The Use of the Grounded Theory Methodology in Investigating Practitioners' Integration of COTS Components in Information Systems

George Allan
Portsmouth University

Follow this and additional works at: http://aisel.aisnet.org/icis2007

Recommended Citation
http://aisel.aisnet.org/icis2007/149

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 2007 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
THE USE OF THE GROUNDED THEORY METHODOLOGY IN INVESTIGATING PRACTITIONERS’ INTEGRATION OF COTS COMPONENTS IN INFORMATION SYSTEMS

George Allan PhD
University of Portsmouth UK
School of Computing, Buckingham Building, PO1 3HE, UK
e-mail george.allan@port.ac.uk

Abstract
This paper challenges the view that Grounded Theory is an unacceptable methodology for IS research lacking rigor and robustness. A demonstration is given of the systematic and rigorous use of the grounded theory data analysis techniques in researching the use of commercial IT components for developing and maintaining business IS. The data analysis processes of coding, conceptualizing and categorizing are shown in action and explained. The techniques of memoing, constant comparison and theoretical coding are shown step-by-step so that the grounded theory methodology can be seen as a systematic framework for IS data analysis. When the methodology is properly applied it provides the structure and rigor needed to reach meaningful conclusions from qualitative data. This research article contributes to the literature showing grounded theory as a powerful research methodology applicable to IS research.

Keywords: grounded theory methodology, COTS components, conceptualization

Introduction
There is a growing use of the grounded theory methodology (GTM) in information systems (IS) research (Allan, 2006; Bryant, 2002; Urquhart, 2001; Urquhart & Fernández, 2006). However, the use of GTM is not without its problems and is viewed with suspicion by academics skeptical of the rigor of the method and repeatability of results and conclusions produced (known as grounded theories). GTM has been perceived by some researchers as a non-robust method for soft data analysis and therefore unacceptable for scientific, empirical research. The aim of this paper is to challenge these views and enable the reader to see that GTM is a rigorous research methodology that provides a set of structured and robust techniques for empirical IS data analysis. This is done by demonstrating the various parts of the methodology in action.

The vehicle for this demonstration is a recent research project investigating IS practitioners’ uses of commercial-off-the-shelf IT components (COTS) for IS development and/or maintenance. In the IS discipline, COTS components are software and hardware artifacts developed by independent companies and offered for sale, lease or license in multiple, identical copies (SEL 1998). COTS components can range from small, simple software modules to large, complex IS application packages.

It should be emphasized that the aim of this paper is to demonstrate and discuss GTM as a research methodology and not to dwell on the philosophy of using COTS components. However, for the benefit of readers unfamiliar with the concept of COTS, a brief explanation of the terminology used is given later in the genesis of the research project.

Reading this paper should benefit IS practitioners who spend their time and effort attempting to integrate COTS components into existing systems because explanations of integration problems are given which will assist those practitioners. The paper will also benefit international IS researchers who are looking to understand how to apply the grounded theory methodology to their work. The paper adds value to the growing but yet still small number of contributions to the literature on grounded theory in the IS discipline.

The paper is structured to give the reader an insight to the systematic approach that Glaser and Strauss (1967) intended for GTM. This is followed by a brief explanation of the genesis of the COTS research project to set the scene for the reader. Details of the three Case Studies are given in the section on empirical data collection. The grounded theory methodology is then applied with step-by-step explanations of the processes. Common problems encountered in GTM are discussed and special attention is given to conceptualizing and memoing.
Finally, the epistemology of the grounded theory methodology is summarized. The next section gives an outline of GTM as proposed by Glaser and Strauss.

**Grounded Theory Methodology**

GTM is well covered in the literature (Glaser and Strauss 1967; Glaser 1978; 1998; Strauss and Corbin 1994; 1998; Dey 1993; Charmaz 2001; Allan 2006) but an outline of the four fundamental analysis techniques may assist readers unfamiliar with the methodology to understand the data analysis. The four analysis techniques are open coding, constant comparison, memoing, and theoretical coding. Each is a separate analysis method in its own right and each performs a distinct function necessary to the researcher’s eventual understanding of what the data mean. The following sections describe the four techniques, how they are performed and how they contribute to data analysis.

