

Ontology Design Patterns in Use

Lessons Learnt from an Ontology Engineering Case

Karl Hammar

Jönköping University
P.O. Box 1026
551 11 Jönköping, Sweden
`karl.hammar@jth.hj.se`

Abstract. Ontology Design Patterns show promise in enabling simpler, faster, more correct Ontology Engineering by laymen and experts alike. Evaluation of such patterns has typically been performed in experiments set up with artificial scenarios and measured by quantitative metrics and surveys. This paper presents an observational case study of content pattern usage in configuration of an event processing system. Results indicate that while structural characteristics of patterns are of some importance, greater emphasis needs to be put on pattern metadata and the development of pattern catalogue features.

Keywords: Ontology design pattern, evaluation, case study, event processing

1 Introduction

Ontology Design Patterns (ODPs) are intended to help guide domain experts in ontology engineering work, by packaging reusable best practice into small blocks of ontology functionality, either to be used as-is by practitioners, or to be used as inspiration and guidance for own development. This idea has gained some traction within the academic community, as evidenced by the WOP series of workshops held at ISWC 2009, 2010, and now 2012. If such patterns are to be accepted as useful artifacts also in practice, it is essential that they both model concepts and phenomena that are relevant to practitioners, and that they do so in a manner which makes them accessible and easy to use by said practitioners in real-world use cases.

This paper presents an attempt to learn more about ODP in-use qualities, how patterns are being used in a real case, and what users think about patterns, pattern portals, and pattern usage. In order to help formalize the study of the aforementioned issues, the following research questions have been selected for study:

1. What ODP characteristics do participants find helpful or harmful in ODP use?
2. How do users select and make use of ODPs?

3. What effects of ODP use on ontology engineering performance and resulting ontologies can be observed?

Within the ODP research community there exist different perspectives on what constitutes a pattern and how patterns should be categorized and sorted. The author of this paper largely subscribes to the definitions and pattern taxonomies presented within the NeOn project, published via the ODP portal¹. The patterns mentioned and studied in this work are all examples of NeOn Content ODPs, consisting of both a pattern documentation and a reusable OWL building block.

The paper is structured as follows; Section 2 introduces some related work on ODP evaluation and Event Processing, Section 3 presents the project in which this case study took place, Section 4 describes the method employed, and Section 5 presents the findings and some recommendations based on these.

2 Related Work

The project that frames this case study concerns the development of a system for Semantic Complex Event Processing (SCEP) using ODPs as system configuration modules. The following sections present some background on CEP in general and on existing pattern evaluation work.

2.1 Complex Event Processing

Complex Event Processing (CEP) is introduced by Luckham & Frasca in [12]. In their approach, patterns based on temporal or causal links between events are defined and formalized into mapping rules. When executed over incoming time-indexed data streams, patterns connect lower level basic events to form higher level complex events. Luckham develops these ideas further in [11]. CEP has since been established as a useful method in many domains, and CEP based on sensor data feeds has been explored in many papers, using RFID sensors, cameras, accelerometers, etc.

As indicated by Anicic et al. in [2], most CEP approaches however have some drawbacks, particularly in terms of recognizing events using background knowledge. Only those relations between events and entities which are made explicit in the input data stream can be used for detection and correlation purposes. In order to overcome these limitations Anicic et al. suggest Semantic Complex Event Processing (SCEP), in which background knowledge is encoded into knowledge bases that are accessed by a rules engine to support CEP.

2.2 ODP Evaluation

The effects of object oriented pattern use in software engineering and the harmful or beneficial properties of such patterns have been studied extensively, see for

¹ <http://ontologydesignpatterns.org>

instance [1, 13, 15]. Evaluations of pattern use in conceptual modeling is less common, but some examples of this type of research have been published, e.g. [14]. When it comes to pattern use in ontology engineering, as the author has previously found [8], the amount of work is also rather limited.

Possible benefits of ODP usage in ontology engineering have been shown by Blomqvist et al. in [3] and [4], both of which tested ODP usage according to the eXtreme Design method, by way of experimental setups with master and PhD student groups, and quantitative and qualitative surveys. Their results indicate that within this setting and for the modeled scenario, ODPs are perceived as useful, and the use of them result in fewer instances of a set of common modeling mistakes. However, they also report a perceived overhead associated with using the XD methodology and tooling, and find no strong support for ODPs improving the speed of ontology development. These experiments do not study the characteristics of the individual patterns in use.

