Association for Information Systems
AIS Electronic Library (AISeL)

UK Academy for Information Systems Conference Proceedings 2015

UK Academy for Information Systems

Spring 4-1-2015

Grounded Ontology Methodology – Illustrating the Seed Ontology Creation

Syed Irfan Nabi
Institute of Business Administration, Pakistan, syedirfannabi@gmail.com

Zaheeruddin Asif
Institute of Business Administration, Pakistan, zasif@iba.edu.pk

GROUNDED ONTOLOGY METHODOLOGY – ILLUSTRATING THE SEED ONTOLOGY CREATION

Syed Irfan Nabi and Zaheeruddin Asif
Institute of Business Administration, Main Campus, University Road, Karachi, Pakistan
Email: {snabi, zasif}@iba.edu.pk.

Abstract

This paper is an extension of a paper that suggested Grounded Ontology (GO) as a new methodology of ontology engineering. It adds an example of application of first two stages of GO Methodology to create an initial (seed) ontology to a summarized discussion from another paper on Grounded Ontology (GO) Methodology. Its efficacy in deriving entities and their relationships directly from the data along with ontologization is illustrated through a step-by-step example. The GO Methodology proposes that 'a domain ontology developed using text-coding technique contributes in conceptualizing and representing state-of-the-art as given by published research in a particular domain.' The motivation behind GO Methodology is to make the state-of-the art available to the researchers of a particular domain and help them come to common understanding through an ontology. Ontology developer are given a leading role by the existing ontology engineering methods. This has led to a general observation regarding dominating influence of personal perspective of ontology developer and/or expert on the resultant ontology. However, if coding of data is done such that entities and their relationships are directly obtained from and are closely linked to the text of the published research, the resultant ontology stands a better chance of being unbiased. Therefore, a new methodology (Grounded Ontology - GO) was proposed for deriving an ontology directly from text of published research. Such and ontology will not only help in bringing forth the research already done by other but can also help in highlighting areas where new research efforts are needed.

Keywords: Ontology Engineering, Grounded Theory, Text Coding, Common Understanding, Intended Meaning, Unbiased

1. Introduction

What is ***Ontology?*** There are many descriptions of this term; what we are concerned with is a conceptual representation of a domain of interest showing entities and their relationships. . The determination of entities and their relationships is usually considered to be a subjective exercise in design by the ontology expert. The GO Methodology proposes that text-coding techniques can be effectively employed in the design phase of ontology development in order to furnish a more objective representation of a domain of interest as seen in published research.

The GO (Grounded Ontology) is based on text coding techniques developed in Grounded Theory Methodology (GTM) and uses them to extract entities and their relationships from published research on a given domain of interest. As Gruber (1995) points out conceptualization is a process of constructing a simplified view of the world that we wish to represent for some purpose. In other words it is "an abstraction over domain of interest in terms of its conceptual entities and their relationships" (Hepp 2007).

Ontology provides a way of combining and consolidating knowledge in a domain. (Chandrasekaran, Josephson, &Benjamins, 1999; Gómez-Pérez &Benjamins, 1999; Gruber, 1991, 1993; Guarino, 1995; Noy & McGuinness, 2001). An ontology also helps in developing a mutually agreed upon understanding of a domain by providing a common lexicon. (Basile, 2011; Chandrasekaran *et al*., 1999; Ćosić, Ćosić, & Bača, 2011; Harter & Moon, 2011). The current methods of ontology development suffer from personal biases that inevitably creep in as the determination of entities and their relationships is a creative exercise undertaken by the designer of ontology. The resultant ontology reflects the personal understanding and the background experience of the domain expert.

However, if the ontology is derived directly from the text of published research papers in that domain through coding (Charmaz, 2006; Strauss, 1987), it stands a better chance of being biased toward an individual perspective. Saldana (2009) describes a type of coding called **In-vivo** coding where exact terms are taken from the text and used as codes, which may further be regarded as entities. This process isolates coder's perspective from the emerging codes. The advantage of using **in-vivo** coding is that the resultant categorization of entities follows more closely the structure of entities found in the literature. This process is recognized as being similar to ontology engineering (Kuziemsky, Downing, Black, & Lau, 2007; Urban, 2009).

The objective of this paper is to illustrate the application of the GO Methodology proposed by Nabi and Asif (2014)to create seed ontology as a proposed solution to the above criticisms.

The rest of the paper is organized as follows. The next section discusses ontology, starting with fundamental concept of ontology going through its purpose and concluding on it usage. The subsequent section is about ontology engineering. It discusses existing methodologies and their limitations. The next section describe a possible solution to overcome these limitations through the use of text coding. Subsequently the Grounded Ontology (GO) methodology is described along with a step-by-step example to illustrate seed ontology creation through this methodology. The paper concludes with limitations of this paper, the GO Methodology and future research directions.

