expert knowledge into the assessment of complex learning scenarios (Bhatt & Zaveri, 2002). Therefore, this article addresses the aforementioned practical problems by incorporating established techniques from the field of expert and intelligent systems into the DLL process. The aim is to provide not only an approach which facilitates the semi-automated discovery and modeling of historic lessons learned from textual project documentation, but also an approach for their systematic assessment and implementation in the course of new projects. For this purpose, a combined application of two complementary methods is proposed: *Latent Semantic Analysis* (LSA) and *Analytic Network Process* (ANP). The application of LSA is aimed at solving the first problem (*RQ1*) and synthesizes thematic core concepts (*i.e.*, lessons learned) from extensive collections of textual project documentation. By this means, potential success factors and sources of error in past projects, and thus learning potential for future projects will be identified. Building on that, the application of ANP aims to solve the second problem (*RQ2*), in which the extracted lessons learned are modeled as basic project-governing variables in a hierarchical decision network (the so-called Project Evaluation Network) and systematically integrated into the evaluation of new project concepts and current project management routines. Based on its purpose and procedure, the proposed solution can therefore be characterized as an intelligent decision-support approach (Turban & Watkins, 1986), which facilitates the decision-making ability of project-based organizations in the course of a continuous learning process. The application and benefit of this implementation approach will be demonstrated in a practical case study. Based on a total of 78 historical project documentations, 26 central lessons learned are initially extracted and modeled in a Project Evaluation Network (PEN) using LSA. This decision network is then used for the evaluation, prioritization, and further development of two alternative project concepts using ANP.

This article is structured as follows. Section 2 initially discusses the concept of DLL in the project environment as well the associated role of project documentation. Section 3 presents the conceptual framework of the proposed implementation approach, the methods used (LSA and ANP), as well as the demonstrative case study. Section 4 contains the first part of the demonstrative analysis: the determination of the historical lessons learned based on the LSA. Building upon that in Section 5, the second part of the analysis is demonstrated: the evaluation of project concepts based on the ANP. Section 6 discusses the tangible benefits of the proposed approach as well as future avenues for research in this field. Section 7 provides a closing summary.

2. Research background

2.1. Double-loop learning in project environments

There is a widespread agreement in the discourse of the knowledge-based perspective of the firm (see Grant, 1996) that organizations must continuously gain new knowledge in order to achieve and secure business success (see, *e.g.*, Argote, 2011; Easterby-Smith & Lyles, 2011; Huber, 1991; Pemberton & Stonehouse, 2000). Such organizational learning can be seen as further developing organizational skills, which is based on the critical exploration of learned lessons (*e.g.*, solved problems) and the subsequent changes in behavior (Argyris, 1977; Garvin, 1993). In light of this understanding, two such organizational learning processes can be distinguished: *singe-loop learning* and *double-loop learning* (Argyris, 1977; Argyris, 1999; Argyris & Schön, 1978). Single-Loop Learning (SLL) occurs when problems and errors are identified and resolved, the appropriate lessons are documented and stored in organizational knowledge repositories, but no analysis of the underlying causes in the entire organizational context is carried out. Thus, SLL is an isolated problem-solving process based on the change in behavior in future, similar situations (Kantamara & Vathanophas, 2014). Double-Loop Learning (DLL), on the contrary, is a concept of reflective learning that occurs when the learned lessons are not only used for isolated problem-solving, but also to examine and modify the standards, procedures, policies, and objectives underlying the problems in the whole organizational context (Kantamara & Vathanophas, 2014). DLL therefore aims for a continuous and profound change in the high-level governing variables of an entire organization. Accordingly, Holt et al. (2000), p. 417) aptly describe DLL as "a high level of evaluation and analysis of information into knowledge, enabling changes to be made for mutual benefit." Several studies also confirm its suitability for promoting organizational learning and for deeply developing the knowledge and skills of an organization (see, *e.g.*, Wong et al., 2009).

The DLL concept can also be applied to the specific context of a project-based organization. Project-based organizations are characterized in particular by the fact that major business functions are performed in unique and temporary projects rather than in typical departments of functional organizations (for a definition, see Hobday, 2000; Middleton, 1967). In this context, Walker and Christenson (2005) affirm that such project environments are particularly well-suited for DLL. This assumption is based on the familiar principle of project-based learning, which is based on the idea of iteratively learning and benefiting in future projects from the successes and failures of past projects (see Keegan & Turner, 2001; Middleton, 1967; Parnell et al., 2005; Schindler & Eppler, 2003). Following this idea, DLL in project environments means not only that problems and errors experienced in the individual projects are resolved and the appropriate lessons learned documented for use in future, similar project situations (according to SLL), but also that such lessons are incorporated into the continuous, *i.e.*, iterative, modification of the basic governing variables throughout the whole project-based organization.

Several proposals for an implementation of the DLL concept have been made in the literature, which can be divided into organizational and technological approaches. Organizational approaches, on the one hand, intend to promote collective DLL through supportive organizational structures, policies, or processes (for corresponding studies, see Ayas & Zeniuk, 2001; Brady & Davies, 2004; Holt et al., 2000; von Zedtwitz, 2002; Walker & Christenson, 2005; Wong et al., 2009). Holt et al. (2000), for example, suggest establishing learning alliances, which are intended to foster DLL through the exchange of knowledge in inter-organizational partnerships. Furthermore, Walker and Christenson (2005) propose the establishment of a Project Management Center of Excellence (CoE), which should help to overcome mere individual learning in projects (intra-project learning) and accelerate collective DLL in a cross-project context. Technological approaches, on the other hand, should facilitate DLL processes through the integration of knowledge management systems and intelligent expert systems. Although the supporting role of such systems to organizational learning activities has been emphasized for a long time (Bhatt & Zaveri, 2002; Chou, 2003), technological approaches seem to be underrepresented in the body of DLL literature in comparison (Jaaron & Backhouse, 2017). Table 1 gives an overview of related research contributions in this specific field. In order to make the goals and motivation behind this article more understandable, the related technological approaches will be discussed and their limitations described in the following.

Previous technological approaches can roughly be divided into three main research streams. The first field of research focuses on the development of knowledge databases, which support the structured storing of relevant historic experiences as well as their targeted retrieval and representation in specific knowledge reuse scenarios (see, *e.g.*, Reychav, Kumi, Sabherwal, & Azuri, 2016; Weber, Aha, Muñoz-Ávila, & Breslow, 2010). Weber et al. (2010), for example, propose an intelligent lessons learned system which supports decision-making processes through clear lessons learned representation that highlights specific reuse conditions. The second research stream focuses on the simulation modeling in order to explain and predict learning factors (*e.g.*, consequences of problems) in complex environments (see, *e.g.*, Kim, MacDonald, & Andersen, 2013; Pérez-Bennett, Davidsen, & E. López, 2014; Takahashi, 2006). Kim et al. (2013), as a distinctive example, demonstrated how simulation models can enhance DLL through the ability to test multiple hypotheses and to calculate potential outcomes for a specific learning scenario of interest. Takahashi (2006), as another example, developed simulation models which can be applied in a multi-layer hierarchical structure in order to simulate diverse problems in adaptive organizational situations. The third research stream aims at the implementation of statistical methods and machine learning algorithms in order to uncover patterns and causal relationships between certain learning factors which, ultimately, deliver interpretable explanations of failures or success factors for decision-making (see, *e.g.*, Bohanec et al., 2017a; Ting, Kwok, Tsang, & Lee, 2010; Zhan & Zheng, 2016). Zhan and Zheng (2016), for example, suggested a solution that explores the causal relationships within a pre-defined framework comprising a set of failures and their potential causes.

In general, all previous DLL approaches have in common that they support decision-making in specific learning situations and thereby ultimately facilitate DLL processes in organizations. However, in spite of the value of these existing approaches, three central limitations can be identified. First, the discussed approaches largely follow the strategy of exploiting historic databases in order to answer specific but already known questions of interest. In most cases, a certain set of learning variables (*i.e.*, problems or success factors) is pre-defined by human experts and subsequently analyzed in order to uncover explanatory factors (*i.e.*, causes or consequences). However, since DLL is based on the idea of overcoming "opportunity blindness" (Grisold & Peschl, 2016) and pursues the goal of identifying and correcting currently unrecognized defects in an organization, the open exploration of databases and discovery of previously unknown issues is of central interest. Consequently, in order that new ideas and learning be generated for the future, the data should speak for itself instead of merely being exploited for answers to the questions asked by human experts. Therefore, tools are needed which can discover and extract such hidden issues from historic data.

Second, especially the approaches centering on knowledge databases aim particularly at the identification and explicit representation of relevant knowledge contents by means of information retrieval techniques (*e.g.*, ontology-based knowledge retrieval systems). However, such approaches are limited in solving the information overload problem when a particularly large number of cases is gathered for a specific problem. These cases (or documents) must then, in turn, be manually screened and summarized, since the isolated analysis of individual cases often does not give a representative insight into the standard problems, success factors, or their relationships. In order to arrive at an appropriate basis for decision-making, tools are needed which can synthesize representative lessons learned expressed in multiple contextually related cases.

Third, the clear interpretation and assessment of lessons learned is of vital importance for assuring a targeted learning in organizations (Bhatt & Zaveri, 2002). Most of the discussed approaches include supportive procedures for interpreting and assessing the identified learnings in the course of decision-making. However, the approaches do not allow for a comparative analysis of lessons learned in regard to their relationships and mutual influences. Such assessments should take cognizance of the fact that a decision on one partial aspect often directly influences additional aspects in a learning situation – positively as well as negatively. Therefore, in order to deploy the limited capacities of a project effectively, tools are needed that support such a comparative evaluation and, finally, the prioritization of lessons learned to be implemented. In sum, such limitations, combined with the practical challenges described in the introduction, can lead to a diminished ability to learn from past failures or successes.

The motivation behind this study is to address the limitations of existing solutions and thereby to contribute to the implementation of DLL in organizations. For this purpose, an implementation approach for DLL in the project environment is suggested (see Section 3), which utilizes techniques from the field of expert systems and supports the computerized exploration of new and representative lessons learned as well as their systematic assessment

Table 1
Technological approaches for implementing DLL in organizations.