Iannone et al. in [10] propose a semantics for expressing and method for computing the modularity (and consequently reusability characteristics) of ontology patterns. The method is implemented in a plugin for Protégé 4. The patterns under study are OPPL patterns and the algorithm presented is therefore incompatible with the view of ODPs as presented within the ODP community portal. However, conceptually the calculation of local and non local effects of pattern use seem to be relevant also for Content ODPs.

3 Case Characterization²

The project framing this case study is a small spinoff project from a larger project on threat detection using sensor systems where the partner research institute (hereafter RI) is involved. The work at RI focuses on development of a rule-based CEP subsystem intended to help isolate and correlate critical situations and threats based on incoming data. Within the spinoff project the aim is to develop the same functionality using semantic technologies. The motivation for this project is the increased flexibility of reasoning associated with using description logic languages, and the perceived gain in ease of reconfigurability associated with the use of ODPs. The following sections introduce the case participants and the architecture of the SCEP system.

3.1 Participants

Three participants attended the modeling workshops, participants A, B, and C. They are all male, and in the age bracket from 35 to 55 years. All three are researchers (two PhDs, one MSc) in software engineering or conceptual and data modeling within RI, and all three have some experience in such modeling. B and C have little or no prior knowledge of semantic web ontologies and

² For reasons of integrity and confidentiality, the case description and published data has been anonymized.

semantic technologies, whereas A has worked on these topics quite extensively, among other things researching rule languages for reasoning over semantic web ontologies. Their respective specialities are as follows:

- A has published on ontology matching, rule languages, model transformations, semantic technology use cases, etc.
- B has published on information logistics, mobile computing, context- and task-aware computing, etc.
- C has published on component based software engineering, middlewares, service orientation, system architectures, garbage collectors, etc.

3.2 System Architecture

The core of the system is a live observation knowledge base, defined according to an ontology schema. The ontology consists of both general features that are always relevant in the context of such a system (vocabularies of time, geographical locations and distances, sensor metadata, etc), and of features that are scenario and deployment specific. The latter features are imported from a set of four configuration knowledge bases, that together define system knowledge fusion behavior. These knowledge bases define, respectively: scenario configuration (i.e. background/context knowledge), situation correlation behavior, observation/entity fusion behavior, and critical situation detection behavior. Their contents are constructed by importing and adapting content ontology design patterns from a pattern repository.

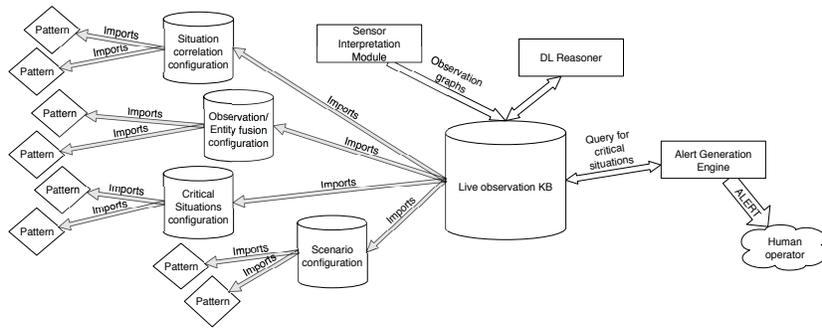


Fig. 1. Semantic Knowledge Fusion system architecture.

Data input from the deployed sensor subsystems is mapped to the general ontology vocabulary by sensor interpretation modules and stored as observation graphs in the knowledge base. A description logic reasoner is executed on the knowledge base and inferences about these observations are made. Then a rule engine is executed on top of the inferred knowledge, allowing greater expressivity in reasoning than that provided by description logic only (with the

rules employed being embedded within the utilized CODPs by way of annotation properties). If any observations are inferred to be instances of the critical situations defined within the critical situation configuration knowledge base, an alert is raised for a human operator to investigate the situation. For an overview of the system architecture, see Figure 1, and for a more in-depth description of the system see [9].

4 Method

Observation and data gathering was performed at a two-day modeling workshop at RI. The purpose of this workshop was to present the developments on the proposed system architecture and a prototype of the software to the participants, and to encourage them to develop configurations for it, thereby validating the applicability of the approach to their deployment scenarios.

