Behaviour & Information Technology, 2014
Vol. 33, No. 5, 443–451, http://dx.doi.org/10.1080/0144929X.2013.815276

Ontology engineering step in design science research methodology: a technique to gather and reuse knowledge

Lukasz Ostrowski[a*], Markus Helfert[a] and Nelson Gama[b]

[a]*School of Computing, Dublin City University, Glasnevin, Dublin 9, Ireland;* [b]*Instituto Superior Técnico, Praça do Município, 1100-001 Lisbon, Portugal*

(*Received 2 April 2012; final version received 7 June 2013*)

Design science (DS) is an emerging research paradigm in the information systems field. One of its challenges is to systematically structure knowledge for business solution artefacts. In this paper, we address this challenge by presenting an ontology engineering process. It structures gathered knowledge based on domain-specific concepts and relations. Application of the process results in an improvement in DS artefacts in terms of representational information quality. The goal of the paper is to place the ontology engineering process in the DS research methodology and provide pragmatic steps to follow the process.

Keywords: design science research; ontology engineering; meta-design

1. Introduction

Considering the whole tray of ideas involved what we are most struck by – what we are evidently most interested in – is how the unseen gives reason for the seen. We ask how unknown motives generate known acts and how unknown talent creates knowable artefacts (Herrman 2009). We want to provide techniques that make success more available, more frequent, consistent and correct. The study of such techniques is methodology.

We understand it as a collection of procedures, techniques, tools and documentation aids which helps the systems developers in their efforts to implement a new information system (IS). A methodology consists of phases, themselves consisting of sub-phases, which guides the system developers in their choice of techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate ISs projects. … 'But a methodology is more than merely a collection of these things. It is usually based on some philosophical view; otherwise it is merely a method, like a recipe' (Avison and Fitzgerald 1988).

Design science (DS) research methodology has received increased attention in computing and ISs research (Kuechler and Vaishnavi 2008). It has become an accepted approach for research in the IS discipline, with dramatic growth in the related literature (Carlsson *et al.* 2011). However, its state of art does not offer consistent and comprehensive phases, which will guide DS researchers in their choices of techniques (Alturki *et al.* 2011). In this paper, we refer to the reference model (Ostrowski and Helfert 2012) which mainly provides operational steps for researchers aiming for process-oriented artefacts in DS research. However, its ontology engineering step, which we present in this paper, can be applied to wider types of DS artefacts. This step provides a solid foundation on which we can build sharable knowledge bases for wider usability than that of a conventional knowledge base. Application of this technique is to systemise knowledge before an artefact is being instantiated.

If we regard a methodology itself as a human activity system, we can then apply the same sort of quality criteria as we should apply to any such system, including ISs themselves (Veryard 1985). From these quality criteria, we can derive the principles of methodology design, in terms of the quality characteristics expected of a good methodology. Thus, to evaluate how the ontology engineering step impacts the current shape of DS methodology to produce artefacts, we apply representational information quality (Wang and Strong 1996).

In our research, we try to detail DS activities into the operational level as much as possible. In this paper, we introduce the ontology engineering process to structure gathered knowledge whose implication leads to the improvement of quality of DS artefacts. Operational steps in DS increase the efficiency and further decrease the cognitive effort involved. This paper is organised as follows. The next section reviews the DS research literature and proposes its challenges and potential ways of further development. Based on that review, the subsequent sections present the reference model that includes the ontology engineering phase. The following section describes process and activities of the ontology engineering. The final section presents

*Corresponding author. Email: lostrowski@computing.dcu.ie

an experiment to measure improvement of information quality of artefacts upon applying the ontology engineering process. To evaluate it we used representational information quality dimensions (Ge 2009). This paper may help to define future directions and phases of DS methodology within the full spectrum of ISs research.

2. Literature review

DS focuses on creations of artificial systems. It addresses research through the *building* and *evaluation* of artefacts designed to meet identified business needs (Hevner *et al.* 2004). Understanding the nature and causes of these needs can be a great help in designing solutions; however, DS does not limit itself to the understanding, but also aims to develop knowledge on the advantages and disadvantages of alternative solutions (Van Aken 2005). Literature reflects healthy discussion around the balance of rigor and relevance (Hevner *et al.* 2004) in DS research, which reflects it as a still shaping field (Iivari and Venable 2009).

Views and recommendations on the DS methodology vary among papers (Baskerville *et al.* 2009, Peffers *et al.* 2007). DS methodological guidelines from the precursors (Hevner 2004; Walls *et al.* 1992) are seldom 'applied', suggesting that the existing methodology is insufficiently clear, or inadequately operationalised – still too high level of abstraction (Peffers *et al.* 2007). Descriptions of activities (procedures, tools and techniques) that are needed to follow the methodology are only briefly indicated. Three main activities were identified as crucial in the development of DS artefacts and stated as a core of the reference model (Ostrowski *et al.* 2012). These are: literature review, collaboration with practitioners and relevant modelling techniques (Ostrowski *et al.* 2011). The reference model details these activities for the purpose of gaining abstract design knowledge. For a better overview, where the model fits in the DS methodology, we first introduce our understanding of the current state of art of DS and its artefacts.