Literature	DLL Implementation	Techniques and Tools
Bohanec, Robnik-Šikonja, and Borštnar (2017a)	Integrating interpretable recommendations of machine learning models into knowledge-based decision-making processes	Black-box machine learning models
Bohanec, Robnik-Šikonja, and Borštnar (2017b)	Applying machine learning models and general explanation methodology for creating predictions as a basis for learning	Machine learning models; general explanation methodology
Reddick, Chatfield, and Ojo (2017)	Facilitating organizational learning and public policy-making through exploratory text analyses of government social media channels	Text mining; topic modeling
Reychav et al. (2016)	Using tablets in medical consultations to search for health related information and to improve shared decision-making	Knowledge-based system; mobile devices
Zhan and Zheng (2016)	Learning from incidents through the calculation of causal relationships among manually predefined classification of factors	Human Factor Analysis and Classification System (HFACS); causal analysis
Labib and Read (2015)	Applying a hybrid modelling approach that facilitates the understanding of failures and their consequences	Simulation modeling; failure mode effect and criticality analysis; fault tree analysis; AHP
Pérez-Bennett et al. (2014)	Promoting double-loop learning through specific case-based simulations as a pedagogical complement	Simulation modeling, system dynamics; case method
Rauffet, Cunha, and Bernard (2014)	Facilitating the transfer of good practices and their organizational reuse in groups by means of a digital learning system	Roadmapping approach; knowledge-based system
Kim et al. (2013)	Implementing simulation modeling as a way to examine multiple hypotheses in managerial decision-making practices	Simulation modeling; system dynamics; dynamic hypothesis testing
Ting et al. (2010)	Supporting knowledge sharing and the medical prescription process by providing peer-based comparisons based on statistical perspectives	Case-based reasoning; Bayesian theorem
Takahashi (2006)	Using simulation models for facilitating decision-making in a multi-layer hierarchical structure	Simulation modeling; agent-based organizational cybernetics
Weber et al. (2010)	Improving representation of lessons learned that highlights reuse conditions in the context of an intelligent lessons learned system	Knowledge-based system; knowledge modeling

and practical implementation. The proposed approach is based on the idea of discovering lessons hidden within the experience reports of completed projects in a first step (see also the following Section 2.2). Thus, potential success factors and sources of error in past project concepts and project management routines will be identified, and these may, in turn, reveal possible learning potential for future projects. In a second step, these lessons are modeled as basic project-governing variables in a summarizing multi-criteria decision network (Project Evaluation Network), which is then used for the mutual assessment of their impacts in future projects.

2.2. Codified knowledge in project environments

Documentation of past projects forms an important knowledge basis for the DLL process (von Zedtwitz, 2002). Accordingly, project-based organizations invest a lot of time and energy in the extensive documentation of their projects (Barclay & Osei-Bryson, 2010; Prencipe & Tell 2001). The so-called post-project reviews, which include a critical reflection of the successes and failures of completed projects, play an important role in this context (Disterer, 2002; von Zedtwitz, 2002). Such codified project knowledge is collected in organizational knowledge repositories and made available to subsequent projects so that they can learn from these historic lessons learned (Boh, 2007; Koners & Goffin, 2007). Such knowledge resources may have a central significance for a continuously learning organization (Almeida & Soares 2014). For example, the PMBoK® guide describes appropriate knowledge resources as an "organization process asset," which should be used for the sound design of future projects in particular as well as for the continuous development of project management skills (PMI, 2008). Several empirical studies have demonstrated that recourse to such historical project knowledge can have a significant positive impact on project success (see, *e.g.*, Hanisch, Lindner, Mueller, & Wald, 2009; Hong, Kim, Kim, & Leem, 2008; Kotnour, 2000; Kululanga & Kuotcha, 2008).

The knowledge codified in project documents can cover various topics. Several researchers have already studied the corresponding nature of project-based knowledge and tried to characterize its contents (see, *e.g.*, Chan & Rosemann, 2001; Kang & Hahn, 2009; Reich, Gemino, & Sauer, 2008). Although the concrete interpretations differ, comparable classifications can be summarized. Zhao and Zuo (2011) bundled the existing classifications and describe five main types of project knowledge (see Table 2): (1) *business domain knowledge*, (2) *project product knowledge*, (3) *project engineering (technical) knowledge*, (4) *organization management knowledge*, and (5) *project management knowledge*.

The regular analysis of codified project knowledge is specifically recommended by common project management guidelines (OCG, 2009; PMI, 2008), although no concrete guidance for the practical analysis of the extensive and mostly textual documentation is supplied. However, several proposals for such text analyses can be found in the scientific literature. Here, a fundamental distinction can be made between manual and computational approaches. Schalken, Brinkkemper, and Van Vliet (2006), for example, suggest a manual content analysis approach in order to extract meaningful lessons learned from a repository of textual project documents. Carrillo et al. (2011) and Choudhary et al. (2009), on the contrary, suggest computer-aided analyses to summarize large textual content in a partially automated fashion. Given the often extensive document repositories in project-based organizations (Prencipe & Tell 2001), these partially automated solutions are increasingly gaining importance (Grobelnik & Mladeni, 2005; King, 2009). This article, therefore also uses such a computer-aided text analysis (LSA) for extracting project knowledge. The five project knowledge categories (Table 2) introduced play an important role in this analysis, since they will give a thematic framework for the knowledge concepts extracted (see Section 4.1).

3. Conceptual framework, methodologies, and case study

3.1. Conceptual framework

At its core, the proposed DLL approach aims to evaluate future project concepts based on a set of historical project lessons. Specifically, this means that the project concept that is most likely to

Table 2
Classification of project knowledge (*cf.* Zhao & Zuo, 2011, p. 268).

	Business domain knowledge	Project product knowledge	Project engineering (Technical) knowledge	Organization management knowledge	Project management knowledge
Question	What lessons can be drawn from a business domain perspective?	What lessons can be drawn with regard to the project outcome itself?	What lessons can be drawn from a technical perspective?	What lessons can be drawn with regard to the management of organizations?	What lessons can be drawn with regard to the management of projects?
Description	Refers to knowledge about a companyś business context (*e.g.*, industry specifics, business processes, employees)	Refers to knowledge on how to select, develop, implement, or operate a specific business solution (*e.g.*, system customizations)	Refers to knowledge about the technical characteristics of a technology implementation (*e.g.*, data migration)	Refers to knowledge on how to coordinate the various stakeholders involved in complex projects (*e.g.*, relationship management)	Refers to knowledge on how to conceptualize, plan, coordinate, measure, and manage projects (*e.g.*, project team composition)

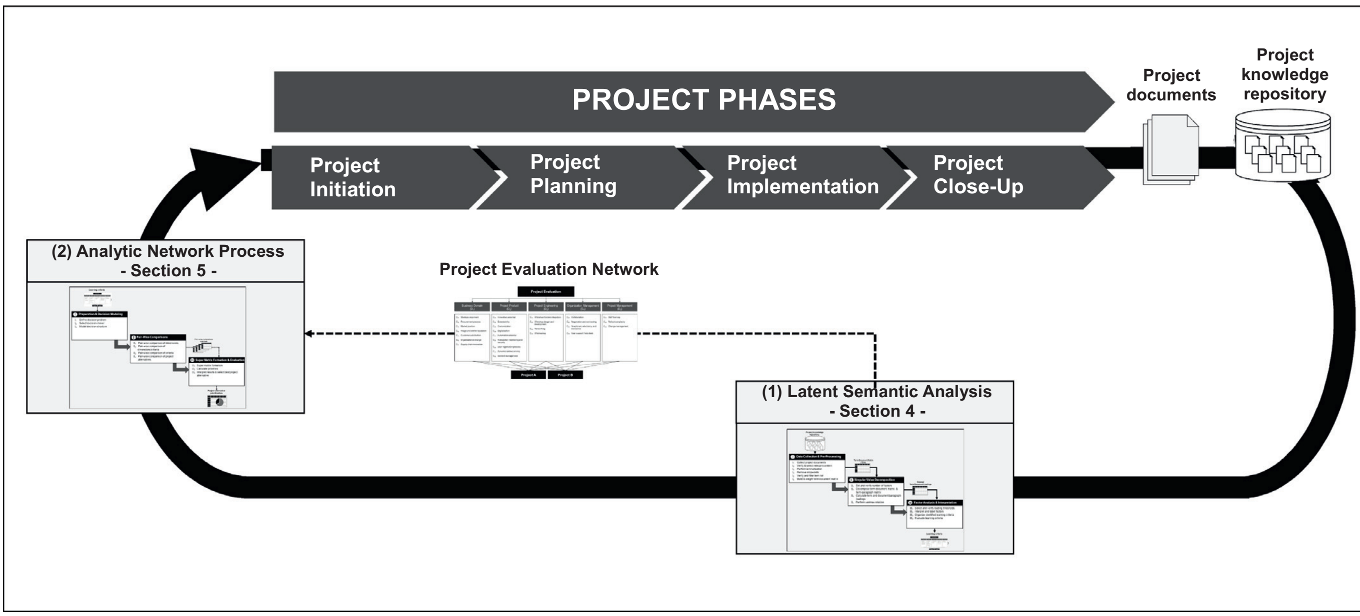

Fig. 1. Conceptual Framework: Double-Loop Learning in Project Environments.

be able to implement the lessons collected in previous projects – that is, implement success factors or address problem areas – is preferable in terms of a cross-project learning process. Furthermore, lessons that are not sufficiently addressed indicate possible learning potential and the need for modification of the project design or the established project management procedures.

The proposed implementation approach (see Fig. 1) represents an iterative routine in which future projects are evaluated, modified, and developed further based on a set of historic lessons learned. In this context, the approach integrates two complementary analyses. As part of the first analysis, central lessons learned which are necessary for later project evaluation, are extracted based upon project knowledge stored in organizational repositories. By this means, learning potential to be considered in future projects is identified. Instead of having these learning criteria manually collected and subjectively determined, this article pursues the idea that a more objective and complete set of criteria can be extracted using *Latent Semantic Analysis* (LSA). As part of the second analysis, the *Analytic Network Process* (ANP) is used for the systematic evaluation of project concepts. Here the previously extracted lessons learned are modeled as project-governing variables in a hierarchical decision or criteria network – the so-called *Project Evaluation Network* (PEN) – including their interdependencies. This decision network is the foundation for the solution of the multi-criteria decision problem. The partial analyses used, the case study, as well as the database will now be presented in more detail below.

3.2. Latent semantic analysis

Latent Semantic Analysis (LSA) is a computerized method for extracting thematic core concepts contained in textual document collections (for an introduction, see, *e.g.*, Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Dumais, 2004; Landauer, Foltz, & Laham, 1998). LSA has its roots in the fields of information retrieval and content analysis and, since its development in the 1980s, has enjoyed increasing applicability for semi-automated summarization of large textual databases (Evangelopoulos, Zhang, & Prybutok, 2012). LSA is based on the fundamental idea that each document collection consists of certain contexts (*i.e.*, documents, chapters, paragraphs, and sentences), and that each of these contexts is characterized by the presence or absence of certain words or word patterns. Based on this understanding, statistical analyses are able to extract thematic concepts by identifying frequently used word patterns (*i.e.*, correlating words regularly occurring in contexts) or other similar documents (*i.e.*, documents that use similar word patterns). LSA is mathematically based on the principles of factor analysis, *i.e.* specifically on *Singular Value Decomposition* (SVD), which resolves the dimensionality of large data sets (*i.e.*, document contents) with the help of factors. Each analyzed factor stands for a latent semantic relationship (*i.e.*, a thematic concept), which stands for correlating high-loading terms or high-loading documents. The precise analytical process of LSA is presented in greater detail in Section 4.1.

LSA is particularly suitable for the proposed implementation approach for two reasons. First, LSA allows for the mostly automated summarization of large repositories of project documents and the extraction of thematic core concepts (*i.e.*, in this study, lessons learned). Secondly, LSA is an exploratory analysis, which is able to identify unknown core concepts, without the analysts having to define specific issues or provide an analytical framework in advance (a priori). This ensures the objectivity and completeness of potentially relevant topics for the following project evaluation.