Two scenario descriptions developed within the project were used to describe system deployment contexts³. The participants then attempted to model some typical relevant critical situations associated with each of these scenarios. Two examples of such critical situations are listed below:

- A gang is four or more people who have been seen together via at least three cameras over at least fifteen minutes and who are all wearing the same color clothing. A critical situation occurs when a gang of five or more football fans are loud and have within the last hour been spotted by a camera at a bar.
- Two vehicles are the same if they have the same license plate number *or* have the same brand, model and color and are observed by two cameras located at the same physical place within five seconds. A vehicle is behaving oddly if observed driving less than 15 km/h in three different cameras.

To their aid, the participants had a set of twenty ontology design patterns, of which fourteen were selected from the ODP community portal, and six were selected from other research projects. They were not provided with any training in pattern use, and were not recommended any particular development method, on the basis that providing such recommendations or training would restrict the participants' behavior and interaction with the patterns and the possibility of learning from their work.

During the modeling sessions data was gathered by way of audio and video recordings of the work in progress, photographs taken of ontology prototypes on the whiteboard, and notes taken on perceived key actions, behaviors, and trends taken independently by two researchers, the author and a senior professor with extensive experience of this research method. At the end of the second workshop day a semistructured group interview was held where the participants were queried about a number of different aspects of their experience and opinions on ODP use. Additionally, issues and statements of particular interest observed during the workshop were revisited and discussed, and conflicting interpretations resolved.

³ Downloadable from <http://purl.karlhammar.com/data/wop2012/>

4.1 Data Analysis

Upon completing the workshop, the recorded material was transcribed into text. The vast majority of the material was immediately understandable. In the cases where ambiguities required interpretation, markers were put down. Those sections were revisited at the end of transcription, when a greater experience of the participants' voices was established, and in the majority of cases then resolved. The few uncertainties that remained were clearly marked out in the transcribed text, and subsequently ignored in later analysis steps.

Table 1. Codes used in data analysis.

Code Label	Code Label
1 ODP structure	3 ODP catalogue and selection
1.1 ODP size	4 ODP effects
1.2 ODP imports	4.1 Efficiency
1.3 ODP complexity	4.2 Usefulness
2 OE method observations	5 Pattern insufficient
2.1 ODP method observations	6 ODP usage prerequisites
2.1.1 ODPs-as-guidance	7 DL/semantics limitations
2.1.2 ODPs-as-error-control	8 Top-down/bottom-up choices
2.1.3 ODP-attributable errors	9 Existing implicit ontology effects
2.1.4 ODPs-as-ground ontologies	10 Method/metamodel adequacy
2.2 Modeling errors	

The text material (notes and transcripts) was then analyzed according to established transcript analysis methods [5, 7]. All the texts were read through and fragments coded by theme. The texts were read twice, once to establish coding categories in the material (see Table 1), and once to apply codes to the text corpus. The fragments were grouped by code, and the collected material pertaining to each code studied to see what conclusions could be drawn regarding participant experiences, opinions and behavior.

4.2 On Validity and Generalizability

Many methods of increasing validity and reliability in case studies and theories based on them have been proposed [6, 16]. A common approach in such methods is to limit the potential bias in data collection, coding, and analysis, by involving multiple researchers, to verify each other's analyses and data. Another recommendation is to involve case participants and to let them verify the perceived veracity of data and analyses. Yet another common approach is to triangulate results by using multiple data collection methods. In this study, two researchers were involved in data collection and note-taking. Multiple data collection methods were employed (audiovisual recordings of modeling sessions, researcher notes, interview transcripts). Preliminary analyses were verified against case partici-

pant opinion by way of a group interview. Due to resource limitations, coding and analysis was performed by only one person.

A downside to case studies is that the generalizability of results is limited. In fact, case study results not supported by similar results from other cases or by well-established theory cannot be said to be scientifically generalizable at all. That is not to say that these types of results are useless – on the contrary, if the characteristics of a studied case are similar to those of a new project in development, recommendations can often be reused from such results. However, there can be no *guarantees* of applicability made, no warranty of a causal link between specified behavior and some expected outcome, granted by the qualitative researcher. The author adheres to this perspective and makes no guarantees of replicability of the results, but is convinced that the ODP community will anyway benefit from the knowledge gained in this study.