**Open Coding** - the data are analyzed piece-by-piece with an ‘open mind’ looking for the underlying conceptual issues. Codes are linguistic labels selected by the researcher for each underlying conceptual issue found in the data and it is good practice to note each code alongside the data from which it emerged. Open coding means analysing data (in this case interview data from interviewee responses) sentence-by-sentence to uncover the conceptual issue or issues underlying the incident reported in each sentence. GTM coding also demands that the researcher follows a framework of six rules all the way through the coding (Glaser 1978).

- **Rule 1** – ask these three questions frequently during the research: What is this data a study of? What concept or category does this incident indicate? What is actually happening in this data? This will continually remind the researcher of the original research intentions and prevents him/her to stay in focus without getting lost amongst masses of data. It also helps to identify the social, structural and psychological processes driving the events.

- **Rule 2** – analyse data sentence-by-sentence not paragraphs, nor pages. This prevents the researcher attempting to analyze too much data in one go which could lead to important conceptual issues being missed. As the analysis reveals a code it is immediately compared with previous codes in the process known as ‘constant comparison’ (see next section).

- **Rule 3** – do your own coding with no preconceived codes. This rule emphasizes the need for an unbiased mind.

- **Rule 4** – interrupt coding often to write memos. Rule 4 promotes the researcher’s thinking and clarification of ideas and emergent concepts.

- **Rule 5** – stay within the substantive area until the core variable is saturated. This maintains focus on the research topic.

- **Rule 6** – ignore face-sheet variables which have no bearing on the research. This rule warns us to avoid distractions that have nothing to do with the real research data, such as names, contact details and other administrative information.

Rule 2 above refers to the important analysis technique of **constant comparison** which is a corner-stone of GTM and is explained in the next section.

**Constant Comparison** - as each code emerges it is compared with all previous codes one at a time as the researcher looks for similarities, connections and patterns. Codes that share commonalities are grouped together representing a higher order of causality known as ‘concepts’. The constant comparison method is applied again later in the analysis comparing concept with concept to identify commonalities at the concept level, resulting in higher order abstractions known as ‘categories’. In any complete research study there should not be more than about 6 categories: this is a matter of granularity for the researcher to maintain the focus of the research.

The category with the largest number of connections to other categories becomes the core theme of the research and becomes the ‘core category’. Coding is now focused on this core category and the process becomes selective for this ‘substantive variable’ (Glaser and Strauss 1967; Glaser 1978; 1998). At this stage, analysis would be considered similar to other qualitative data analysis in that the researcher is now looking for relevant data related to a specific issue. The mind is no longer open but is searching for evidence to support a specific idea.

**Memoing** - the analyst writes reflective notes on codes, concepts, categories and the connections between them as they emerge. The practice of reflection is a key technique for reaching a clearer and deeper understanding of the underlying issues in the research. Memoing is used extensively and throughout the analysis. The numerous memos written during this research cannot be included in this text but an example is given to illustrate the technique. Memos are helpful in the constant comparison technique when connections are not obvious. However, with deeper thought facilitated by the memoing process, a hidden connection may be uncovered as relevant to the research.
**Theoretical Coding** is the exploration of the relationships between the core category and other categories. It occurs after all coding is finished in the Glaser and Strauss approach. By again applying the techniques of constant comparison between the core category and each of the other categories, and writing a memo on each relationship, more and more discoveries will emerge about the core variable and an explication of why events happened will grow in substance. This explication is known as a grounded theory of that research area. The explication is substantive only in the area of that research area. Generalizability of application to the world at large requires another step explained in the next section.