3.3. Analytic network process

The *Analytic Network Process* (ANP) was originally developed by Saaty (1996) for solving multi-criteria-decision-making (MCDM) problems. ANP distinguishes itself in that it is able to consider interdependent criteria in complex decision problems. For this purpose, the decision criteria are modeled in a hierarchical decision network, including their respective relationships and dependencies, then evaluated in a logical process, and finally analyzed based on a supermatrix (see, *e.g.*, Niemira & Saaty, 2004, for a practical application). The precise analytical process of ANP is presented in greater detail in Section 5.1.

ANP is especially well-suited for the evaluation of alternative project concepts for two reasons. First, projects are by definition complex exercises, in which many heterogeneous aspects must often be considered, such as economic, customer-oriented, technical, or organizational issues. ANP can be used to thematically structure and model such diverse aspects based on its hierarchical decision network. Secondly, building on that, projects are not static structures. In fact, a decision on one partial aspect often directly influences additional aspects of a project – positively as well as negatively. An example: The more complex and therefore challenging the organizational rollout of a new online shop is, the more complex is the required change management and training of the staff. However, ANP is able to analytically consider such interdependencies between individual project criteria. Due to these advantages mentioned, ANP has already been used for the evaluation of project alternatives (see, *e.g.*, Hsu & Hu, 2009; Lee & Kim, 2000; Meade & Presley, 2002).

3.4. Case study and data collection

The proposed implementation approach will be demonstrated in a practical case study. In this case study, two alternative project concepts from the perspective of a medium-sized online retailer are analyzed: first, the *upgrade* of an existing, internally-developed online shopping system with associated system architecture and second, the acquisition and implementation of a *new solution* from an external specialist provider. These two project alternatives will now be presented in more detail.

Upgrade: The first project alternative involves the extensive upgrade of the existing online shopping system and corresponding system architecture. The existing solution was mostly self-developed, is integrated into the enterprise system landscape (*e.g.*, as in product databases), and is validly integrated into the organizational processes (*e.g.*, payment processes). Nevertheless, extensive improvements have to be carried out by internal IT developers in order to meet modern requirements (content management, usability, recommender system).

New solution: The second project alternative involves the purchase of an entirely new online shopping solution from an external specialist supplier and its integration into the existing system architecture and organizational processes. The acquisition costs and the integration and conversion expense of this alternative are significantly higher, but it provides access to state-of-the-art solutions (*e.g.*, in the area of online product presentation and live support system).

For the purposes of LSA, a thematically relevant and at the same time large textual database is required. The document collection of this demonstration includes 78 practical post-project reviews, which describe the design, development, and implementation of online shopping solutions. These documents follow a standardized format and describe the project background and design of the project as well as the objectives and results. In addition, a final reflection on the lessons learned in the projects is included. As such descriptions of lessons learned are particularly relevant for the implementation approach presented, only these final reflective sections ($N = 78$) were used for the following analyses. This contributes significantly to the validity of the analysis.

4. Modeling a project evaluation network using LSA

4.1. Analytical process and application

With LSA, lessons learned should automatically be extracted from the relevant project documentation. The LSA study is based on a three-step analytical process (see context diagram in Fig. 2): starting with data collection and pre-processing (I), followed by the statistical Singular Value Decomposition (II), and the final factor analysis and interpretation (III). The respective phases and tasks will be presented in more detail in the context of the demonstrative case study. For a detailed presentation of LSA and its mathematics, see *e.g.* Dumais (2004), Landauer et al. (1998), or Müller, Schmiedel, Gorbacheva, and vom Brocke (2016). The text mining solutions used were *Provalis Research's* QDA Miner and WordStat 7.1.

(I) Data collection and pre-processing

(I_1) *Collect project documents*: First, a collection of project documentation potentially relevant for the extraction of project lessons was to be compiled. The post-project reports ($N = 78$) presented in Section 3.4 were therefore used for this case study. All documents were transferred into a plain text format for further processing.

(I_2) *Select and verify relevant content*: The document collection should then be checked for completeness, relevance, and quality of content. In particular, irrelevant documents or document content can lead to statistical distortions in the subsequent statistical analyses. Since the final descriptions of lessons learned are of particular interest for the proposed approach, exclusively the respective lessons learned sections of the reports were selected. General company descriptions and technical details are irrelevant and were excluded from the LSA for the purposes of validity.

(I_3) *Perform lemmatization*: In the course of a subsequent pre-processing, the unstructured text contents were transferred into a structured and numerical format (see Manning & Schuetze, 1999; Miner et al., 2012). In the first step, using lemmatization, different word variations were traced back to their grammatical base forms via extensive dictionaries (*e.g.*, “analyzing,” “analyzes,” and “analyzed” to “analyze”). In this way, the textual database (a list of 3220 unique words) could be reduced to a word or term list with a total of 2209 unique words.

(I_4) *Remove stop-words*: Using so-called stopword removal, trivial words (such as “and,” “or,” and “in”) were removed from the database. This process further reduced the term list to a total of 1923 unique words.

(I_5) *Filter and verify term list*: The term list should then be evaluated and further specified. The aim is that the final word collection reflects the issues to be examined in content as accurately as possible, that is, from an analytical point of view, with as little statistical noise as possible. In a first step, therefore, words with

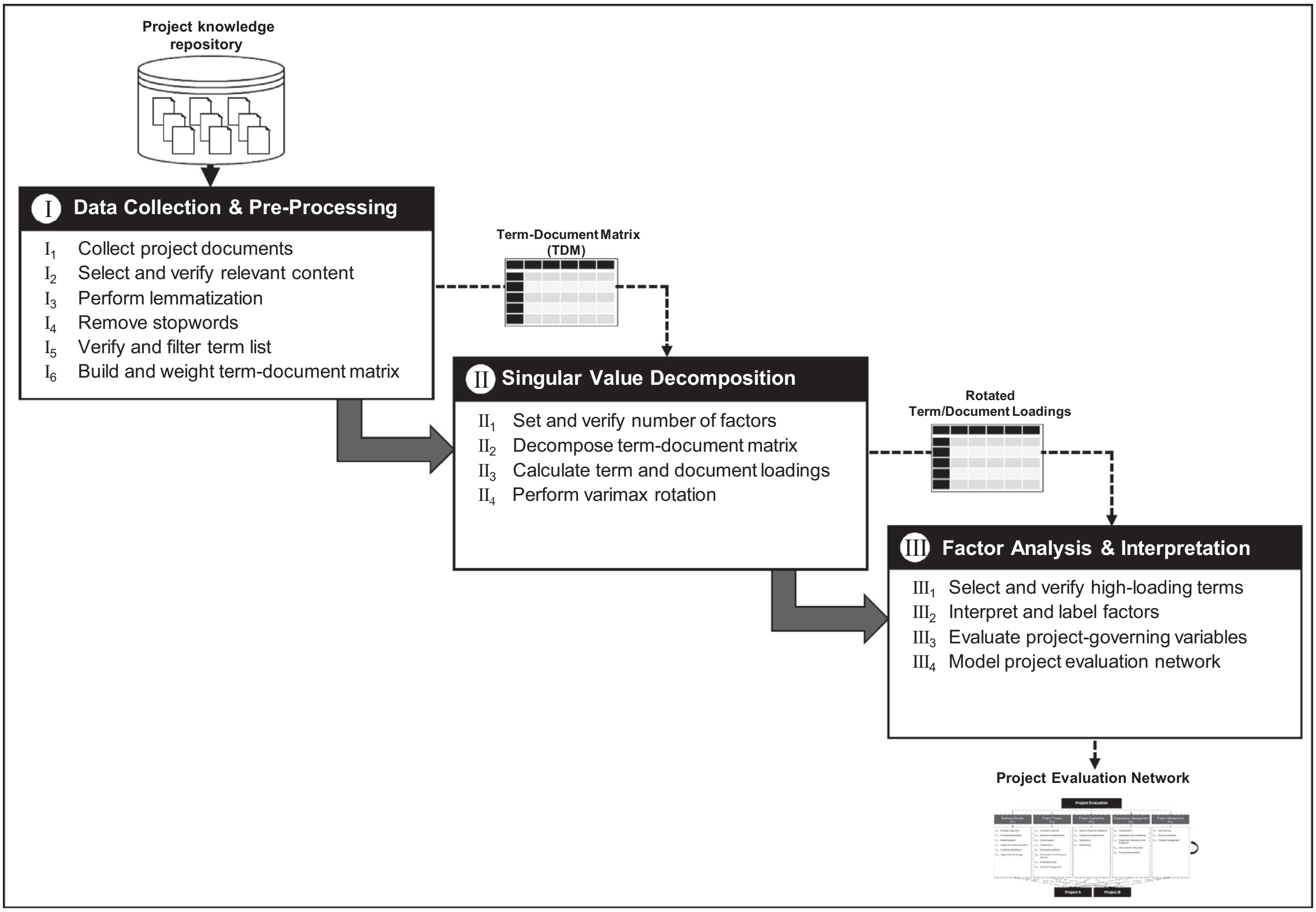

Fig. 2. Phases and Tasks of the Latent Semantic Analysis (LSA).

an occurrence frequency of less than 2 were filtered out throughout the document collection. Such words do not contribute to the identification of statistical patterns. This filtering process reduced the term list to 1017 unique words. Next, the reduced list was manually checked by two independent analysts for substantive significance and correctness (*e.g.*, misspellings). Irrelevant terms (*e.g.*, proper nouns, pronouns, adverbs, and prepositions) were filtered out in this step as well. Because this manual process included an element of subjectivity, the two independently specified lists were checked for sufficient agreement (84% in terms of intercoder reliability) and deviations revised according to a consensus. Finally, a specified term list with 819 unique words was available for the LSA.

(I_6) *Build and weight term-document matrix*: The final term list was then transferred to a *Term-Document Matrix* (TDM). In this matrix, the documents contained in the collection are plotted in the columns, the contained terms in the rows, and the respective term frequencies in the cells. To give the terms or, more specifically, the term frequencies a more representative weight, a so-called TF-IDF transformation was applied (Evangelopoulos et al., 2012; Sidorova et al., 2008). The TF-IDF weighting multiplies the absolute frequency of a term in each document (TF = term frequency) with the inverse frequency of the term throughout the document collection (IDF = inverse document frequency), which in addition to the simple term frequency also considers the regularity of a term included in the document collection. Ultimately, rarer terms receive a higher weight and more frequent, ordinary terms a lower weight. The corresponding weighted TDM is ultimately the structured, quantitative database for the subsequent *Singular Value Decomposition* (SVD).

(II) Singular value decomposition (SVD)

(II_1) *Set and verify number of factors*: In the first step of the SVD, the number of factors must be established. A smaller number of factors corresponds to a higher level of abstraction and will reveal general core subjects. A greater number of factors corresponds to a more detailed degree of abstraction and will thus cover more specific thematic concepts. In this regard, Evangelopoulos et al. (2012) recommend testing the significance of varying numbers of factors iteratively as part of test runs. In addition, the implementation of a scree test may assist in the identification of a statistically meaningful number of factors. In the present analysis, a number of factors of 30 was considered significant after several test runs. A greater number of factors would reveal no further meaningful concepts; a lower number, however, would exclude relevant concepts from the analysis. The factor number of 30 was then also statistically confirmed by a final scree test.

(II_2) *Decompose term-document matrix*: Along the 30 factors, the weighted TDM was then decomposed using SVD; that is, individual matrices were determined with eigenvectors of the terms and documents, as well as a diagonal matrix of singular values (*i.e.*, the square root of the common eigenvalues between terms and documents).