5 Findings and Recommendations

The following sections describe the data gathered, and present some observations and analyses pertaining to the research questions garnered from said data. Some of the analyses are accompanied by brief recommendations for ODP researchers, based on what has been observed in this case.

5.1 Data

The resulting dataset comprises some 21600 words, or approximately 85 pages of text. Of these, 16 pages are researcher notes, and 69 pages are audio or video transcriptions. The participants were initially skeptical about being recorded on film, and their behavior changed noticeably when cameras were present, becoming quite a lot more formal and tense. In order to promote a good natural working environment for observations, the researchers chose to turn off the recording equipment initially, turning it back on only when the participants had gotten warmed up to the task and seemed less concerned about this. Due to the triangulation in analysis, this is believed to have little effect on the reliability of the results however.

Additionally, six whiteboard illustrations were photographed. There were 187 applications of codes to fragments, with the distribution of fragments over codes shown in Table 2.

5.2 Important ODP Features

During the modeling and subsequent interviews, the issues of ODP *size* and ODP *import count* were brought up.

The participants initially expressed divergent opinions regarding effect of OWL import statements in ODPs. Participant A considered imports quite helpful in that the reconciliation of imported base- or lower-level concepts with one’s own model provided a good opportunity for validating the soundness of one’s

Table 2. Distribution of fragments to codes.

Code label	Fragments	Code label	Fragments
ODP structure	1	ODP catalogue and selection	28
ODP size	3	ODP effects	3
ODP imports	10	Efficiency	11
ODP complexity	1	Usefulness	13
OE method observations	8	Pattern insufficient	12
ODP method observations	19	ODP usage prerequisites	11
ODPs-as-guidance	21	DL/semantics limitations	8
ODPs-as-error-control	7	Top-down/bottom-up choices	7
ODP-attributable errors	8	Existing implicit ontology effects	1
ODPs-as-ground ontologies	8	Method/metamodel adequacy	6
Modeling errors	1		

own design. He also emphasized the advantage of getting a foundational logic “for free” that one would not otherwise have had time to develop. Participant C expressed an understanding of the tension between reuse and applicability presented by the import feature and large import closures, comparing it to discussions in the OOP design pattern community in the nineties. Participant B criticized the use of imports on the grounds that the expansion of ODP size that such imports imply negatively affects ODP usability, and on the grounds that the base concepts included by imported patterns may be incompatible with one’s own world view, being written for some other purpose:

“I really have to know what is there and what does it mean. And maybe it’s written with some other focus, some other direction, some other goal. And I don’t believe in this general modeling of the universe that fits all purposes.” – Participant B

Participant B also indicated that he would use the idea of a pattern as presented in a pattern catalogue and reimplement it, rather than reuse an existing OWL building block, if that block contained too many imports or dependencies. After some discussions Participant A agreed to the soundness of such a method in the case of a large import closure not directly relevant to the problem at hand. Both participants A and B proposed that a better solution would be to add support for partial imports to tools and standards.

In terms of the size of patterns, the participants emphasized during the interview session the importance of patterns being small enough to be easily understood in a minute or two of study. They considered an appropriate size to be three-four classes and the object- and datatype properties associated with them. They drew parallels to OO design patterns which are frequently of approximately this size. This expressed preference is consistent with the patterns they selected during modeling, all of which contained three or fewer classes.

Recommendations Avoid using imports in patterns unless the imported concepts or properties are necessary for pattern functionality. Support the development of partial import functionality in standards and tools. When possible, develop smaller rather than larger patterns.

5.3 Pattern Selection

It was observed by both researchers present that the single most important variable in ODP selection from the pattern catalogue was pattern naming. If a name “rang a bell” the participants proceeded with studying the pattern specifics to see whether the pattern was suitable in their case. This observation is supported by participant feedback at the interview session. The participants also suggested that description texts and competency questions (formalizations of design requirements as questions that the ODP is able to answer) were important selection criteria that should be emphasized in an ODP catalogue. Additionally, they considered the possible negative consequences of applying a certain pattern to a problem to be of particular importance in selecting and applying patterns.