**Generalizability**

To make a particular grounded theory more generalizable it has to be applied in areas outside the field of the original study. For instance, a grounded theory about the difficulties IS practitioners have when integrating COTS components could be applied to the automobile industry where components are purchased from suppliers and integrated on an assembly line. This will require amendments to the grounded theory for a broader perspective on component integration. A grounded theory about the difficulties clients have in articulating their IS requirements could be applied to a study of sales persons trying to elicit and understand customer’s requirements in buying a product such as an automobile, a new home or a holiday abroad thereby increasing the grounded theory’s generalizability. The amended grounded theory could then be further applied to higher education where students try to guess their professor’s requirements embedded in examination questions, or marketing departments attempting to ascertain the needs of prospective employers and students. Further refinements to the grounded theory make it more robust and applicable in general.

The main focus of this paper is the application of the above steps as a research method and to do this it may be helpful to set the scene by explaining the genesis and background of the research project. This is done in the next section.

**Genesis of the Research Project**

In the 1980s and 1990s many business organizations outsourced responsibility for supplying their IT needs (Classe 1996; Caldwell et al. 2001; CIO 2004). Software houses recognized opportunities to develop specialist software packages for niche markets. For example in heath-care, software packages were developed for operating dental surgeries; for patient record systems in hospitals; in the legal profession for conveyancing real-estate; planning packages for IT project management; resource planning in the manufacturing industry. Software suppliers developed software components (packages or modules) that would satisfy the common requirements within a specialist area. Their strategy was to design once and build multiple copies of the product and market these as commercially available. These became known as commercial-off-the-shelf (COTS) products.

**Benefits from Using COTS Components**

The benefit to IS practitioners is the immediate availability of tested components to build/maintain computer-based systems. The benefit perceived by business people is the ability to develop or maintain their MIS faster at reduced costs because COTS software components have already been through the stages of design, build and test and are immediately available for use. This avoids the problems and delays associated with developing bespoke software. The phrase plug and play (Boehm and Abts, 1999; Oberndorf et al, 2000) acts as a motivator to use COTS components because of the implied immediacy of availability.

**COTS Terminology**

The practice of using COTS components in software engineering is known as component-based software engineering (CBSE) and systems developed entirely of COTS are known as COTS-based systems (CBS). It is arguable that the term COTS component in CBSE could apply to any self-contained software application that is commercially available and will be integrated into a larger system. Examples could be a complete financial system, or an enterprise resource planning system (ERP) but this is not the accepted meaning in CBSE. In this paper, the term COTS is taken to mean small to medium sized, self-contained software components. The research project was concerned with the use of COTS software components in the development of a new IS or the maintenance of an existing IS.

The term maintenance is usually applied in IS to four areas: support maintenance that attempts to make a system run as intended; adaptive maintenance that moves a system from one environment to another; corrective maintenance for the removal of bugs; enhancement maintenance that adds new functionality to a system after installation. COTS components are usually used in two of these, adaptive maintenance and enhancement maintenance and it is these forms of maintenance that were the focus of this research and will be used in this
paper. Therefore, this article will be of interest to practitioners involved in developing MIS using COTS products and to those implementing COTS components into an existing MIS to adapt or enhance the system.

The research question can be formed as “How and why using COTS components assist developers/maintainers?” The research design and data collection and analysis methods are discussed in the following sections.

Empirical Data Collection

A quantitative survey of IT Project Managers via a postal questionnaire and statistical data analysis would establish how often the benefits were/were not realized but would not necessarily establish how and why the claimed benefits affected development and maintenance projects. A method was needed to collect and analyze in-depth data about the claimed benefits of using COTS products. The data exists in the heads of IS practitioners with relevant experience in using COTS components and would be qualitative in nature. The Case Study (Yin 1994) was considered an appropriate research design using interviews for data collection, coupled with GTM to provide a framework for data analysis. A single Case Study would provide limited information and possibly introduce bias; therefore multiple Case Studies were considered a more reliable design. The scope of this research was constrained to three European companies for financial and practical reasons. However, future research may extend this boundary and add further cases. The following sections outline the three Case Studies to set the scene for the GTM analysis.