(II_3) *Calculate term and document loadings*: The specific term and document loadings of the individual analyzed factors were then calculated.

(II_4) *Perform varimax rotation*: Then the factor loadings were subjected to a varimax rotation, as recommended by Sidorova, Evangelopoulos, Valacich, and Ramakrishnan (2008). Such a factor rotation makes the individual factor loadings more concise, without distorting their statements. In this way, an even clearer separation of high-loading or low-loading terms is possible for each factor. A set of significant high-loading terms per factor here stands for an extracted thematic concept (*i.e.*, a content-correlating word bundle). The rotation thus simplifies the subsequent interpretation of factors in phase III.

(III) Factor analysis and interpretation

(III_1) *Select and verify high-loading terms*: A meaningful set of high-loading terms then had to be identified per explored factor. Various procedures are conceivable for such a limitation, for example, the fixing of a specific loading thresholds, *i.e.* a certain statistical threshold to which term or document loadings may be assigned to a factor (see Evangelopoulos et al., 2012). In this analysis, the proposed *k*-factors solution proposed by Sidorova et al. (2008) was used to determine the threshold, in which the top-$1/k$ high-loading terms of a factor are selected. This means, accordingly, the top 3.3 per cent of high-loading terms (that is, 27 of a total of 819 terms) per factor were extracted with the 30-factor solution used. Theoretically, this means that on average each term can be high loaded for one factor.

(III_2) *Interpret and label factors*: After application of the thresholds, the high-loading terms (*i.e.*, correlating word bundle) from the individual factors were interpreted as thematic concepts (*i.e.*, lessons learned) and provided with specific labels (see Table A.1 in the Appendix). One example: The factor C_{53} bundles the high-loading, *i.e.* correlated terms "change," "reluctancy," "qualification," "user," "lead," "motivate," "training," "help," "describe," "solve," and "problem," which can in combination be interpreted and labeled as the issue of "Change Management." Since this is a subjective process, it was advisable to have this completed in parallel by several independent analysts and then to evaluate their agreement (in terms of an intercoder reliability). After appropriate examination and revision, ultimately 26 concrete lessons learned were able to be identified from the original 30 factors. Two factors could not be unambiguously thematically interpreted and two other factor couples described the same thematic issue.

(III_3) *Evaluate project-governing variables*: In the final step, the identified lessons learned were examined and confirmed by two experts (web shop developer and e-commerce manager) as relevant project-governing variables, which thus represent the underlying criteria for the subsequent evaluation of project concepts. Over the course of this evaluation, only the labels of these governing variables were specified.

(III_4) *Model project evaluation network*: To give the 26 thematically heterogeneous project-governing variables a structured framework, they were organized in a final step along the five categories of project knowledge (see Section 2.2). Finally, a structured criteria model, the so-called *Project Evaluation Network* (PEN), for the assessment of project concepts was modeled (see Fig. 2 in Section 4.2).

4.2. Results

The modeled *Project Evaluation Network* (see Fig. 3) is the basis for the evaluation of two alternative project concepts (P_1 and P_2). This decision model consists of five thematic dimensions (D_1–D_5) with a total of 26 criteria (C_{11}–C_{53}). The dimensions and decision criteria will be presented in more detail below.

Business Domain (D_1): This dimension includes decision criteria with respect to fundamental business backgrounds and characteristics. Criterion C_{11} can be interpreted as the importance of a strategic alignment of the desired solution. Additional criteria related to the company are the influence of the solution on the organizational procurement process (C_{12}), the market position (C_{13}), the company image (C_{14}), and customer satisfaction (C_{15}). Another criterion is the organizational change (C_{16}) initiated by the respective project concept.

Project Product (D_2): This dimension summarizes the characteristics of the project product to be evaluated (*i.e.*, the online shopping solution). This includes the innovative potential of targeted solutions (C_{21}), their expandability and upgradeability (C_{22}), and their respective capabilities in terms of customization (C_{23}), digitalization (C_{24}), automation (C_{25}), transaction monitoring and security (C_{26}), and content management (C_{28}). The associated investment costs should also be evaluated (C_{27}).

Project Engineering (D_3): This dimension includes decision criteria with regard to technical aspects. In this context, criteria such as the ability to integrate the solution into existing system environments (C_{31}), the respective design and development efforts (C_{32}), the networking capability (C_{33}), and the technical webhosting (C_{34}) are identified.

Organization Management (D_4): This dimension describes the organizational aspects of the project, such as the need for collaboration with other actors (C_{41}), the requirements in terms of negotiation and contracting (C_{42}), the skepticism and resistance implied by the respective solution (C_{43}), the relevance of well-founded support of the users (C_{44}), and the implied coordination efforts in the procurement process (C_{45}).

Project Management (D_5): This dimension includes decision criteria related to the actual management of the projects. In this regard, the scope of the required staff training (C_{51}), the complexity of the rollout (C_{52}), as well as the relevance of a professional change management (C_{53}) were identified as relevant.

5. Evaluating alternative project concepts using ANP

5.1. Analytical process and application

The PEN was used in a subsequent analysis to evaluate the two alternative project concepts and also to identify potential need for modification. The ANP study used for this purpose is based on a three-step analytical process (see context diagram in Fig. 4): from the definition and tool-aided modeling of the decision problem (I), to the pair-wise assessment of decision criteria and project alternatives (II), on to the final evaluation and prioritization of the alternative project concepts based on a supermatrix formation (III). The phases and working steps are discussed in more detail below. For a detailed description of the ANP application and its practical working steps see, *e.g.*, Hsu and Hu (2009).

(I) Preparation and decision modeling

(I_1) *Define decision problem*: The decision problem must first be defined. In the present case, the assessment and evaluation of alternative project designs is the focus of attention. The project concept that best meets the previously extracted evaluation criteria (*i.e.*, project-governing variables) should be prioritized in respect of the learning effects collected in previous projects. The decision problem can therefore be formulated as: *Evaluation and prioritization of the most desirable project concept.*

(I_2) *Select decision-maker*: Conducting an ANP study requires complex assessments with varying aspects. Sound knowledge of the project issues is therefore required for professional evaluation. In this case, the corresponding assessments were carried out by three experts in the field of e-commerce: one of the authors of this

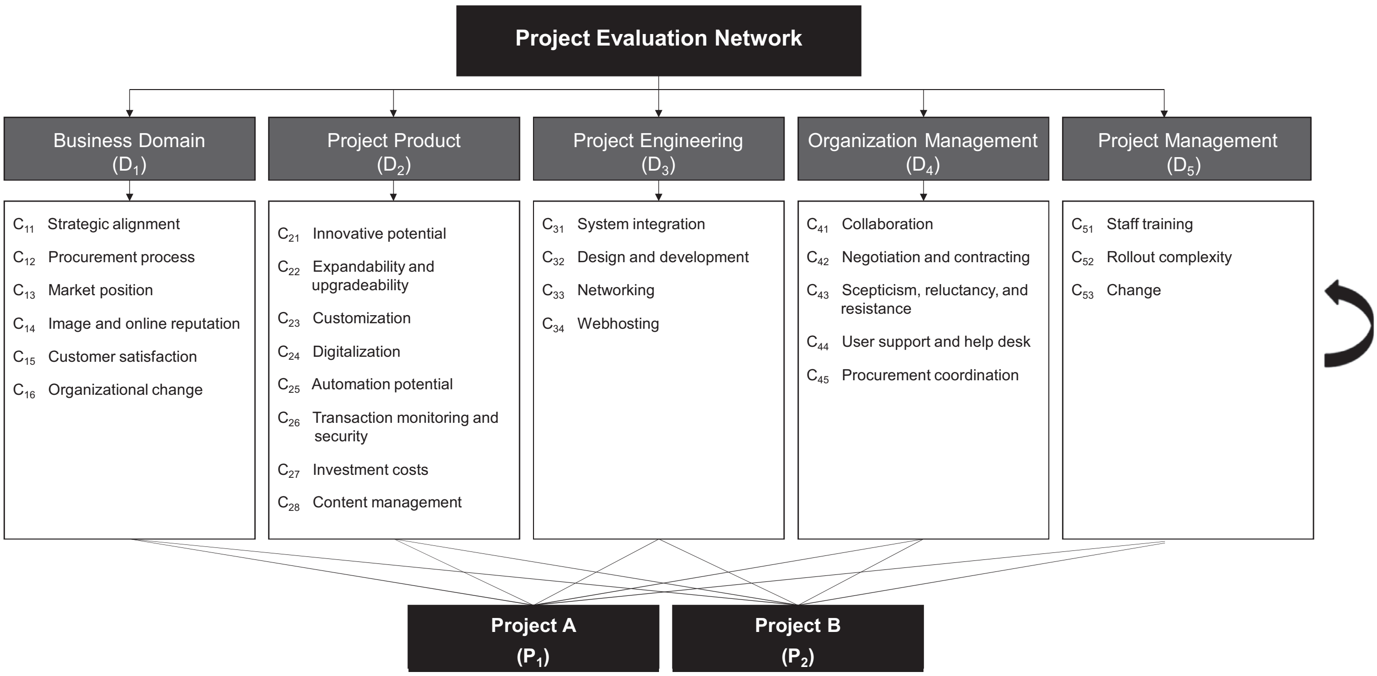

Fig. 3. Project Evaluation Network (PEN).