On the subject of pattern catalogues, the participants indicated that they considered the two catalogues to which they had been exposed (the ODP community portal and the one developed for these sessions) to be unordered and unintuitive, holding patterns of varying completeness, abstraction level and domain, all mixed in one long list. The participants suggested that they would find it easier to navigate a catalogue that was structured according to topic, architecture tier, abstraction level, or some other hierarchy:

“You also know the old classification of upper ontologies, domain ontologies, and task ontologies. You know this old picture. This, at least this structure should be present.” – Participant A

Further participant suggestions for improvements to ODP catalogue usability included the addition of graphical illustration of pattern dependencies, and providing a semantic search engine across ODPs held in the catalogue. The former suggestion was inspired by an illustration from the Core J2EE Patterns web page⁴ that the participants found helpful in deciphering pattern intent, and which Participant C in particular argued would be helpful in understanding the structure of a set of ODP patterns. The latter suggestion was that a search engine be added allowing users to search through concepts and properties present in ODPs in the catalogue, ideally including NLP techniques to match for synonyms and related terms.

Recommendations Ensure that pattern catalogues include complete and consistent pattern metadata, paying particular attention to pattern names, descriptions, competency questions, and negative effects. Ensure that pattern catalogues are structured according to a useful task- or abstraction-oriented hierarchy. Clarify interdependencies between patterns in pattern catalogues. Develop pattern catalogue search engines.

⁴ <http://java.sun.com/blueprints/corej2eepatterns/Patterns/>

5.4 ODP Usage Method

The participants initially developed their designs on a whiteboard rather than on their computers. They used the patterns as guidance in development, rather than as concrete building blocks to be applied directly. When questioned on why this method of working was preferred, they stated that it was more flexible and required less commitment to a design in progress than immediately formalizing to OWL code. The participants would build a prototype solution to a whole problem in one go, rather than tackle one part of the problem at a time. This method is contrary to eXtreme Design, which emphasizes modular development and unit testing. However, the individual critical situations being modeled were rather small and self-contained, and it is uncertain whether this way of working would scale to larger and more complex problem spaces.

The guidance that the participants got out of the patterns appears to be of two types. To begin with, to the extent that patterns provided reasonable solutions to difficult to model problems, the pattern solutions were used as archetypes for own solutions on the whiteboard. This was the most common usage of patterns observed. In the second case, patterns were used to verify the correctness of modeling, by ensuring that the developed solution was consistent with the patterns selected:

Participant B: *"Is a vehicle an agent?"*

Participant A: *"Let's check the pattern!"*

The latter usage was observed both on the whiteboard and later on when attempting to formalize results into OWL files on a computer. In usage, the selected patterns were seen as optimal solutions to problems, and no reflections on the suitability of the patterns in question were observed. On the contrary, in some situations the participants attempted to realign their solutions to available design patterns even when this needlessly significantly increased the complexity of their solution. One example of this is the observed use of the *AgentRole* pattern in categorizing different types of staff, which in the scope of the problem could just as easily have been done via subsumption.

During modeling there were occasions when the work process slowed down, and the participants got caught up in discussions on how to define some very fundamental concepts such as situation, time, event, etc. When questioned, participants expressed a strong preference for such foundational concepts being available as patterns. While a few such foundational patterns have been extracted from DOLCE and made available in the community portal, their documentation is at the time of writing limited.

Recommendations Keep in mind the effects on pattern documentation requirements that arise when patterns are used as guides to development, rather than simply as building blocks. Ensure that common usage mistakes for individual patterns are clearly documented.

5.5 Effects of ODP Usage

Across the two days of working, a noticeable improvement in modeling speed among the participants could be observed. Tasks that in the morning took an hour to complete were in the afternoon performed in fifteen-twenty minutes. While this learning effect cannot be solely attributed to pattern use, the participants indicated that a certain efficiency gain is certainly due to them:

"I think it was helpful, it makes it clearer and furthers reuse, saving time." – Participant A

This efficiency gain was most pronounced when the participants reused patterns which they had already tried once or twice on other problems. The participants also indicated that in order to get the most out of the design patterns, a practitioner needs to have developed some degree of familiarity with them:

"For me it's a new type of modeling [...] but it's understandable, and I can imagine if you know patterns, you are quite faster at inventing everything." – Participant C

As has been mentioned in Section 5.4, the effects of ODP use on the process and resulting ontologies were not all beneficial. In some cases, over-dependence on patterns complicated the resulting ontologies needlessly, and misunderstanding of pattern documentation led to generally strange results. An example of the latter is the modeling of the characteristic "loudness", where the resulting model had time being loudness-indexed rather than the other way around. On the whole however these problems were minor compared to the observed and perceived benefits of ODP usage in guiding modeling.