Case Study A was with a large European Air Traffic Control Centre. The organizational structure was bureaucratic with rigorously defined management levels. The company organizational chart showed distinct functional areas and a strict reporting hierarchy. COTS products were used in a multi-million pound development involving the large-scale integration of approximately 200 powerful mainframe processors into a safety-critical IS. Data collection was over fifteen months by 15 interviews with IS Managers, IS practitioners and 3 focus-group meetings with the integration team members. The IS Manager had 15 years experience of integrating COTS components in the organization. Interviews were spread over a twenty-month period with the IS Manager only as access to other working staff was prohibited by this organization.

Case Study B was with the IS department of a medium-sized European organization focused on the business of service provision and customer billing. The management structure was hierarchical but the IT department culture tended to be more organic. The IT department used a project approach to IS work units and each project used a matrix structure to accomplish tasks. There were 342 IT staff supporting 16 discrete interconnected computer systems. The IS Manager had 15 years experience of integrating COTS components in the organization. Interviews were spread over a twenty-month period with the IS Manager only as access to other working staff was prohibited by this organization.

Case Study C was in the European branch of a multi-national organization that manufactured equipment containing large amounts of COTS software. The organization management structure was divisional but had a large number of managerial levels which tended to lead to bureaucracy within a division. This Case Study was concerned with one part of the company, known as the European IS Division, of approximately 1000 staff who were well aware of functional boundaries. When a prototype had been developed and accepted, the MIS Division of approximately 500 IT staff assembled many replicas for customers. Data collection was over a twelve-month period by nine interviews with the Divisional IS Manager and the IS Configuration Manager. Interviews took the form of facilitated discussions in which the interviewee was encouraged to enter into a discourse on their experiences. An example of how a discourse was initiated would be “We are talking about the use of COTS components. Could you tell us of your experiences of using them?” This is an open question which does not lead the interviewee down any particular route but leaves the way open for them to enter into a discourse of their own making. The researcher may have to facilitate from time to time to bring that discourse back nearer the focus of the research bearing in mind the rules of open coding (Rule 1 in particular). Information about COTS components was freely forthcoming and enthusiastically volunteered by the practitioners.

As soon as any data gathering interview starts, the researcher is usually faced with a stream of words from the interviewee. Some of these words will be fill-in phrases, where the interviewee is buying thinking time while generating a response, some will be thoughts unconnected to the research focus and some will contain key information. The researcher has to distinguish between these three and pick out the key information that is directly in line with the research focus. These form the Key Points in the data collection. Key points made in each Case Study are indicated as PAx where P signifies a Key Point, the subscripts A, B or C indicate which Case Study the Key Point came from, and x shows the sequential identification number of the Key Point.

A key point can also be tangential information volunteered by the interviewee on which he/she holds strong views even though this may appear (to the researcher) to be unconnected to the research focus. Strongly held
views will influence the participant’s work and therefore have effects unapparent, at this stage, to the researcher. It is the Key Points that form the focal data for analysis. This is described in detail in the next section.

**Data Analysis**

Key Points are open-coded as they arise during the interview by the experienced GTM researcher by attaching conceptually meaningful labels that indicate (as best as can be done at the time) underlying conceptual issues. Open coding can be continued after the interview when replaying recordings or viewing transcripts but the reality of the moment, such as attitude and body language of the interviewee, may be lost. A less experienced researcher will need the recordings and the transcripts in order to open-code in slower time. The following is an actual analysis from interviews.