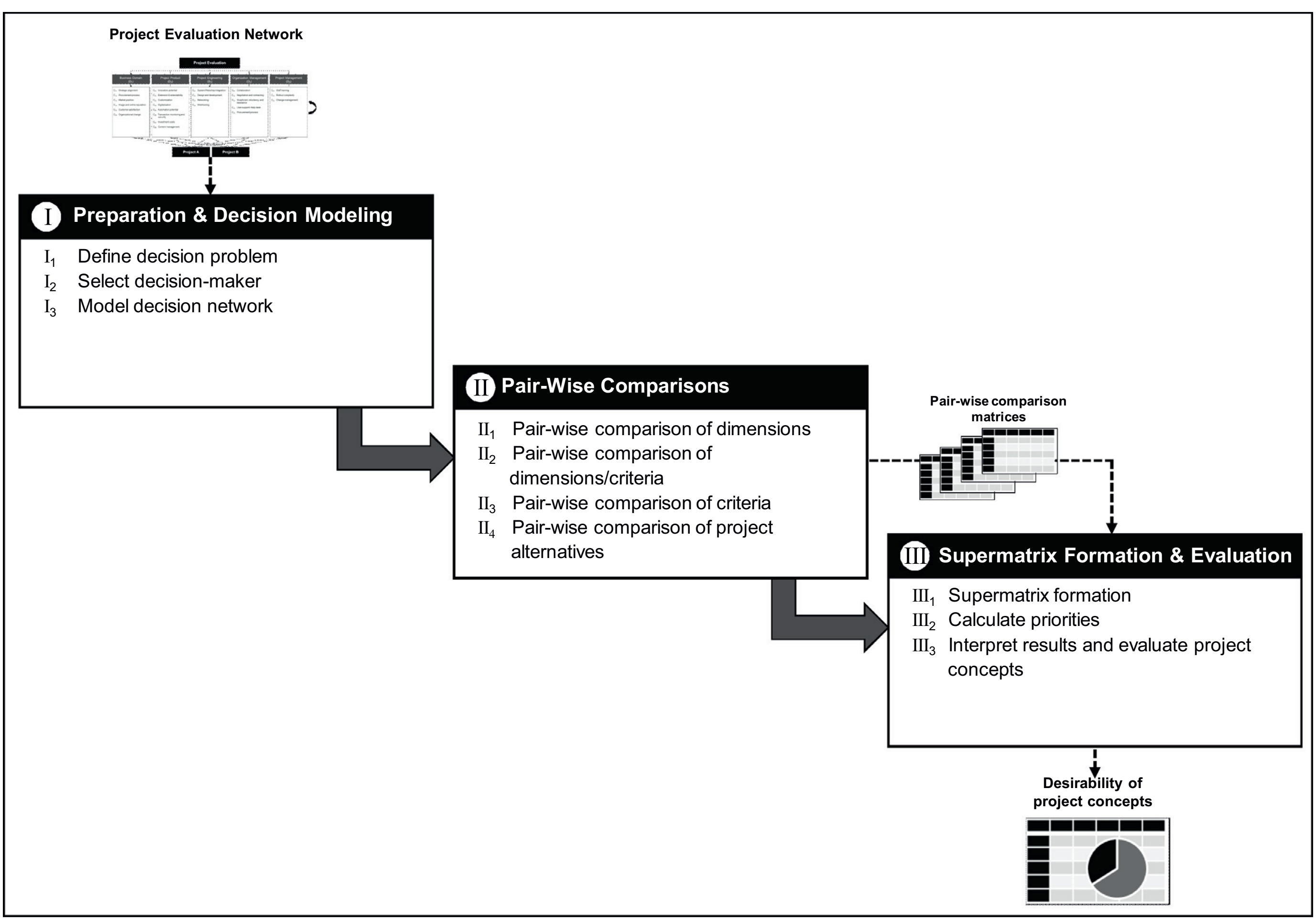

Fig. 4. Phases and Tasks of the Analytic Network Process (ANP).

Table 3
Priorities of dimensions (D_1–D_5).

	D_1	D_2	D_3	D_4	D_5	e-vector
D_1	1.0000	2.0000	4.3333	3.3333	4.6667	0.4063
D_2	0.5000	1.0000	4.0000	3.6667	4.3333	0.3093
D_3	0.2308	0.2500	1.0000	0.2727	1.6667	0.0744
D_4	0.3000	0.2727	3.6667	1.0000	2.0000	0.1473
D_5	0.2143	0.2308	0.6000	0.5000	1.0000	0.0627

Table 4
Priorities of dimensions/criteria (C_{11}–C_{16}).

	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	e-vector
C_{11}	1.0000	2.3333	3.6667	0.7500	5.0000	2.6667	0.2914
C_{12}	0.4286	1.0000	2.3333	0.6000	3.6667	2.3333	0.1815
C_{13}	0.2727	0.4286	1.0000	0.4286	2.3333	1.3333	0.1010
C_{14}	1.3333	1.6667	2.3333	1.0000	4.3333	2.6667	0.2719
C_{15}	0.2000	0.2727	0.4286	0.2308	1.0000	0.2308	0.0459
C_{16}	0.3750	0.4286	0.7500	0.3750	4.3333	1.0000	0.1083

Table 5
Priorities of Criteria (C_{12}–C_{16}) with respect to strategic alignment (C_{11}).

	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	e-vector
C_{12}	1.0000	5.6667	0.6000	5.3333	2.3333	0.3349
C_{13}	0.1765	1.0000	0.3750	4.6667	0.4286	0.1092
C_{14}	1.6667	2.6667	1.0000	4.6667	2.6667	0.3456
C_{15}	0.1875	0.2143	0.2143	1.0000	0.2500	0.0463
C_{16}	0.4286	2.3333	0.3750	4.0000	1.0000	0.1640

article, as well as two experts of the practice (a web-developer and an e-commerce manager). The quantitative assessments of the individuals were pooled and were included in the evaluations as the calculated mean value.

(I_3) *Model decision network*: The decision problem must be modeled into a hierarchical decision network, including all the relationships and interdependencies of the decision criteria to be evaluated. The Project Evaluation Network (PEN) extracted by the LSA was modeled as such a decision network using the ANP software *Super Decision*.

(II) Pair-wise comparisons

(II_1) *Pair-wise comparison of dimensions*: The ANP analysis is inherently based on the pair-wise comparisons of the underlying decision dimensions and criteria. In an initial pair-wise comparison, the decision makers had to assess the overarching decision dimensions (D_1 – D_5) against one another in order to determine the relative weights (*i.e.*, priorities) of each dimension. In this context, the underlying question was: Which decision dimension has greater relevance for the overall decision problem (= *Evaluation and prioritization of the most desirable project concept*)? The pair-wise comparison of any two dimensions is based on a predetermined scale of 1–9 (“1”: equal importance of dimensions D_a and D_b; “9”: overwhelmingly stronger importance of dimension D_a) or 1/9–1 (“1/9”: overwhelming dominance of dimension D_b; “1”: equal importance of dimensions). The independent evaluations of the three decision makers were summed and calculated into a mean value. A reverse comparison of the corresponding dimensions must always be logically consistent, *i.e.*, reciprocally conducted. Thus, for example, the Dimension D_2 is strongly more important than D_3 (evaluation: 4.000) and Dimension D_3 is in turn reciprocally less important (0.2500). After the pair-wise comparisons, the opinions from the comparison matrix are converted per dimension into relative priorities (so-called eigenvectors or e-vectors) and then normalized, so that the individual priorities of the dimensions result in a total of 1.0 or 100% (see Table 3). Thus, for example, Dimension D_1 (Business Domain) has the highest priority (0.4063) for the evaluation of the decision problem and Dimension D_5 (Project Management) the lowest (0.0627).

(II_2) *Pair-wise comparison of dimensions/criteria*: In the next step, additional pair-wise comparisons were carried out, in which the decision criteria (C_n) were evaluated against one another in relation to their relevance for the overall decision dimension (D_n). The underlying question here was: What is the relevance of the decision Criterion C_a for the overall decision Dimension D_n compared to the Criterion C_b? These pair-wise comparisons also resulted in a comparison matrix, which reflects the priorities of the decision criteria for each dimension using the relative, normalized weights (Table 4). Thus, for example, Criteria C_{11} (Strategic Alignment) and C_{14} (Image and Online Reputation) have the highest priorities (e-vectors) with regard to the aspects of the business domain (0.2914 and 0.2719).

(II_3) *Pair-wise comparison of criteria*: Other pair-wise comparisons are aimed at evaluating the internal relations of the criteria among one another. Here, the criteria of a dimension are evaluated as to how they demonstrate influence on a related control variable. Should, for example, a decision dimension receive 4 criteria, four pair comparisons for each criterion are carried out, each with a changing control variable. The underlying question is: Looking at the control criterion C_c and its influence on the overall decision dimension D_n, how great is the influence of Criterion C_a on the Control Criterion C_c when compared to Criterion C_b? The pair-wise comparisons also resulted in a comparison matrix with priority weightings for each of the criteria (see an exemplary evaluation in Table 5).

(II_4) *Pair-wise comparison of project alternatives*: In the last required pair-wise comparison, the project alternatives were evaluated in terms of their respective advantageousness toward the individual decision criteria. In this context, the underlying question was: Which project alternative (P_n) is more advantageous with respect to the considered decision criterion C_n? A relative priority weighting of the project alternatives per decision criterion appears in Table 6 (see columns P_{1cj} and P_{2cj}).

(III) Supermatrix formation and evaluation

(III_1) *Supermatrix formation*: A supermatrix is used to analytically synthesize the interdependencies of the decision system. The supermatrix processes the sub-matrices previously created in step II_3, in which the interdependencies of decision criteria were calculated for each dimension (see, *e.g.*, Table 5). As 26 criteria are processed in the decision network, the supermatrix thus contains 26 columns with the weightings of pair-wise evaluations of interdependencies (see Table A.2 in the Appendix). The supermatrix then went through another convergence process (see Hsu & Hu, 2009) to produce stable overall priority weightings of each criterion for each column (see Table A.3 in the Appendix). To achieve the convergence of the supermatrix, it must be column stochastic, *i.e.*, the sums of the columns of the interdependent relationships must result in 1. To this end, the supermatrix is multiplied with itself (a so-called “power method”) until the column values converge and stabilize, so that the priority weightings are finally identical (see also Niemira & Saaty, 2004).

(III_2) *Calculate priorities*: In the next step, the priorities of the project alternatives A and B were evaluated on the basis of the summarizing desirability indices (see also Hsu & Hu, 2009):

$$D_i = \sum_{j=1}^{J} W_j A^D_{cj} A^I_{cj} P_{icj}$$

The desirability index D_i is calculated for each project alternative (j) through the summary of several partial weightings: W_j is the relative importance weight, *i.e.* the priority of the individual

Table 6
Calculation of desirability indices per project alternative.

Dimension	W_j	Criteria	A^D_{cj}	A^I_{cj}	P_{1cj}	P_{2cj}	Project A	Project B
D_1	0.4063	C_{11}	0.2914	0.1960	0.7692	0.2308	0.0179	0.0054
	0.4063	C_{12}	0.1815	0.1814	0.8125	0.1875	0.0109	0.0025
	0.4063	C_{13}	0.1010	0.1770	0.7857	0.2143	0.0057	0.0016
	0.4063	C_{14}	0.2719	0.2425	0.7857	0.2143	0.0210	0.0057
	0.4063	C_{15}	0.0459	0.0777	0.1667	0.8333	0.0002	0.0012
	0.4063	C_{16}	0.1083	0.1254	0.8500	0.1500	0.0047	0.0008
D_2	0.3093	C_{21}	0.2936	0.1433	0.8800	0.1200	0.0115	0.0016
	0.3093	C_{22}	0.1096	0.1478	0.8125	0.1875	0.0041	0.0009
	0.3093	C_{23}	0.1532	0.1818	0.1765	0.8235	0.0015	0.0071
	0.3093	C_{24}	0.0500	0.1407	0.7500	0.2500	0.0016	0.0005
	0.3093	C_{25}	0.0391	0.1142	0.7692	0.2308	0.0011	0.0003
	0.3093	C_{26}	0.0716	0.0689	0.8636	0.1364	0.0013	0.0002
	0.3093	C_{27}	0.2295	0.0956	0.1765	0.8235	0.0012	0.0056
	0.3093	C_{28}	0.0534	0.1077	0.7692	0.2308	0.0014	0.0004
D_3	0.0744	C_{31}	0.4988	0.3708	0.1765	0.8235	0.0024	0.0113
	0.0744	C_{32}	0.3153	0.3113	0.8125	0.1875	0.0059	0.0014
	0.0744	C_{33}	0.1285	0.2154	0.7500	0.2500	0.0015	0.0005
	0.0744	C_{34}	0.0574	0.1026	0.7273	0.2727	0.0003	0.0001
D_4	0.1473	C_{41}	0.4253	0.2866	0.8125	0.1875	0.0146	0.0034
	0.1473	C_{42}	0.1829	0.2411	0.2308	0.7692	0.0015	0.0050
	0.1473	C_{43}	0.0996	0.2084	0.2308	0.7692	0.0007	0.0024
	0.1473	C_{44}	0.0958	0.0527	0.8125	0.1875	0.0006	0.0001
	0.1473	C_{45}	0.1964	0.2111	0.7000	0.3000	0.0043	0.0018
D_5	0.0627	C_{51}	0.1307	0.2727	0.2000	0.8000	0.0004	0.0018
	0.0627	C_{52}	0.6060	0.4545	0.1579	0.8421	0.0027	0.0145
	0.0627	C_{53}	0.2633	0.2727	0.2000	0.8000	0.0009	0.0036
Desirability indices D_i							0.1200	0.0798
Normalized desirability indices D_{iN}							0.6006	0.3994