2. Ontology

As discussed in Nabi and Asif (2014), ontology is a "specific artifact expressing the ***intended meaning*** of a ***vocabulary*** in terms of ***primitive*** categories and relations describing the ***nature*** and ***structure*** of a ***domain of discourse"*** Guarino (2012).

Infomation scientists use "ontology" to express a shared taxonomy of entities that has been reduced to its simplest and most significant form possible without the loss of generality (Smith, 2003). "An ontology is in this context a dictionary of terms formulated in a canonical syntax and with commonly accepted definitions designed to yield a lexical or taxonomical framework for knowledge-representation which can be shared by different information systems communities" (Smith, 2003).

Thus, it can be concluded that ontology is a conceptual system of the domain of interest representing entities and their relationships in the universe of discourse.

2.1. Purpose of Ontology

As Nicola Guarino (2002) holds, the primary purpose of an ontology is a mutual understanding of each other and improved communication among people (Jasper & Uschold, 1999; Sowa, 2013). The focus of ontologies is on the content, i.e. on the meaning being conveyed by the entities as well as on the structure of the domain they represent (Fensel, 2001; Guarino, 2002). "The content [that ontologies represent] must be studied, understood, [and] analyzed", however, it must be remembered that understanding of content is not contingent upon its representation (Guarino, 2012).

For the purpose of human to human communication an informal specification is preferred over a strict and formal specification. (Jasper &Uschold, 1999; Uschold, 1998).

Since not everyone possess extensive knowledge of formal logic, an informal participation by users is sufficient to define domain elements in an informal way and supported by a well thought out vocabulary and carefully chosen terminology. The importance of human readable documentation cannot be overemphasized (Hepp, 2007).

As discussed in Nabi and Asif (2014), information systems perspective of ontologies is focused on meaning and understanding conceptual elements and their relationships. In this context "a collection of named conceptual entities with a natural language definition would count as an ontology" (Hepp, 2007).

The above discussion clearly shows that for human to human communication and ontology defined using informal but unambiguous vocabulary is not only sufficient but also preferable.

2.2. Use of Ontology

There are many applications of ontology in the areas of computer science and information systems. Researchers in these areas agree that ontologies let us capture commonly agreed (Chandrasekaran *et al*., 1999) relevant information (Guarino, 1995). The knowledge encapsulated in an ontology is available for sharing and reuse (Gruber, 1993), and can be segregated into domain and operational knowledge (Noy & McGuinness, 2001).

3. Ontology Engineering – Some Limitations

Currently, there are no universally accepted methods of ontology engineering (Gómez-Pérez & Benjamins, 1999). Additionally, the implementation of various engineering methodologies introduces further variations in the resultant ontologies (Gómez-Pérez & Benjamins, 1999).

3.1. Ontology Engineering

Casellas (2011) holds that ontology development could be classified as top-down, bottom-up, and middle-out approach based on where the process begins. It could also be organized on the level of automation: manual, semi-automatic, and fully-automatic. There could be other ways of classification as well. Choosing a particular methodology is an important decision since among others, one of the ways to characterize an ontology is the methodology used to develop it (Casellas, 2011).

3.2. Limitations of Current Ontology Engineering Methodologies

Current ontology engineering methodologies have certain limitations. When developers/experts design ontologies based on their personal understanding and background experience, they reflect individual biases.

Sometimes statistical and syntactical techniques coupled with Artificial Intelligence are used to derive ontologies. However, there are as yet no fully-automatic methodologies for ontology development.

Another limitation is that ontologies are usually not dynamic, which means that with time they become obsolete.

In summary, the following list describes some common limitations of existing ontology engineering methodologies:

1. Focused primarily on systems interoperability and computer-computer interaction.
2. Reflect ontology engineers'/experts' personal understanding of the domain.
3. Require human interventions to make the resultant ontology meaningful and useful.
4. Evolution of ontology for dynamic domains remains a challenge.

4. Possible Choice of Overcoming These Limitations

A possible way of overcoming the above limitations is to code the text directly. The following section reproduces an introductory paragraph on text coding published in Nabi and Asif (2014).