**Data Analysis from Case Study A**

<table>
<thead>
<tr>
<th>Id</th>
<th>Key Points</th>
<th>Open Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA1</td>
<td>“COTS software from external suppliers can be integrated directly into our system.”</td>
<td>Integration</td>
</tr>
<tr>
<td>PA2</td>
<td>“We need to prevent uncontrolled insertion of COTS software”</td>
<td>Controlling insertion</td>
</tr>
<tr>
<td>PA3</td>
<td>“There is a 3-to-5 year strategic plan to use COTS products in development projects.”</td>
<td>COTS strategy</td>
</tr>
<tr>
<td>PA4</td>
<td>“We take a system-wide approach to COTS.”</td>
<td>COTS strategy</td>
</tr>
<tr>
<td>PA22</td>
<td>“and these lead to great difficulties when we try to implement these COTS components.”</td>
<td>Implementation difficulties</td>
</tr>
<tr>
<td>PA37</td>
<td>“We need to somehow prevent third-party developers inserting COTS software …”</td>
<td>Controlling insertion Losing control</td>
</tr>
<tr>
<td>PA41</td>
<td>“A form of Suitability Evaluation is used to compare several offered COTS and select the best value for money for our purposes.”</td>
<td>Evaluating COTS Selecting COTS</td>
</tr>
<tr>
<td>PA42</td>
<td>“… there is a process with a scoring system to evaluate COTS products and their fitness-for-purpose.”</td>
<td>Evaluating COTS Selecting COTS Fitness for purpose</td>
</tr>
<tr>
<td>PA43</td>
<td>“… we rely on our procurement department to match COTS specifications to actual requirements.”</td>
<td>Evaluating COTS Selecting COTS</td>
</tr>
<tr>
<td>PA45</td>
<td>“COTS products are selected prior to our boundary of responsibility.”</td>
<td>Selecting COTS</td>
</tr>
<tr>
<td>PA46</td>
<td>“Strategic decisions made by the executive management are passed onto IT department for action. There is no COTS implementation strategy.”</td>
<td>Implementation strategy</td>
</tr>
</tbody>
</table>

Table 1: Key points and open codes from the grounded theory analysis of COTS data in Case Study A

From PA1 the open code of *integration* emerges. From PA2 the open code *controlling insertion* emerges. These two open codes are immediately compared (constant comparison technique) to see if they lead to a concept on a higher level of abstraction. A GTM memo could be started here to assist the researcher to see any connection. All memos should start with the title of one of these codes, for example *INTEGRATION*, and connections that occur to the researcher are noted. The following is an example.

**Memo on INTEGRATION**

The fact that external suppliers can integrate COTS components directly implies a loss of implementation control by the IS practitioners. This *loss of control* could be a new code and may lead to other implementation issues being uncovered.
Early memos will tend to be short as there is little data to discuss at this point. Later memos will be much fuller and richer in conceptualizations. The title is important for future use in the constant comparison technique.

$P_3$ codes as $COTS$ strategy and $P_4$ gives $COTS$ strategy for the second time and therefore reinforces this code. These may have an influence on the three established codes depending on how that strategy is internalized and implemented by the IS practitioners. Another memo should be started here entitled STRATEGY and added to later as more information becomes available in the research.

$P_37$ reinforces the code controlling insertion. A memo established another concept as losing control over the integration process. The emergence of a concept called Controllability from these codes is demonstrated diagrammatically in Figure 1.

![Figure 1: Diagrammatical emergence of the concept Controllability](image)

The analysis continued and from $P_{22}$ comes the code of implementation difficulties. This is compared with the previous codes and the concept controllability. From $P_{41}$, $P_{42}$, $P_{43}$, $P_{45}$ come the concepts of evaluating COTS and selecting COTS. When these are coupled with codes of $COTS$ strategy, Implementation difficulties (from $P_{22}$) and implementation strategy (from $P_{46}$) we arrive at the concept of Implementing.

![Figure 2: Diagrammatical emergence of the concept Implementation issues](image)

This shows the three GTM techniques of open coding, constant comparison and memoing in action. Data analysis is continued in this way allowing concepts to emerge from the data in the first interview. Before analysis of the second interview, a number of concepts are already established, so the researcher will be looking for evidence to support or contradict these while analysing the new evidence with an open mind.

As the analysis progressed into Case Study B, the code of selecting was found in $P_{33}$, $P_{34}$, $P_{49}$ and $P_{50}$ reinforcing the concept of Implementation issues. Key Points $P_{18}$, $P_{20}$, $P_{25}$, $P_{36}$ supported the concept of Implementation issues showing a substantiation of this concept. Other codes emerged leading to new concepts such as Vendor support (lack of) and Documentation (poor) which established the concept of User Dissatisfaction. These serve as examples of how GTM produces codes and concepts.