Researchers understand artefacts as 'things', i.e. entities that have some separate existence (Goldkuhl 2004). They can be in the form of a construct, model, method and an instantiation (Hevner *et al.* 2004). In construction of the artefact, we observe two activity layers (Goldkuhl and Lind 2010): (1) design practice that produces situational design knowledge and instantiated IT artefacts and (2) meta-design that produces abstract design knowledge. This can be viewed as (2a) a preparatory activity before situational design is started, (2b) a continual activity partially integrated with the design practice and (2c) a concluding theoretical activity summarising, evaluating and abstracting results directed for target groups outside the studied design and use practices (Goldkuhl and Lind 2010). Meta-design artefacts are based on data types as opposed to instances of data. Its solutions are then unreal in some way or ways according to the three realities (Sun and Kantor 2006), such as unreal users, unreal systems and especially unreal problems (not conducted by the desired users). Thus, meta-design constructs solid and generic background for the design practice activities to construct solutions for a real environment (real people, real artefacts and in real settings) (Sun and Kantor 2006); it embraces all of the complexities of human practice in real organisations. In other words, the meta-design step concentrates on providing an optimal solution for the domain by trying to cover the whole spectrum. The design practice layer adjusts and applies the solution of the meta-design (i.e. abstract design knowledge) to a concrete business scenario (i.e. the instantiation of IT artefacts).

As abovementioned, abstract and situational design knowledge can be treated as two individual outcomes of DS. Thus, it seems reasonable to consider two different evaluation methods for each of them; these are artificial and naturalistic (Pries-Heje *et al.* 2008).

Evaluation has been a topic both in general IS research and in DS research. Venable (2006) classified DS research evaluation approaches into two primary forms: artificial and naturalistic evaluation. Artificial evaluation evaluates a solution technology in a contrived and non-realistic way. Naturalistic evaluation explores the performance of a solution technology and its real environment (i.e. within the organisation). Naturalistic evaluation methods offer the possibility to evaluate the real artefact in use by real users solving real problems (Sun and Kantor 2006), while artificial evaluation methods offer the possibility to control potential confusing variables more carefully and prove or disprove design hypotheses, design theories and the utility of design artefacts.

Artificial evaluation is placed after the meta-design activity because of its capability to test design hypotheses (Walls *et al.* 1992). Critical techniques may be used, but these generally supplement the main goal of proving or disproving the design theory and/or the utility of the DS artefact. Artificial evaluation includes laboratory experiments, field experiments, simulations, criterion-based analysis, theoretical arguments and mathematical proofs (Pries-Heje *et al.* 2008). The naturalistic evaluation after the design practice outcome is placed due to the capability of performing evaluation in a real environment. Naturalistic evaluation methods include case studies, field studies, surveys, ethnography, phenomenology, hermeneutic methods and action research (Pries-Heje *et al.* 2008).

Meta-design step plays a crucial role in constructing the knowledge base for a final instantiation and its utility. Figure 1 illustrates its place in DS research, and the general relationship among IS artefacts (Gregor and Jones 2007). The aim of the reference model is to guide DS researchers through the meta-design step (Ostrowski *et al.* 2012).

The next section briefly introduces the reference model, and how all activities cooperate to achieve a desired solution. Then it elaborates on the ontology engineering.

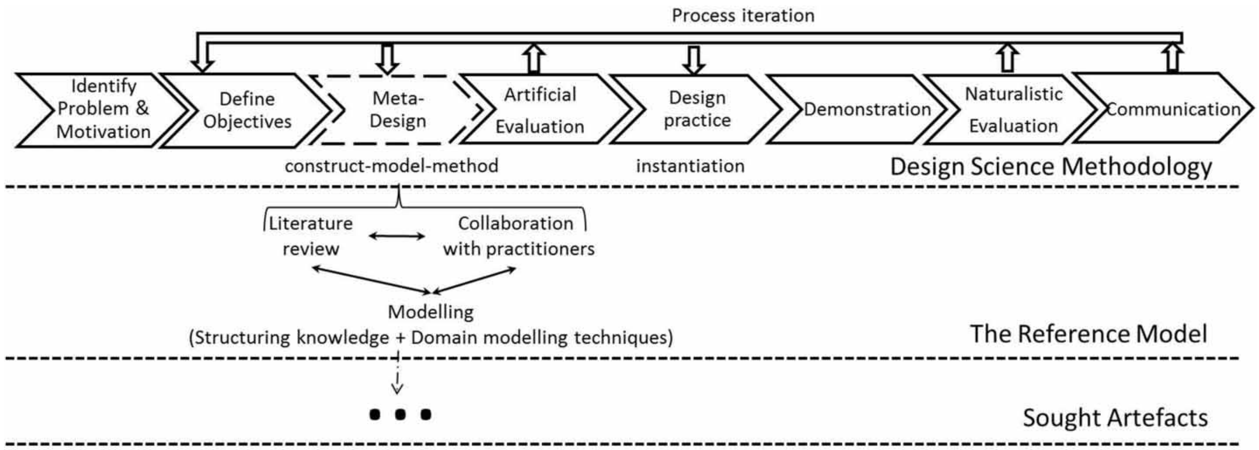

Figure 1. DS research methodology, adapted and updated from Peffers *et al.* (2007) and the core of reference model (Ostrowski *et al.* 2011).