dimensions (determined by the pair-wise comparison of dimensions in step II_1); A^D_{cj} is the relative importance weight of criteria *(c)* in relation to the respective dimension *D* (determined by the pair-wise comparison of dimensions/criteria in step II_2); A^I_{cj} is the stabilized relative importance of the criteria *(c)* per dimension and while taking into account the interdependencies *(I)* between the criteria (determined by the supermatrix in Table A.3); P_{icj} is the relative advantageousness, *i.e.*, priority of the respective project alternatives *(j)* with respect to each of the criteria (determined by the pair-wise comparison of project alternatives in step II_4). Considering these individual relative weights, weighted desirability per criterion can be calculated per project concept (A and B), which results in the sum of the D_i normalized D_{iN} respective advantageousness degree across all criteria (see Table 6). An illustrative example: The specific priority of the criterion C_{11} (0.0179) appears for project alternative A by multiplying the individual relative weights ($0.0179 = 0.4063 \times 0.2914 \times 0.1960 \times 0.7692$). The summation of all these sub-priorities in column "Project A" then results in turn in the summarizing desirability index D_i (0.1200) or normalized in the index D_{iN} (0.6006) for project alternative A.

(III_3) *Interpret results and evaluate project concepts*: The calculated desirability indices represent the ultimate interpretable basis for decisions about the project concept to be favored as well as the identification of potential modification requirements (see Section 5.2).

5.2. Results

The results of the ANP study are summarized in Table 6. Here, the project alternatives A and B are prioritized based on their normalized desirability index D_{iN} with regard to the underlying decision problem *(evaluation and prioritization of the most desirable project concept)*. A higher desirability index means that the appropriate project alternative is most likely to be able to fulfill the project-governing variables. Accordingly, the project alternative A in its entirety is more advantageous (0.6006). The specific desirability measures of the individual sub-criteria can be also used for a more detailed critical look at the project concepts. The individual desirability per dimension and criterion are thereby shown in the rows of columns "Project A" and "Project B." In this case, given a direct comparison of the alternatives A and B, it becomes clear that project A fulfills criterion C_{11} (Strategic Alignment) significantly better, whereas the criterion C_{52} (Rollout Complexity) is better served by project B. It also becomes clear, for example, that project alternative A reveals potential deficits in criterion C_{23} (Customization). Such an engagement with decision-making criteria and their fulfillment by the project alternatives considered thus provides starting points for the identification of potential modification requirements and therefore for the further development of project concepts. In this case, then, the results suggest a need for further investigation of solutions that help to overcome the shortcomings of project alternative A. Among other things, the reevaluation of the system rollout process (C_{52}) and the precise definition of the catalog of requirements (C_{23}) are examples in this respect.

6. Discussion

The aim of this study was to help project-based organizations to increase their learning capabilities by integrating the support of expert tools. For this purpose, the study provides an implementation approach for DLL in project environments and showcased its application in a practical case study. The proposed approach addresses two concrete problems of the project management practice in this context. First, the information overload in project environments is addressed, in which a solution is supplied with the LSA for the semi-automated exploration of lessons learned from large repositories of textual project documentation. Second, the problem of the lack of procedures and methods for the practical implementation of historical project knowledge is addressed, in which a solution is supplied with ANP for the systematic modeling of extracted knowledge and its integration into the evaluation and further development of new projects. The results of the case study demonstrate that by these two complementary techniques, the proposed approach contributes to a practical implementation of DLL in project environments. On the one hand, the knowledge discovery in multiple knowledge sources can be facilitated to a large extent. On the other hand, the approach offers human experts the opportunity to systematically assess and prioritize project-governing variables in a multi-criteria decision-making model and thereby to judge alternative project concepts with regard to potential learning.

The proposed implementation approach contributes to the existing body of literature on organizational learning by addressing three evident opportunities for advancing the DLL process, which were largely overlooked in previous research. First, a contribution is the exploration of new and significant lessons learned hidden in the repositories of textually codified experience reports, which would replace the mere exploitation of the historic data sources for the explanation of already known issues of the past. A central advantage of the proposed approach is, therefore, the potential of uncovering unknown opportunities for the reevaluation of existing policies and practices. Second, a further contribution is the proposal of a technique that is capable of combining and synthesizing multiple knowledge sources (*i.e.*, documents, in this case). Such a summarization of multiple contextually related cases provides more representative input for decision-making than the isolated interpretation of single cases. Third, a further novelty is the integration of the ANP into DLL tasks. The complexity of the ANP application remains accessible for project managers and yet facilitates the assessment of project-governing variables and their complex relationships and mutual dependencies. The consideration of such a network of relationships between variables is an evident lack in previously proposed DLL approaches.

The implications for research are primarily related to the further development and evaluation of the proposed approach. Four distinct research avenues for advancing the approach may be distinguished. First, the precision of the lessons learned extraction from textual databases can be improved. This could be accomplished in two different ways. One is the preparatory categorization of lessons learned sections through text classification algorithms in order to improve the thematic accuracy of the underlying textual database. An idea in this regard could be the initial separation into positive and negative lessons learned (see, for example, Ur-Rahman & Harding, 2012). Another improvement possibility is the development and incorporation of ontology-based knowledge structures, which conceptualize the domain knowledge to be explored. Such ontology-based databases can play a vital role for storing, sharing, and retrieving specific knowledge contents (see Zhang, Yoshida, & Tang, 2009) and can, therefore, improve the extraction of lessons learned related to a specific topic of interest (*e.g.*, system integration problems); this, in turn, improves the thematic precision of LSA results.

Second, the further processing of the results in simulations and predictive models opens up another window for enhancing the current approach. Such techniques can be of benefit in assessing the impacts of forthcoming changes in policies and practices with respect to the DLL idea (see García, Román, Peñalvo, & Bonilla, 2008). In particular, it could be of great interest for project managers to estimate the impacts on the performance of project concepts (*e.g.*, cost, time, or outcome). For this purpose, several approaches are conceivable in the field of project management. One option is the integration of the results into cost contingency estimation models (see Idrus, Nuruddin, & Rohman, 2011) which predict potential cost overruns. Further options are the modeling of the results into scenario analyses or their integration into Monte Carlo simulations (Chou & Tseng, 2011). The compatibility with the existing approach should be tested in this context. Appropriate tests should be carried out in a practical project environment.

Third, the valid extraction and clear interpretation of lessons learned can be a critical aspect. Therefore, further research is needed with regard to computerized text analysis for the discovery of lessons learned from project documentation. For example, the use of alternative text analysis methods, such as the Latent Dirichlet Allocation, or complementary analyses, such as keyword density analyses, could be tested and compared. In this context, an evaluation of the limitations mentioned below should take place (*e.g.*, influences of the databases and methodologies used).

Fourth, the DLL concept itself can be extended into a so-called triple-loop learning concept (see McClory, Read, & Labib, 2017). Triple-loop project learning means that the learning process itself is subject to a continuous reevaluation in terms of learning targets, procedures, and outcomes. Future research should examine ways to operationalize this idea through system support. The periodic reevaluation of the underlying knowledge database is of special interest here in order to continuously assure the extraction of valid lessons learned from a high-quality knowledge base. Therefore, the proposed implementation approach could be accordingly extended by incorporating continuous reevaluation routines of the underlying database.

The proposed approach is not without limitations. First, limitations arise in relation to the database used, consisting of a collection of practical project reports. The content of corresponding experience reports does not necessarily have to be a complete and trustworthy source of all relevant project experiences. The deliberate suppression of negative situations (*e.g.*, problems and errors), because of feared sanctions, is a well-known phenomenon in such reports (see Cheng, Schulz, & Booth, 2009). Furthermore, a large number of documents is required for the sound application of the LSA. In smaller organizations, however, the compilation of such large document collections might not be possible in some circumstances. Second, the interpretation of potential lessons learned is deemed to have a critical influence on the success of organizational learning. However, the interpretation of the extracted thematic concepts was sometimes difficult. Improving the thematic precision of the underlying textual database may be an approach for receiving more clear and interpretable results in the course of the LSA. Other limitations arise in regard to the subjective influences within the analytical techniques used. Thus the LSA requires various decisions of the analyst as part of the data preparation (term filtering) and the setting of the analysis parameters (*e.g.*, determining the number of factors and the thresholds). Likewise, ANP requires a largely subjective review of the decision criteria in the context of pair-wise comparisons. All of these decisions marked by subjectivity can introduce some bias into the analyses. However, measures have been taken to address this problem, such as evaluations by several independent analysts, as well as the examination of intercoder reliability.

7. Conclusions

The novelty of the approach proposed in this paper is the merging of two complementary techniques (LSA und ANP) into a systematic and simultaneously semi-automatic implementation of the DLL concept in the project environment. The LSA has therefore been proposed for the efficient extraction of lessons learned from large repositories of historical project documentation. With ANP, a method was proposed for the systematic integration of the previously extracted project knowledge into the evaluation and design of new projects using a Project Evaluation Network (PEN). By these means, the proposed implementation approach contributes to the project management practice not only by demonstrating a solution for a continuous DLL process based upon historical lessons learned, but also by presenting an approach for systematic decision making as part of the evaluation and further development of project concepts.

Future research efforts will attempt to develop the implementation approach further by improving the precision of lessons learned extraction, by integrating subsequent simulations of specific project concepts, and, furthermore, by extending the approach in the sense of a triple-loop project learning framework. In this process, the implementation approach will be further tested in a practical project environment.

Conflict of interest

None.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sector.

Appendix

Table A.1
Project-governing variables.