Analysis of the three Case Studies revealed a total of 236 Key Points resulting in 20 meaningful concepts. Constant comparison was used to group and re-group these at a higher level of abstraction known as categories. Four main categories emerged as COTS practices, Vendor issues, User dissatisfaction and Implementation difficulties (Figure 3).

Notice that the category Implementation difficulties started as a code labeled implementation difficulties from $P_{22}$, then became the concept Implementing, then the category Implementation difficulties and finally was nominated the core category. This is an example of how important issues come from roots grounded in the data and shows where GTM gets its name because explanations in the final write-up are grounded in the data.
Emergent Categories.

- Implementing
- Modifications
- Selecting
- Upgrading
- Controllability

COTS practices

- Replicating (not always identical)
- Documentation (poor)
- Fitness for purpose
- Incompatibility
- Supplier’s understanding
- Obsolescence
- Upgrading

Vendor issues

- Changing specifications
- Vendors understanding (lack of)
- Obsolescence
- Vendor support (lack of)
- (poor) COTS documenting
- Communications (between vendor and implementer)
- Replicating (not always identical)

Figure 3: Emergent Categories Derived from the GTM Analysis of Using COTS Components
The final GTM technique is that of Theoretical Coding in which relationships between the core category and the other categories are investigated, again using constant comparison. Memoing leads the researcher to an explication of the main conceptual issues in the research, in other words the findings of the research that are essential material in the write-up. The core category in the COTS component research was *Implementation difficulties* and a memo was written about the relationship between this and *User dissatisfaction* drawing out several important issues. For example:

“Implementation difficulties are the core problem in using COTS components for IS development or maintenance. All other problems stem from this. The relationships between implementation difficulties and other categories indicated that time over-run; cost over-spend; doubts of IT managerial operational practices; vendor difficulties; vendor unreliable specifications and unexpected change were properties of implementation difficulties. The vendor difficulties had sub-properties of vendor unawareness; vendor poor responses; vendor poor documentation; vendor lack of communication and vendor assumptions all of which need further attention by both management, IS practitioners and the vendors themselves.

Replicates of COTS products are not always identical and when this is coupled with the unreliability of current COTS documentation and difficulties in communicating with vendors, the result is over-runs in implementation time and cost over-spend. These reflect badly on the apparent capabilities of the IT operational management in the eyes of business customers and leads to internal organizational tensions.” Etc etc etc giving a full grounded theory of using COTS components for IS development and maintenance.

At this point it is timely to remember that the main focus of this paper is not on the actual COTS research project (that acts here as a vehicle for demonstrating GTM) but is on the grounded theory methodology itself. The next section covers some of the problems researchers may have in trying to apply GTM to their own research.

**Common Problems Encountered in GTM**

**The Term ‘Grounded Theory’**

A major weakness of GTM is that the end product is called a *grounded theory* when it is in fact an explication of the research findings. To many researchers and most practitioners, the word ‘theory’ implies a set of ideas (however arrived at) that need to be tested by controlled application to a number of carefully constructed or well known scenarios in the real world and the results carefully monitored. Data is collected and analyzed in order to verify or falsify that theory. After applying the grounded theory methodology to any area of research (in this case it was the use of COTS components in developing or maintaining information systems), the resulting conclusions come from the very data which the above researchers would now start to collect and analyze. Grounded theories come from real data and do not need to be verified by that data. However, they can and should be applied to other data areas and then be extended to be applicable in general.

**Conceptualizing**

Probably the most difficult and least understood part of GTM is the processes of forming concepts. Conceptualizing is inextricably bound with the open coding of Key Points where an initial attempt is made to encapsulate the Key Point in as few words as possible or a single word code. The concept of *coding* involves ‘encapsulating’ the essence of a Key Point in a single word if possible rather than describing it in a sentence. The concept of *conceptualizing* involves ‘encapsulating’ the essence of a group of codes. Gerunds are useful as concepts because they are nouns turned into verbs and usually end in “–ing”, such as *selecting* which encapsulates the processes and strategy in making selections. A concept is an empirical generalization (Becker, 1998) found by grouping several instances together and examining what is left when the particularities are removed.