3. The reference model

The idea behind the reference model (Figure 2) is to deliver the knowledge base, which combines information from two processes: literature review and collaboration with practitioners. Their main roles are to: (1) gather information related to the investigated domain of interest and (2) represent the information in an understandable way to the stakeholders.

An effective and quality literature review is one that is based upon a concept-centric approach rather than chronological or author-centric approach (Webster and Watson 2002). To avoid the risk of producing mind-numbing lists of citations and findings with not much of a plot, a systematic literature review was employed into the model. The value or importance of an effective literature review is in ensuring that the researcher can gain a full understanding of the body

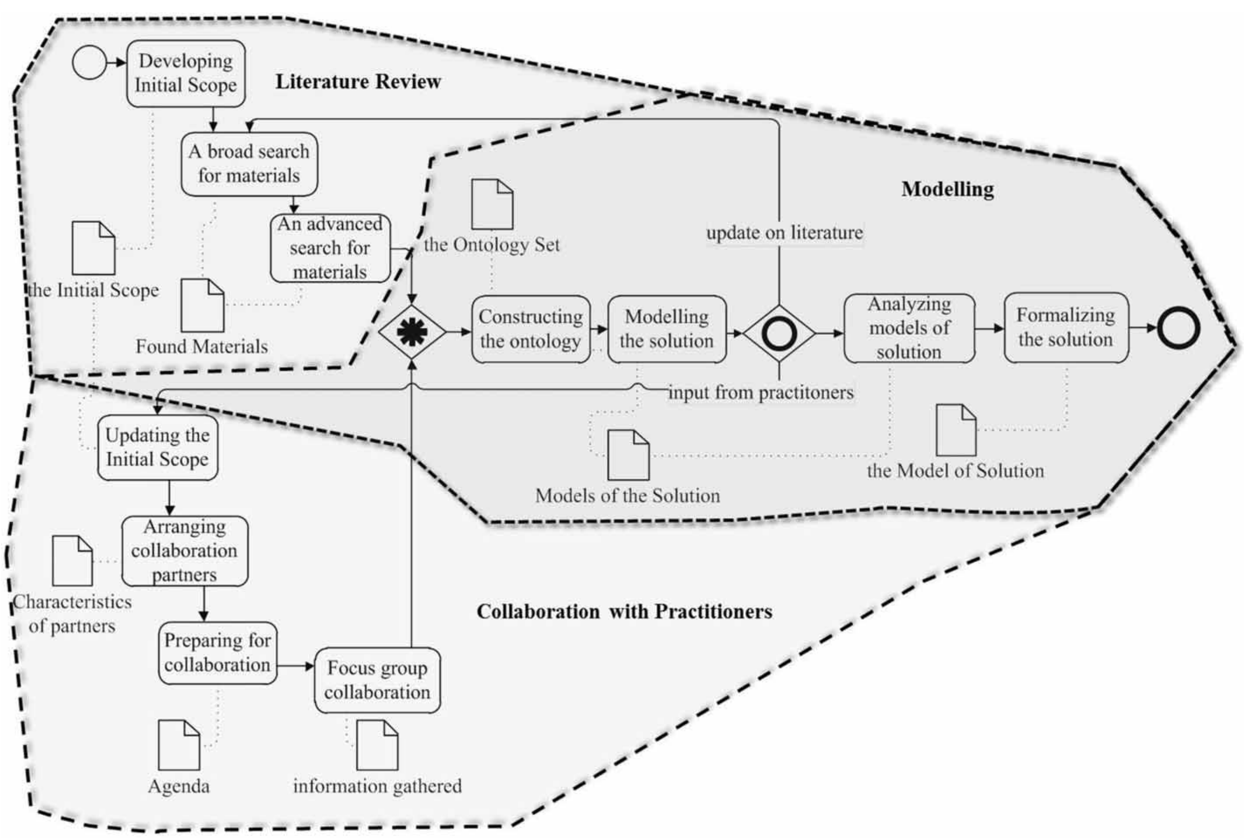

Figure 2. The reference model – overview.

of knowledge related to the phenomenon under study (Levy and Ellis 2006).

To build systematic development of transferable, reusable and predictable collaboration with practitioners, the collaboration engineering approach (Kolfschoten *et al.* 2010) was employed. It focuses on designing purposeful interaction within the context of a sequence of steps that helps a group to achieve its goal. The first step contains an analysis of the collaborative task that the group has to execute. The followings steps concern the decomposition of the collaborative task into different activities, patterns matching to the decomposed activities, and validation of the process to test whether it is likely to yield the desired results. These steps are usually not executed step-by-step, but iteratively (Kolfschoten and de Vreede 2009).