6 Conclusions and Future Work

The users studied preferred small patterns over large ones, for reasons of understandability. They appreciated the foundational knowledge gained by large import closures, but found the consequent increase in pattern size troublesome, and would prefer partial import functionality if such were to be developed. Suggestions for pattern catalogue improvements include improving catalogue structure and search functionality, and increasing pattern documentation coverage. In order to decrease incorrect pattern usage, it is recommended that common pattern usage mistakes be documented. Finally, patterns were perceived as useful by the participants and the use of them was observed to increase the speed with which tasks were solved.

The author will in upcoming work attempt similar analyses in other cases, to study whether the results presented herein are found to apply to other projects in other domains also. Further, the author suggests that the ODP research community take under serious consideration the results presented herein that pertain to improvements of pattern catalogue structure, and would be happy to contribute to such work in the future.

References

1. Ampatzoglou, A., Chatzigeorgiou, A.: Evaluation of object-oriented design patterns in game development. *Information and Software Technology* 49(5), 445–454 (2007)
2. Anicic, D., Rudolph, S., Fodor, P., Stojanovic, N.: Stream Reasoning and Complex Event Processing in ETALIS. *Semantic Web* (2011)
3. Blomqvist, E., Gangemi, A., Presutti, V.: Experiments on Pattern-based Ontology Design. In: *Proceedings of the Fifth International Conference on Knowledge Capture*. pp. 41–48. ACM (2009)
4. Blomqvist, E., Presutti, V., Daga, E., Gangemi, A.: Experimenting with eXtreme Design. In: *Knowledge Engineering and Management by the Masses*. pp. 120–134. Springer (2010)
5. Burnard, P.: A method of analysing interview transcripts in qualitative research. *Nurse Education Today* 11(6), 461–466 (1991)
6. Eisenhardt, K.: Building Theories from Case Study Research. *The Academy of Management Review* pp. 532–550 (1989)
7. Garrison, D., Cleveland-Innes, M., Koole, M., Kappelman, J.: Revisiting methodological issues in transcript analysis: Negotiated coding and reliability. *The Internet and Higher Education* 9(1), 1–8 (2006)
8. Hammar, K.: The State of Ontology Pattern Research. In: *Perspectives in Business Informatics Research: Associated Workshops and Doctoral Consortium (BIR 2011 Proceedings)*. Riga Technical University (2011)
9. Hammar, K.: Modular Semantic CEP for Threat Detection. In: *Operations Research and Data Mining (ORADM) 2012 workshop proceedings*. National Polytechnic Institute (2012)
10. Iannone, L., Palmisano, I., Rector, A., Stevens, R.: Assessing the Safety of Knowledge Patterns in OWL Ontologies. *The Semantic Web: Research and Applications* pp. 137–151 (2010)
11. Luckham, D.: *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc. (2001)
12. Luckham, D., Frasca, B.: *Complex Event Processing in Distributed Systems*. Computer Systems Laboratory Technical Report CSL-TR-98-754. Stanford University, Stanford (1998)
13. Masuda, G., Sakamoto, N., Ushijima, K.: Evaluation and Analysis of Applying Design Patterns. In: *Proceedings of the International Workshop on the Principles of Software Evolution (IWPSE99)* (1999)
14. Mehmood, K., Cherfi, S., Comyn-Wattiau, I., Akoka, J.: A Pattern-Oriented Methodology for Conceptual Modeling Evaluation and Improvement. In: *Research Challenges in Information Science (RCIS), 2011 Fifth International Conference on*. pp. 1–11. IEEE (2011)
15. Prechelt, L., Unger-Lamprecht, B., Philippsen, M., Tichy, W.: Two Controlled Experiments Assessing the Usefulness of Design Pattern Documentation in Program Maintenance. *IEEE Transactions on Software Engineering* 28(6), 595–606 (2002)
16. Riege, A.: Validity and reliability tests in case study research: a literature review with “hands-on” applications for each research phase. *Qualitative Market Research: An International Journal* 6(2), 75–86 (2003)