A good qualitative researcher will see the key points in her/his research data and write descriptively to contribute to knowledge. A grounded theorist is trained to see beyond description by conceptualizing the essence of the research and reach deeper understandings of the issues. *Conceptualizing* is difficult and the process needs to be practiced and developed by the researcher until the process becomes easier.
Memoing

During this analysis many memos were constructed. These were subjected to the constant comparison technique and new memos were written on what these groupings brought to light. Much in-depth thinking occurs when writing GTM memos and the contents are rich in write-up material. Writing memos is a strategy for writing-up the research.

The thinking processes underpinning memo writing build up knowledge of the conceptual issues underlying the research. This is the epistemology of grounded theory analysis and is an integral part of the methodology. The GTM technique of memoing was used to investigate the connections between emergent categories. Writing a conceptual memo on every category makes the researcher think about the relationships and connections between the categories (Glaser and Strauss, 1967; Glaser, 1978; Glaser, 1998). Comparing these memos with each other, revisiting and revising them in the light of newly discovered understandings of the issues underlying the research, leads to clarification of the conceptual relationships. It is from these conceptual relationships between categories that theory emerges from the data.

Epistemology of GTM

In many quantitative research reports, conclusions are a descriptive interpretation of the test results. Much qualitative research is let down because of poor conclusions. In GTM the conclusions are more conceptual than descriptive and follow naturally from writing about the relationships between the emergent concepts and categories.

The technique of constant comparison forces the researcher to think conceptually about relationships which may have been otherwise ignored or simply described. This forces deeper thinking and assists the construction of knowledge in the mind of the researcher which, ultimately, contributes to knowledge through the write-up of the final analysis.

In most qualitative research methods the benefits of using COTS components would have been the focus of the investigation and data collection would have been structured around questions relating to those benefits. This may well have drawn ‘proper-line’ responses (Glaser and Strauss 1967; Glaser 1978; Glaser 1998) which are responses in line with company policy, or responses that interviewees suppose that researchers expect to hear. GTM with its deliberate lack of preconceived notions avoids this situation. In this particular research, the category of benefits remained isolated which indicated that benefits had less to contribute than the other categories found. The main finding revolved around Implementation difficulties which proved to be nearer the realities faced by the IS practitioners. The conclusions from the actual research project were an enlightenment to the IS practitioners who took part.

Conclusion

There have been calls for IS academic research to be relevant to IS practitioners’ needs (Galliers 1994; Zmud 1995; Applegate and King 1999; Benbasat and Zmud 1999; Saunders 1999). This research article attempted to satisfy this and the allied call by Orlikowski and Iacono (2001) for greater consideration of the IT artifact by reporting research where the IT artifacts are COTS components used by IS practitioners who develop and maintain management information systems (MIS) in business organizations.

This research and its findings are in line with the calls for IS researchers to give greater consideration to the information technology (IT) artifacts (Orlikowski and Iacono, 2001; Weber, 2003) which, in this case, are the COTS components used by the IS practitioners who develop or maintain the management information systems in organizations. The research is also in line with the call to align academic research relevance with the needs of IS practitioners (Applegate and King 1999; Benbasat and Zmud 1999; Galliers 1994) which in this case is the need to use COTS components in IS development and maintenance.

Grounded theory is a methodology appropriate for IS research. The methodology consists of a systematic framework that, when followed, provides techniques for data analysis that are repeatable, generalizable and more rigorous than most qualitative research methods. GTM can be classed as a qualitative research method but it has certain processes that are lacking in other qualitative methods and could be considered to be the first in a new genre of research methodologies conceptualizing underlying causal issues rather than describing those issues.
References


Weber, R. “Still Desperately Seeking the IT Artifact”, MIS Quarterly (27:2), June 2003, pp. iii-iv