Before analysis and combination of solutions from these processes take place, each process models its own solution. Thus, to make the analysis and combination part more effective, the same modelling techniques in both processes are introduced. These are the ontology engineering and domain-specific modelling notation. The former gives researchers the design rationale of a knowledge base, kernel conceptualisation of the world of interest, semantic constraints of concepts together with sophisticated theories (Mizoguchi 2003). In the case of process-oriented artefacts, the latter may employ business process modelling notation (BPMN) (OMG 2011). For example, if a researcher investigates a process of an employee engagement, the ontology engineering technique will structure the gathered knowledge retrieved from those two processes. Then, the researcher will model the knowledge into a shape of a process using BPMN.

In this paper, we focus on the first facet of modelling – structuring knowledge. We approach it as a set of concepts within a domain and the relationships between those concepts. We found ontology engineering as an adequate technique for this task. The next section justifies and describes activities of ontology engineering as one of the modelling techniques used in the model. While we acknowledge this iterative nature of the activities involved, we discuss the model as a linear sequence of steps to keep the description straightforward.

3.1. *Ontology engineering*

Ontology community proposed various definitions of ontology (Mizoguchi 2003). Our understanding falls within the approach by Gruber (1993). Ontologies are agreements about shared conceptualisations. Shared conceptualisations include conceptual frameworks for modelling domain knowledge; content-specific protocols for communication among inter-operating agents and agreements about the representation of particular domain theories. In the knowledge sharing context, ontologies are specified in the form of definitions of representational vocabulary. A very simple case is a type hierarchy, specifying classes and their relationships. Relational database schema also serves as ontologies by specifying the relations that can exist in some shared database and the integrity constraints that must hold for them (Gruber 1993). The importance of knowledge engineering as 'content-oriented research' has been gradually recognised these days, and much progress has been attained (Mizoguchi 2003). However, there are still problems in building intelligent systems and in utilising knowledge base technology with its full power.

Ontology engineering gives researchers the design rationale of a knowledge base, kernel conceptualisation of the world of interest, semantic constraints of concepts together with sophisticated theories and technologies enabling accumulation of knowledge which is dispensable for knowledge processing in the real world (Mizoguchi 2003). Ontology engineering characterises the computational architecture of a knowledge-based system (Mizoguchi 1995). It is useful for describing the inherent problem-solving structure of the existing domain independently. This is achieved by analysing structures of real-world problems. The ultimate goal of ontology engineering includes providing a theory of all the vocabulary/concepts necessary for building a model of human problem-solving processes. Figure 3 represents the process to build such concepts.

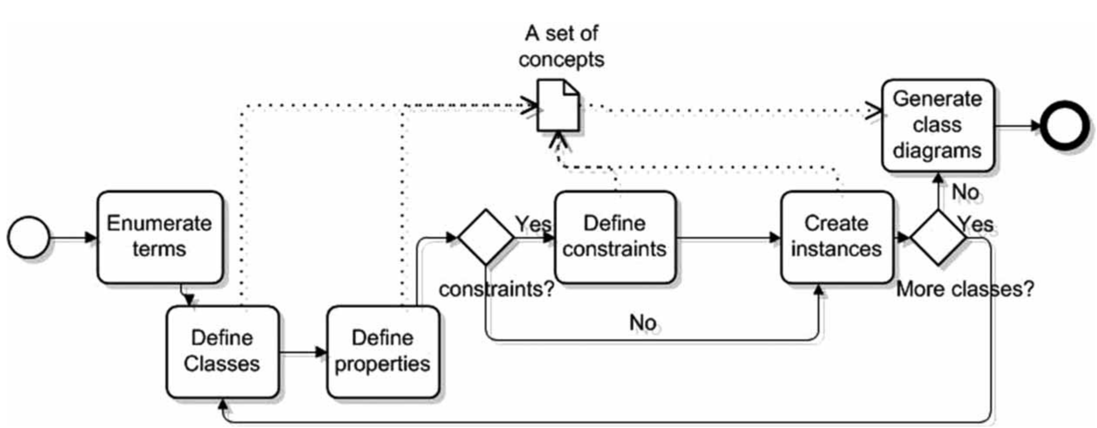

Figure 3. Ontology engineering process.

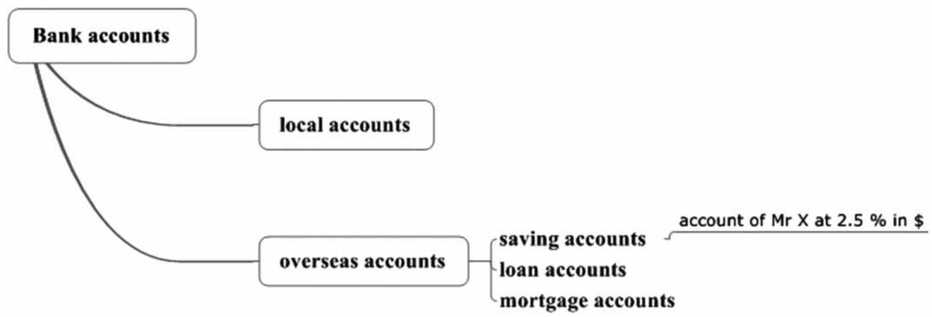

Figure 4. Levels in hierarchy.

The concept of ontology engineering process has been adapted from Noy and Tu (2002). Main activities involve defining terms in the domain and relations among them; defining concepts in the domain (classes); arranging the concepts in a hierarchy (subclass–superclass hierarchy); defining which properties classes can have and constraints on their values and defining individuals and filling in properties values.