Dimensions and criteria		Terms (excerpt)
D_1	Business domain	
C_{11}	Strategic alignment	strategic, enterprise, corporation, SME, agreement, success, factor, challenge, control, compromise, optimize, outstanding, precondition, issue, establish, consultant, member, marketing
C_{12}	Market position	e-shop, position, corporation, marketplace, firm, niche, online, product, service, business, industry, current, market
C_{13}	Image and Online reputation	image, poor, negative, perceive, argue, fear, online, product, website, perception, feedback, confident, positive, risk
C_{14}	Customer satisfaction	customer, satisfaction, focus, high, degree, great, solution, new, shop, consumer, growth, optimize, stay, decide, core, business
C_{15}	Organizational change	employee, work, change, previous, organization, administration, expectation, flexible, handle, work, assign, new, replace, people, check, involve
C_{16}	Supply chain innovation	supply, chain, innovation, deployment, initiative, transformation, advantage, future, commerce, technology, sufficient, procedure, help, application
D_2	Project product	
C_{21}	Innovative potential	innovation, advantage, launch, create, retention, encourage, function, competition, proactive, pressure, tradition, increase, electronic, commerce, allow, save, sell, target
C_{22}	Expandability and upgradeability	perspective, expansion, original, flexible, new, simple, additional, platform, requirement, change, allow, application, enable, implement, integration
C_{23}	Customization	software, customization, check, capability, program, careful, retail, standard, comprehensive, core, complex, expertise, price, development, solution
C_{24}	Digitalization	file, digital, data, print, documentation, eliminate, unnecessary, double, produce, paper, decrease, unify, system, send, cost
C_{25}	Automation potential	automation, high, reach, process, integration, manual, carry, fully, mean, minimum, effort, handle, procurement, remove, capacity
C_{26}	Transaction monitoring and securtiy	monitor, maintain, break, status, performance, data, workflow, process, quality, observe, transaction, data, information, consistent, outside, board
C_{27}	Investment costs	cost, invest, design, acquisition, price, pay, high, low, company, analysis, capital, liability, charge, tailor
C_{28}	Content management	content, provide, remove, collect, data, information, knowledge, announce, commercial, interest, mobile, woldwide, segment, seller, analyse
D_3	Project engineering	
C_{31}	System integration	webshop, system, integration, process, solution, service, incorporate, difficult, transaction, enter, choice, successful, select, option, purchase, delivery
C_{32}	Design and development	design, webshop, acquire, establish, acquisition, development, interface, simple, easy, network, set, integration, vision
C_{33}	Networking	network, broad, exchange, internet, platform, user, integration, entry, open, barrier, access, group, simple, high, data, involve, solution, requirement, involve, allow
C_{34}	Webhosting	Webhosting, internet, HTTP, WWW, PHP, commerce, service, provider, space, distribution, ISP
D_4	Organization management	
C_{41}	Collaboration	collaboration, complex, participant, join, partnership, player, relations, network, lean, workflow, process, training, system organization, portal, effort
C_{42}	Negotiation and contracting	negotiation, push, contract, agency, confirmation, tailor, agreement, save, force, situation, energy, goal, cost, contractor, length,
C_{43}	Scepticism, reluctancy and resistance	reluctancy, user, method, switch, launch, new, experience, sceptic, encourage, accept, employee, introduction, use, solution, platform, trust, adoption
C_{44}	Help desk and user support	user, help, use, specialist, practical, problem, encounter, friendly, workflow, challenge, message, case, test, error, solve, system, support, requirement
C_{45}	Procurement coordination	procure, coordination, partner, central, pricing, storage, simple, connection, feature, structure, completion, simplify, dynamic, offer, data, standard
D_5	Project management	
C_{51}	Staff training	personnel, training, demand, staff, implement, know, how, accept, apply, automation, careful, environment, sufficient, illustration, fail
C_{52}	Rollout complexity	rollout, practical, project, implement, difficult, complex, party, future, process, evaluation, start, positive, emphasize, prerequisite, quick, sufficient, stage, early, convince, adaption, version
C_{53}	Change management	change, motivate, stage, training, early, reluctancy, lead, help, success, provice, user, ability, solve, describe, qualitfication, achieve, know, how, problem

Table A.2
Supermatrix before covergence.

		D_1						D_2								D_3				D_4					D_5		
		C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	C_{26}	C_{27}	C_{28}	C_{31}	C_{32}	C_{33}	C_{34}	C_{41}	C_{42}	C_{43}	C_{44}	C_{45}	C_{51}	C_{52}	C_{53}
D_1	C_{11}	0.0000	0.4048	0.1784	0.0792	0.3683	0.3440																				
	C_{12}	0.3348	0.0000	0.1824	0.2547	0.0798	0.1241																				
	C_{13}	0.1092	0.1730	0.0000	0.4852	0.0325	0.0316																				
	C_{14}	0.3456	0.2942	0.5227	0.0000	0.1855	0.1153																				
	C_{15}	0.0463	0.0379	0.0328	0.0317	0.0000	0.3850																				
	C_{16}	0.1640	0.0901	0.0837	0.1491	0.3339	0.0000																				
D_2	C_{21}							0.0000	0.2023	0.2419	0.0541	0.0490	0.0807	0.3722	0.1400												
	C_{22}							0.2937	0.0000	0.2788	0.0305	0.0222	0.0360	0.1642	0.2792												
	C_{23}							0.2683	0.3013	0.0000	0.0911	0.1752	0.1564	0.2732	0.2700												
	C_{24}							0.0960	0.1665	0.1213	0.0000	0.4136	0.2024	0.0373	0.1443												
	C_{25}							0.0953	0.0829	0.0322	0.3463	0.0000	0.3218	0.0530	0.0599												
	C_{26}							0.0269	0.0290	0.0338	0.2262	0.1209	0.0000	0.0374	0.0500												
	C_{27}							0.1914	0.1809	0.0898	0.0569	0.0601	0.0598	0.0000	0.0566												
	C_{28}							0.0284	0.0370	0.2023	0.1949	0.1591	0.1430	0.0627	0.0000												
D_3	C_{31}															0.0000	0.6127	0.5373	0.6273								
	C_{32}															0.6561	0.0000	0.2535	0.1307								
	C_{33}															0.2543	0.3092	0.0000	0.2420								
	C_{34}															0.0896	0.0780	0.2092	0.0000								
D_4	C_{41}																			0.0000	0.2836	0.5384	0.4400	0.3920			
	C_{42}																			0.4740	0.0000	0.1097	0.0477	0.3783			
	C_{43}																			0.1338	0.4614	0.0000	0.3703	0.1862			
	C_{44}																			0.0491	0.0809	0.0479	0.0000	0.0434			
	C_{45}																			0.3431	0.1741	0.3040	0.1420	0.0000			
D_5	C_{51}																								0.0000	0.5000	0.1667
	C_{52}																								0.8333	0.0000	0.8333
	C_{53}																								0.1667	0.5000	0.0000

Table A.3
Supermatrix after covergence.

		D_1						D_2								D_3				D_4					D_5		
		C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	C_{26}	C_{27}	C_{28}	C_{31}	C_{32}	C_{33}	C_{34}	C_{41}	C_{42}	C_{43}	C_{44}	C_{45}	C_{51}	C_{52}	C_{53}
D_1	C_{11}	0.1960	0.1960	0.1960	0.1960	0.1960	0.1960																				
	C_{12}	0.1814	0.1814	0.1814	0.1814	0.1814	0.1814																				
	C_{13}	0.1770	0.1770	0.1770	0.1770	0.1770	0.1770																				
	C_{14}	0.2425	0.2425	0.2425	0.2425	0.2425	0.2425																				
	C_{15}	0.0777	0.0777	0.0777	0.0777	0.0777	0.0777																				
	C_{16}	0.1254	0.1254	0.1254	0.1254	0.1254	0.1254																				
D_2	C_{21}							0.1433	0.1433	0.1433	0.1433	0.1433	0.1433	0.1433	0.1433												
	C_{22}							0.1478	0.1478	0.1478	0.1478	0.1478	0.1478	0.1478	0.1478												
	C_{23}							0.1818	0.1818	0.1818	0.1818	0.1818	0.1818	0.1818	0.1818												
	C_{24}							0.1407	0.1407	0.1407	0.1407	0.1407	0.1407	0.1407	0.1407												
	C_{25}							0.1142	0.1142	0.1142	0.1142	0.1142	0.1142	0.1142	0.1142												
	C_{26}							0.0689	0.0689	0.0689	0.0689	0.0689	0.0689	0.0689	0.0689												
	C_{27}							0.0956	0.0956	0.0956	0.0956	0.0956	0.0956	0.0956	0.0956												
	C_{28}							0.1077	0.1077	0.1077	0.1077	0.1077	0.1077	0.1077	0.1077												
D_3	C_{31}															0.3708	0.3708	0.3708	0.3708								
	C_{32}															0.3113	0.3113	0.3113	0.3113								
	C_{33}															0.2154	0.2154	0.2154	0.2154								
	C_{34}															0.1026	0.1026	0.1026	0.1026								
D_4	C_{41}																			0.2866	0.2866	0.2866	0.2866	0.2866			
	C_{42}																			0.2411	0.2411	0.2411	0.2411	0.2411			
	C_{43}																			0.2084	0.2084	0.2084	0.2084	0.2084			
	C_{44}																			0.0527	0.0527	0.0527	0.0527	0.0527			
	C_{45}																			0.2111	0.2111	0.2111	0.2111	0.2111			
D_5	C_{51}																								0.2727	0.2727	0.2727
	C_{52}																								0.4545	0.4545	0.4545
	C_{53}																								0.2727	0.2727	0.2727

References

Almeida, M. V., & Soares, A. L. (2014). Knowledge sharing in project-based organizations: Overcoming the informational limbo. *International Journal of Information Management, 34*(6), 770–779.

Argote, L. (2011). Organizational learning research: Past, present and future. *Management learning, 42*(4), 439–446.

Argyris, C. (1977). Double loop learning in organizations. *Harvard Business Review*, (September–October), 115–125.

Argyris, C., & Schön, D. A. (1978). *Organizational learning: A theory of action perspective*. Wokingham: Addison-Wesley.

Argyris, C. (1999). *On organizational learning*. Malden, Mass: Wiley-Blackwell (2nd ed.).

Ayas, K., & Zeniuk, N. (2001). Project-based learning: Building communities of reflective practitioners. *Management Learning, 32*(1), 61–76.

Barclay, C., & Osei-Bryson, K.-M. (2010). An exploration of knowledge management practices in IT projects: A case study approach. In Proceedings of the Americas conference on information systems *(AMCIS): 452* Paper.

Bhatt, G. D., & Zaveri, J. (2002). The enabling role of decision support systems in organizational learning. *Decision Support Systems, 32*(3), 297–309.

Boh, W. F. (2007). Mechanisms for sharing knowledge in project-based organizations. *Information and Organization, 17*(1), 27–58.

Bohanec, M., Robnik-Šikonja, M., & Borštnar, M. K. (2017a). Decision-making framework with double-loop learning through interpretable black-box machine learning models. *Industrial Management & Data Systems, 117*(7), 1389–1406.

Bohanec, M., Robnik-Šikonja, M., & Borštnar, M. K. (2017b). Organizational learning supported by machine learning models coupled with general explanation methods: A Case of B2B sales forecasting. *Organizacija, 50*(3), 217–233.

Brady, T., & Davies, A. (2004). Building project capabilities: From exploratory to exploitative learning. *Organization studies, 25*(9), 1601–1621.

Caniëls, M. C., & Bakens, R. J. (2012). The effects of project management information systems on decision making in a multi project environment. *International Journal of Project Management, 30*(2), 162–175.

Carrillo, P., Harding, J., & Choudhary, A. (2011). Knowledge discovery from post-project reviews. *Construction Management and Economics, 29*(7), 713–723.

Chan, R., & Rosemann, M. R. (2001). Managing knowledge in enterprise systems. *Journal of Systems and Information Technology, 5*(2), 37–54.

Cheng, M. M., Schulz, A. K. D., & Booth, P. (2009). Knowledge transfer in project reviews: The effect of self-justification bias and moral hazard. *Accounting & Finance, 49*(1), 75–93.

Chou, J. S., & Tseng, H. C. (2011). Establishing expert system for prediction based on the project-oriented data warehouse. *Expert Systems with Applications, 38*(1), 640–651.