The ontology engineering should be distinguished from object-oriented modelling. Ontology engineering approach reflects the structure of the world. It is often about structure of concepts; actual physical representation is not an issue whereas the object-oriented modelling reflects the structure of the data and code. This usually describes the physical representation of data (e.g. integer and char).

Now, we will elaborate on the ontology engineering process (Figure 3). First, researchers need to be fully aware about what terms they need to talk about; what the properties and what the purpose of these terms are. The idea behind it is to get a head around the domain before defining any classes. A typical answer to these questions could be a set of terms (for example, bank accounts, location, currency, saving account with 3% interest and loan) (Noy and Tu 2002). Thus, the first step in the ontology engineering process is to *enumerate terms*. Next is to *define classes and hierarchies of classes*. A class is a concept in the domain (e.g. a class of bank accounts or a class of customers). A class is a collection of elements with similar properties. Instances of classes would be for example, a saving account at 2.5% of a customer X. Classes usually constitute a taxonomic hierarchy (a subclass–superclass hierarchy). A class hierarchy is usually an IS-A hierarchy (Figure 4): an instance of a subclass is an instance of a superclass. If we regard a class as a set of elements, a subclass is a subset. For example: apple is a subclass of fruit (every apple is a fruit) and an overseas account is a subclass of a bank account (every saving account is a bank account).

The following task is to *define properties*. Properties, in a class definition, describe attributes of instances of the class and relations to other instances (for example, each account will have an assigned customer, currency type, opening date and interest rate). There are different types of properties: intrinsic (e.g. currency type and opening date), extrinsic (e.g. assigned customers) and relations to other objects (e.g. assigned customers – customers' details).

A subclass inherits all the properties from the superclass (if a bank account has an opening date and currency type, an overseas account also has an opening date and currency type). Classes and properties usually have documentation. They are described in natural language; listed with domain assumptions, synonyms relevant to the class definition.

Some properties may require limitations. Thus, *define constraints* is another task in the ontology engineering process. Property constraints describe or limit the set of possible values for a property (for example, the name of a customer is a string; the interest rate is integer, the saving account has exactly one customer). There are four common constraints of a property: cardinality – the number of values a property has; value type – the type of values a property has (e.g. string, number and Boolean); minimum and maximum value – a range of values for a numeric property and default value – the value a property has unless explicitly specified otherwise.

Second last task of the ontology engineering process is to *create an instance of a class*. The individuals in the class extension are called the *instances* of the class. The class becomes a direct type of the instance. However, the role of the individuals is to give a better understanding of the class rather than provide a particular instance that will be used in development in the design practice step (Figure 1). Any superclass of the direct type is a type of the instance. Researchers need to assign property values for the instance frame (e.g. Mr X's account – Figure 4). Properties values should conform to the facet constraints, but knowledge-acquisition tools often check that (e.g. protégé frames). Once all classes are identified and described, a unified modelling language (UML) class diagram (OMG-UML-Sup 2005) is generated. This diagram represents graphically the ontology (Figure 5). The visual is to help understand and better comprehend the domain of interest.

Next step following the reference model is to use this structured knowledge developed with the ontology engineering to model a domain using relevant modelling notation. These steps are out of scope of this paper. The next section presents the application of the ontology engineering process on a DS research project. We evaluate

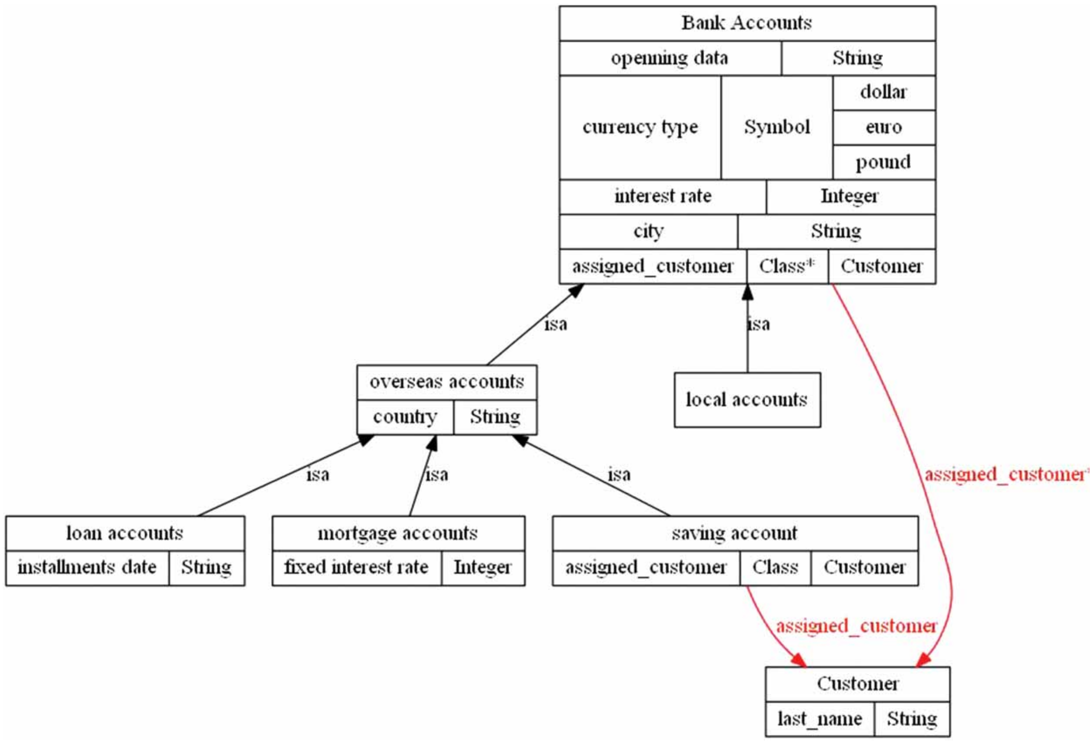

Figure 5. An example of ontology for IT services.

the outcome of the ontology engineering process with information quality dimensions.

4. Evaluation of the ontology process

To demonstrate the use of the ontology engineering process, we applied it to an on-going DS research project. The project focuses on issues of the IT service catalogue. Researchers followed the DS research paradigm (Figure 1), and stated the *research motivation* as the act of transforming resources into services is the base of the service management and without it an organisation is just an aggregate of resources that by itself does not bring value to the business (Goldstein 2002). Moreover, IT departments are mostly imposed to justify their services only from a cost–benefit perspective (Jauvé *et al*. 2006). Researchers decided to look into the transformation from resources into IT services and hope to smooth that process.

Objective of a solution was an artefact which identifies IT services through incidents. It aims to reach accurate identification of IT services, reduces the gap between the services provided by the organisation and users' perception of these services. It can be summarised as follows: when an incident is created, it implies the existence of the service in the organisation.

We split researchers into two groups at random. Each group followed the reference model (Figure 2) to search for information needed to build the IT service catalogue artefact. However, the difference between those groups was the fact that one group was given the ontology engineering process and the other was not. The incidents for the purpose of creating both artefacts came from the incidents database of a public organisation. Researchers spent over 3 months on that assignments having access to the incidents database, practitioners on the site and college resources.

Since the desired artefact (i.e. IT service catalogue) did not imply a domain-specific modelling technique; the abstract design knowledge of the artefact was the ontology created either with the ontology engineering process or without it (i.e. by any other means chosen by researchers). Having those two artefacts, we conducted an experiment.

We looked for a difference between the artefacts in terms of improvement of the perceived representational information quality. We asked practitioners, hereinafter called participants, to examine two artefacts, first developed with the ontology engineering and second without it, but both being following DS research. Then, we asked participants to respond to a questionnaire. In our experiment, to achieve the content validity, we built the questionnaire on the representational information quality dimensions: concise representation, consistency, ease of understanding and interpretability. Questions for each dimension were constructed based on their identified attributes. In terms of measurement, we used an 11-point Likert-type scale. Number 10 was labelled as 'extremely good', while 0 as 'not at all' and 5 as 'average'. Most questions in the questionnaire were formulated as 'how <Attributes of the Item> is the artefact?' For example, 'How easy is the artefact to understand?' The data then consist of each participant providing a score (rating) of how they found the artefact

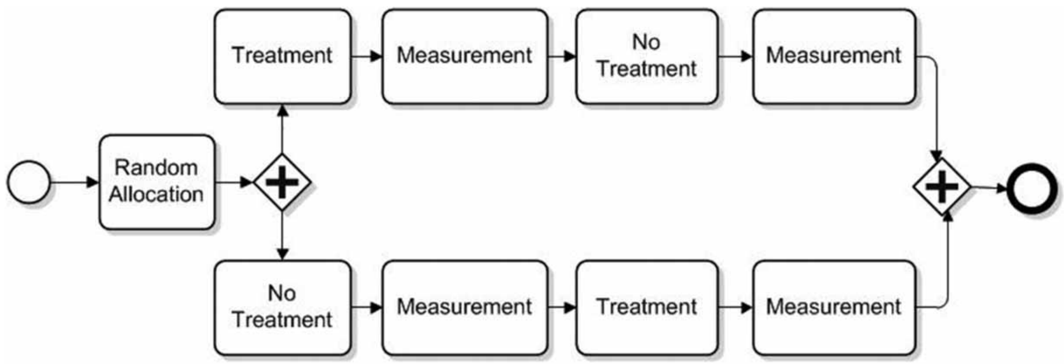

Figure 6. A basic two condition repeated measures design.

in terms of the quality of represented information. Quality cannot be taken for granted or assumed. Instead, quality is a subjective term for which each person has his own definition (Fishman 2009). We can be reasonable confident that a score of 8 refers to a better representation than a score of 7 and that a score of 9 almost certainly represents information better than a score of 8. However, we cannot conclude by how much guidance having the score 9 is better compared with other guidance having the score 8 or a 7. A score of 8 might represent an enormous difference over a score of 7, whereas a score of 9 might represent only a minor gain over a score of 8 or vice versa. We might question whether two artefacts, which both were rated as 7, are likely to be equally good. Hence, we treated these data (ratings) as ordinal data (Sheskin 2007).