Chou, S. W. (2003). Computer systems to facilitating organizational learning: IT and organizational context. *Expert Systems with Applications, 24*(3), 273–280.

Choudhary, A. K., Oluikpe, P. I., Harding, J. A., & Carrillo, P. M. (2009). The needs and benefits of text mining applications on post-project reviews. *Computers in Industry, 60*(9), 728–740.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent se-mantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Disterer, G. (2002). Management of project knowledge and experiences. *Journal of Knowledge Management, 6*(5), 512–520.

Dumais, S. (2004). Latent semantic analysis. *Annual Review of Information Science and Technology, 38*, 189–230.

Easterby-Smith, M., & Lyles, M. (2011). The evolving field of organizational learning and knowledge management. In *Handbook of organizational learning and knowledge management* (pp. 1–20). Chichester: Wiley.

Eppler, M. J., & Mengis, J. (2004). The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines. *The information society, 20*(5), 325–344.

Evangelopoulos, N., Zhang, X., & Prybutok, V. R. (2012). Latent semantic analysis: Five methodological recommendations. *European Journal of Information Systems, 21*(1), 70–86.

García, M. N. M., Román, I. R., Peñalvo, F. J. G., & Bonilla, M. T. (2008). An association rule mining method for estimating the impact of project management policies on software quality, development time and effort. *Expert Systems with Applications, 34*(1), 522–529.

Garvin, D. A. (1993). Building a Learning Organization. *Harvard Business Review, July-August*, 78–91.

Grant, R. M. (1996). Toward a knowledge-based theory of the firm. *Strategic Management Journal, 17*(S2), 109–122.

Grisold, T., & Peschl, M. F. (2016). Facilitating radical innovation by double-loop learning: Introducing a strategic framework and toolbox. *Theory and Applications in the Knowledge Economy, 440*.

Grobelnik, M., & Mladeni, D. (2005). Automated knowledge discovery in advanced knowledge management. *Journal of Knowledge Management, 9*(5), 132–149.

Haksever, A. M., & Fisher, N. (1996). A method of measuring information overload in construction project management. In *Proceedings of the CIB, 21–24 October, Beijing, China*.

Haksever, A. M. (2000). A model to predict the occurrence of information overload of project managers. In *Proceedings of the CIB: 78*.

Hanisch, B., Lindner, F., Mueller, A., & Wald, A. (2009). Knowledge management in project environments. *Journal of Knowledge Management, 13*(4), 148–160.

Hobday, M. (2000). The project-based organisation: An ideal form for managing complex products and systems? *Research policy, 29*(7), 871–893.

Holt, G. D., Love, P. E., & Li, H. (2000). The learning organisation: Toward a paradigm for mutually beneficial strategic construction alliances. *International journal of project management, 18*(6), 415–421.

Hong, H. K., Kim, J. S., Kim, T., & Leem, B. H. (2008). The effect of knowledge on system integration project performance. *Industrial Management & Data Systems, 108*(3), 385–404.

Hsu, C. W., & Hu, A. H. (2009). Applying hazardous substance management to supplier selection using analytic network process. *Journal of Cleaner Production, 17*(2), 255–264.

Huber, G. P. (1991). Organizational learning: The contributing processes and the literatures. *Organization science, 2*(1), 88–115.

Idrus, A., Nuruddin, M. F., & Rohman, M. A. (2011). Development of project cost contingency estimation model using risk analysis and fuzzy expert system. *Expert Systems with Applications, 38*(3), 1501–1508.

Jaaron, A. A., & Backhouse, C. J. (2017). Operationalising "double-loop" learning in service organisations: A systems approach for creating knowledge. *Systemic Practice and Action Research, 30*(4), 317–337.

Kang, K., & Hahn, J. (2009). Learning and forgetting curves in software development: Does type of knowledge matter?. In Proceedings of the International Conference on Information Systems *(ICIS): 31* Paper.

Kantamara, P., & Vathanophas, V. (2014). Single-loop *vs*. double-loop learning: An obstacle or a success factor for organizational learning. *International Journal of Education and Research, 2*(7), 55–62.

Keegan, A., & Turner, J. R. (2001). Quantity *versus* quality in project-based learning practices. *Management learning, 32*(1), 77–98.

Kim, H., MacDonald, R. H., & Andersen, D. F. (2013). Simulation and managerial decision making: A double-loop learning framework. *Public Administration Review, 73*(2), 291–300.

King, W. R. (2009). Text analytics: Boon to knowledge management? *Information Systems Management, 26*(1), 87.

Koners, U., & Goffin, K. (2007). Learning from postproject reviews: A cross-case analysis. *Journal of Product Innovation Management, 24*(3), 242–258.

Kotnour, T. (2000). Organizational learning practices in the project management environment. *International Journal of Quality & Reliability Management, 17*(4/5), 393–406.

Kululanga, G. K., & Kuotcha, W. S. (2008). Measuring organisational learning through project reviews.. *Engineering,* Construction and Architectural Management, *15*(6), 580–595.

Labib, A., & Read, M. (2015). A hybrid model for learning from failures: The Hurricane Katrina disaster. *Expert Systems with Applications, 42*(21), 7869–7881.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Dicourse processes, 25*(2-3), 259–284.

Lee, J. W., & Kim, S. H. (2000). Using analytic network process and goal programming for interdependent information system project selection. *Computers & Operations Research, 27*(4), 367–382.

Liebowitz, J. (2001). Knowledge management and its link to artificial intelligence. *Expert Systems with Applications, 20*(1), 1–6.

Manning, C. D., & Schuetze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.

McClory, S., Read, M., & Labib, A. (2017). Conceptualising the lessons-learned process in project management: Towards a triple-loop learning framework. *International Journal of Project Management, 35*(7), 1322–1335.

Meade, L. M., & Presley, A. (2002). R&D project selection using the analytic network process. *IEEE Transactions on Engineering Management, 49*(1), 59–66.

Middleton, C. J. (1967). How to set up a project organization. *Harvard Business Review, 45*(2), 73–82.

Miner, G., Elder, J., Hill, T., Nisbet, R., Delen, D., & Fast, A. (2012). *Practical text mining and statistical analysis for non-structured text data applications*. Cambridge, MA: Academic Press.

Müller, O., Schmiedel, T., Gorbacheva, E., & vom Brocke, J. (2016). Towards a typology of business process management professionals: Identifying patterns of competences through latent seman-tic analysis. *Enterprise Information Systems, 10*(1), 50–80.

Niemira, M. P., & Saaty, T. L. (2004). An analytic network process model for financial-crisis forecasting. *International Journal of Forecasting, 20*(4), 573–587.

OGC. (2009). *Managing and directing successful projects with PRINCE2* (5th ed.). Norwich: The Stationery Office Books Office of Government Commerce.

Parnell, J. A., Von Bergen, C. W., & Soper, B. (2005). Profiting from past triumphs and failures: Harnessing history for future success. *SAM Advanced Management Journal, 70*(2), 36.

Pemberton, J. D., & Stonehouse, G. H. (2000). Organisational learning and knowledge assets-an essential partnership. *The learning organization, 7*(4), 184–194.

Pérez-Bennett, A., Davidsen, P., & E. López, L. (2014). Supercharging case-based learning via simulators: Achieving double-loop learning with case-sims. *Management Decision, 52*(9), 1801–1832.

PMI. (2008). *PMBOK guide*. Pennsylvania:PRoject Management Institute, Inc 4th ed.

Prencipe, A., & Tell, F. (2001). Inter-project learning: Processes and outcomes of knowledge codification in project-based firms. *Research Policy, 30*(9), 1373–1394.

Rauffet, P., Cunha, C. D., & Bernard, A. (2014). A double–loop learning system for knowledge transfer and reuse in groups: Application of a roadmapping approach. *International Journal of Knowledge and Learning, 9*(1-2), 63–86.

Reddick, C. G., Chatfield, A. T., & Ojo, A. (2017). A social media text analytics framework for double-loop learning for citizen-centric public services: A case study of a local government Facebook use. *Government Information Quarterly, 34*(1), 110–125.

Reich, B. H., Gemino, A., & Sauer, C. (2008). Modeling the knowledge perspective of IT projects. *Project Management Journal, 39*(S1), 4–14.
Reychav, I., Kumi, R., Sabherwal, R., & Azuri, J. (2016). Using tablets in medical consultations: Single loop and double loop learning processes. *Computers in Human Behavior, 61*, 415–426.
Saaty, T. L. (1996). *Decision making with dependence and feedback: The analytic network process*. Pittsburgh: PA: RWS Publications.
Schalken, J., Brinkkemper, S., & Van Vliet, H. (2006). A method to draw lessons from project postmortem databases. *Software Process: Improvement and Practice, 11*(1), 35–46.
Schindler, M., & Eppler, M. J. (2003). Harvesting project knowledge: A review of project learning meth-ods and success factors. *International Journal of Project Management, 21*(3), 219–228.
Sidorova, A., Evangelopoulos, N., Valacich, J. S., & Ramakrishnan, T. (2008). Uncovering the intellectual core of the information systems discipline. *MIS Quarterly, 32*(3), 467–482.
Strait, C. (2006). Too much information. Project Management Institute (PMI). http://www.pmi.org/learning/point-information-overload-management-process-2769. Accessed 07.12.2015.
Takahashi, S. (2006). Agent-based organizational cybernetic approach to organizational learning. In *IEEE International Joint Conference SICE-ICASE* (pp. 4591–4595).
Ting, S. L., Kwok, S. K., Tsang, A. H., & Lee, W. B. (2010). CASESIAN: A knowledge-based system using statistical and experiential perspectives for improving the knowledge sharing in the medical prescription process. *Expert Systems with Applications, 37*(7), 5336–5346.
Turban, E., & Watkins, P. R. (1986). Integrating expert systems and decision support systems. *MIS Quarterly, 10*(2), 121–136.
Ur-Rahman, N., & Harding, J. A. (2012). Textual data mining for industrial knowledge management and text classification: A business oriented approach. *Expert Systems with Application, 39*(5), 4729–4739.
Von Zedtwitz, M. (2002). Organizational learning through post–project reviews in R&D. *R&D Management, 32*(3), 255–268.
Walker, D. H., & Christenson, D. (2005). Knowledge wisdom and networks: A project management centre of excellence example. *The Learning Organization, 12*(3), 275–291.
Weber, R., Aha, D., Muñoz-Ávila, H., & Breslow, L. (2010). An intelligent lessons learned process. *Foundations of Intelligent Systems*, 39–41.
Wong, P. S., Cheung, S. O., & Fan, K. L. (2009). Examining the relationship between organizational learning styles and project performance. Journal of Construction Engineering and Management, , *135*(6), 497–507.
Zhan, Q., & Zheng, W. (2016). An in-depth learn from incidents (LFI) approach based on HFACS-RAs. In IEEE International Conference on Intelligent Rail Transportation *(ICIRT)* (pp. 177–185).
Zhang, W., Yoshida, T., & Tang, X. (2009). Using ontology to improve precision of terminology extraction from documents. *Expert Systems with Applications, 36*(5), 9333–9339.
Zhao, D., & Zuo, M. (2011). Knowledge Transferred across projects and its characteristics in an it service enterprise. In *Proceedings of international conference on information technology, computer engineering and management sciences (ICM)* (pp. 267–270).