In this experiment we used a basic two condition repeated measures design (Figure 6) (Field and Hole 2003). Under this design, each participant was randomly assigned to the order in which the artefacts were examined. The improvement was measured after examining each artefact. To maximise our chances of finding a difference we used a sample of 50 participants. We got each participant to take part in both conditions (they examined both artefacts). The order in which artefacts were assessed was counterbalanced, and there was a delay of 20 min between examining the artefacts.

Data of our experiment did not meet requirements of parametric tests; therefore, we chose a non-parametric test– the Wilcoxon-signed-rank test. It is used for testing differences between groups when there are two conditions and the same participants have been used in both conditions. In our experiment, each participant examined both artefacts (developed with or without the ontology engineering process). We measured the total scores of how well information provided by artefacts fits for use. It was hypothesised that using the ontology engineering process for artefacts in DS research would improve their quality of information provided.

The test showed that 8 of the 50 participants found better information quality of the artefact developed without the reference model, whereas 40 of the 50 participants favoured the artefact developed with the reference model. There was two tied rank (i.e. a participant who equally assessed both artefacts). The test statistic based on the negative ranks was -4.536 and that this value is significant at $p = .004$. This means that most people fell into the category of scoring better for the artefact developed with the ontology engineering process. There were significantly more people who had positive ranks than had negative ranks. Therefore, we can conclude that significantly information provided by the artefact developed with the ontology engineering process is of better representational information quality. The effect size of the experiment ($r = -0.4536$) represents a medium effect. This tells us that the effect of whether the artefact developed with or without the ontology engineering was examined was a substantive effect. We can say that using the ontology engineering process we can explain 20% of the total variability in total scores of the representational information quality of artefacts.

5. Conclusion and discussion

We observed challenges in structuring and standardising phases of the DS research methodology which will guide researchers in their choices of techniques, and also help them plan, manage, control and evaluate ISs projects. In this paper, we used the reference model to carry out DS research at the operational level. We introduced the ontology engineering process that aims to guide researchers in structuring gathered knowledge. We also showed that representational information quality of artefacts improved significantly upon applying the process. We showed improvement in representational information quality between business process model artefacts developed with and without usage of the ontology engineering process steps. It explained 20% of the total variability in total scores of the representational information quality of artefacts.

Several limitations have to be considered concerning the results of this research work. First, the ontology engineering as a part of the process-oriented reference model artefact is meant to be followed by a single DS researcher conducting DS research on a particular business process model artefact. Although, it is possible to split the research work in a way that researchers can follow the reference model in collaboration, all the results and findings outlined in this research work were based on a single usage of our artefact. A longitudinal study of the developed business process model artefacts with the reference model would enable us to conduct a more in-depth analysis of researchers' roles including such factors as research experience, use of technology and required supervision. However, the experiments did demonstrate that the ontology engineering step of process-oriented reference model had a positive impact on the perception of representational information quality of DS research methodology and its outcomes.

From the methodological perspective, in our measurement of representational information quality, we used questionnaire based on information quality dimension. A problem with a questionnaire relies on the respondents' accurately reflecting their viewpoints and how these reflect the real world. Based on previous work on representational information quality, this research study moderates this limitation by adopting the validated questionnaire and providing its reliability and validity analysis.

Finally, it cannot be claimed that the current version of the process-oriented reference model for business process model artefacts represents a stable, final version. For this purpose, more DS research projects are required. For example, the model could be used with DS research experts rather than with novice designers such as the students. Also, the identified six activities of the meta-design phase should be applied to a variety of DS research focusing on artefacts other than business process models.

Based on our findings and limitations, we propose a number of directions for future research on the reference model. It would be beneficial to explore which operational activities of the process-oriented reference model can only be applied to build business process model artefacts and which can also be used to build other process models that are industry specific or other types of DS artefacts in general.

Other efforts could be made to develop an instantiation layer artefact of the process-oriented reference model that would support and automate its activities. This would be the design practice phase of DS research methodology. Such instantiation in a form of a computer application could support the automatic generation and facilitation of research forms such as initial research scope, found research materials or focus group agenda. In addition, it could navigate DS researchers through all the activities to be undertaken in research and provide description for each step. Furthermore, given the prescription of capabilities within the process-oriented reference model, it should be possible to create an interface for a specific DS research project.

DS has become more widely accepted and employed research methodology in the IS field. The findings of our research work along with process-oriented reference model artefact provide a means for enhancing this methodology. Our research provides the additional structures, methods, activities and steps that are compulsory in conducting DS research in these ever demanding projects in ISs field.

References

Alturki, A., Gable, G.G., and Bandara, W., 2011. A design science research roadmap. *DESRIST 2011*, Heidelberg, 107–123.

Avison, D. and Fitzgerald, G., 1988. *Information systems development methodologies, techniques and tools*. Oxford: Blackwell Scientific Publications.

Baskerville, R., Pries-Heje, J., and Venable, J., 2009. Soft design science methodology. *DESRIST 2009*, Malvern, Caen.

Carlsson, S.A., *et al.*, 2011. Socio-technical IS design science research: developing design theory for IS integration management. *Information Systems and E-Business Management*, 14 (2), 109–131.

Field, A. and Hole, G., 2003. *How to design and report experiment*. London: Sage Publication.

Fishman, N.A., 2009. *Viral data in SOA: an enterprise pandemic*. New Jersey: IBM Press.

Ge, M., 2009. *Information quality assessment and effects on inventory decision-making*. Dublin: Dublin City University.

Goldkuhl, G., 2004. Design theories in information systems – a need for multi-grounding. *Journal of Information Technology and Application*, 3 (3), 59–72.

Goldkuhl, G. and Lind, M., 2010. A multi-grounded design research process. *Global perspectives on design science research DESRIST 2010*, Berlin, 45–60.

Goldstein, S.M., 2002. The service concept: the missing link in service design research? *Journal of Operations Management*, 20 (2), 121–134.

Gregor, S. and Jones, D., 2007. The anatomy of a design theory. *Journal of the Assocication for Information Systems*, 8, 312–335.

Gruber, T.R., 1993. A translation approach to portable ontologies. *Knowledge Acquisition*, 92 (71), 199–220.

Herrman, C.S., 2009. Fundamentals of methodology – part I: definitions and first principles (April 6, 2009). Available from: http://ssrn.com/abstract=1373976 or http://dx.doi.org/10.2139/ssrn.1373976

Hevner, A.R., *et al.*, 2004. Design science in information systems research. *MIS Quarterly*, 28 (1), 75–106.

Iivari, J. and Venable, J., 2009. Action research and design science research – seemingly similar but decisively dissimilar. *17th European conference on information systems*, Verona.

Jauvé, J., *et al.*, 2006. An introductory overview and survey of business-driven IT management. *Proceedings of the 1st IEEE / IFIP International Workshop On Business-Driven IT Management*, in conjunction with NOMS 2006, Canada: Vancouver, pp. 1–10.

Kolfschoten, G.L. and de Vreede, G.-J., 2009. A design approach for collaboration process: a multimethod design science study in collaboration engineering. *Journal of Management Information Systems*, 26 (1), 225–256.

Kolfschoten, G.L., *et al.*, 2010. Collaboration engineerability. *Group Decision Negotiation*, 25 (2), 301–321.

Kuechler, B. and Vaishnavi, V., 2008. On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, 17 (5), 489–504.

Levy, Y. and Ellis, J.E., 2006. Towards a framework of literature review process in support of information systems research. *The informing science and IT education joint conference*, Aalto, 171–181.

Mizoguchi, R., 1995. Task ontology for reuse of problem solving knowledge. *KB&KS95*, Enshede, The Netherland.

Mizoguchi, R., 2003. Tutorial on ontological engineering. *New Generation Computing*, 27 (3), 363–384.

Noy, N.F. and Tu, S.W., 2002. *Developing medical informatics ontologies with protege* [online]. Available from: http://protege.stanford.edu/amia2003/AMIA2003Tutorial.ppt

OMG, 2011. *Business process modelling notation* [online]. Available from: http://www.omg.org/spec/BPMN/2.0/ [Accessed March 2012].

OMG-UML-Sup, 2005. *Unified modeling language: superstructure* [online]. Available from: http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF/ [Accessed March 2012].

Ostrowski, L. and Helfert, M., 2012. Reference model in design science research to gather and model information. *18th Americas conference on information systems*, Seattle, WA.

Ostrowski, L., Helfert, M., and Hossain, F., 2011. A conceptual framework for design science research. *Business informatics research, LNBIP90*, Riga, Latvia, 345–354.

Ostrowski, L., Helfert, M., and Xie, S., 2012. A conceptual framework to construct an artefact for meta-abstract design. *45th Hawaii international conference on system sciences*, Maui, 4074–4081.

Peffers, K., Tuunanen, T., and Rothenberger, M., 2007. A design science research methodology. *Journal of Management Information Systems*, 4 (3), 45–77.

Pries-Heje, J., Baskerville, R., and Venable, J., 2008. Strategies for design science research evaluation. *16th European conference on information systems*, Caen, 255–266.

Sheskin, D., 2007. *Handbook of parametric and nonparametric statistical procedures*. Boca Raton, FL: Chapman & Hall/CRC.

Sun, Y. and Kantor, P.B., 2006. Cross-evaluation: a new model for information system evaluation. *Journal of the American Society for Information Science and Technology*, 11 (5), 614–662.

Van Aken, J.E., 2005. Management research as a design science: articulating the research products of mode 2 knowledge production in management. *British Journal of Management*, 15 (3), 19–36.

Veryard, R., 1985. What are methodologies good for? *Data Processing*, 27 (6), 9–12.

Walls, J., Widmeyer, G., and El Sawy, O., 1992. Building an information system design theory for vigilant EIS. *Information Systems Research*, 8 (2), 36–59.

Wang, R.Y. and Strong, D.M., 1996. Beyond accuracy: what data quality means to data consumers. *Journal of Management Information Systems*, 24 (2), 5–34.

Webster, J. and Watson, R.T., 2002. Analyzing the past to prepare for the future: writing a literature review. *MIS Quarterly*, 11 (5), 13–23.