4.1. Text Coding

According to Strauss and Corbin (1998), textual data can be coded and analyzed to find concrete description of abstract categories. Among other sources, historical data is used to establish relationships between categories and their descriptions. This technique is based on 1967 work of Glaser and Strauss (1967). It is a "discovery methodology that allows the researcher to develop a theoretical account of the general features of a topic while simultaneously grounding the account in empirical observations or data" (Martin & Turner, 1986). Constant comparison is an important rigorous "tool" for scrutiny of the codes and gathering of analytical insights (Urquhart, Lehmann, & Myers, 2010). It is about discovering concepts, categories and relationships among them (Bryant & Charmaz, 2007). This methodology has clearly defined data analysis procedure, which results in elaborate and novel findings that are substantiated by data (Orlikowski, 1993). Thus, one of the outputs is a list of emergent concepts, categories and sub-categories, and their properties derived directly from the text.

Grounded Ontology (GO) is based on coding text directly from top peer-reviewed journals. This selection will help in making the resultant ontology more acceptable and relevant. This methodology not only reduces ontology engineer's bias but it also helps in consolidating domain knowledge.

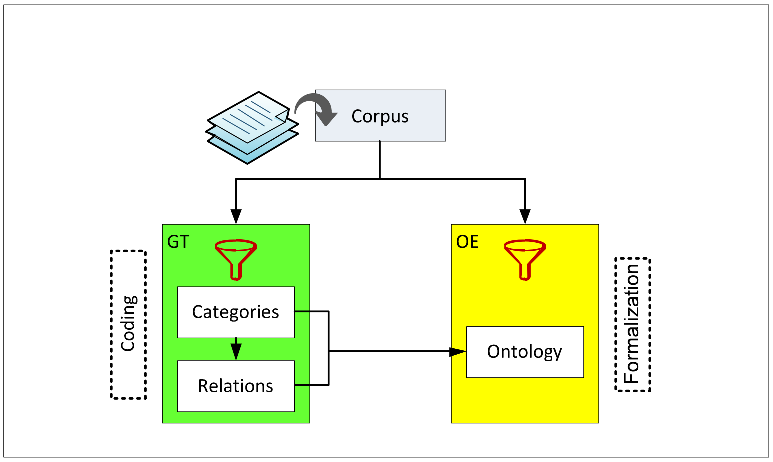

Figure 1: Use of Grounded Theory Method and ontology engineering for creating an emergent ontology (Nabi & Asif 2014).

5. Concept of GO Methodology

GO methodology is a "multi-stage multi-step knowledge summarization and representation process". It is used to organize and exhibit knowledge in a concise manner. The methodology creates ontology through discovery involving codifying existing knowledge.

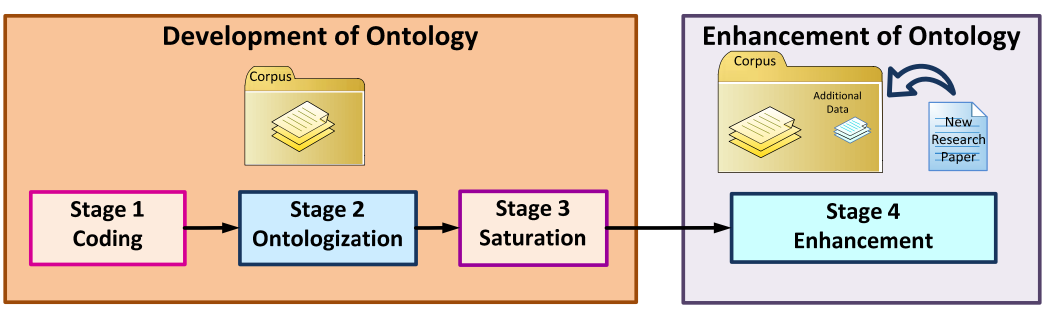

Figure 2: Stages of ontology development and enhancement (Nabi & Asif 2014)

As discussed in Nabi and Asif (2014) , the methodology is organized in four stages shown in Figure 2. Stage 1 is coding of the text in the corpus. Stage 2 is giving a structure to the categories and relationships emergent from the codes and creating seed ontology. Stage 3 is finding other categories and relationship and incorporating them in the seed ontology to form a saturated ontology. Stage 4 is the ongoing enhancement to the saturated ontology. It is done by adding more data (research papers in this case) to the corpus and processing the additional data through stage 1 coding and merging the additional categories and their relationships to form an enhanced version of the ontology. This stage 4 can be run as and when more data becomes available.

GO methodology makes use of the fact that most important entities in the text can be found in the specific significant portions of the paper. So instead of coding the entire paper, it focuses on the key sections of the paper, such as abstract, introduction and conclusion, first. This results in the generation of a seed ontology through **in-vivo** coding technique. Subsequently, this seed ontology is enhanced to make core ontology through **selective** coding of the relatively less significant sections of the text. In GO methodology, for seed ontology the abstracts are coded using **in-vivo** technique. Conclusions are coded using **selective** coding technique. Discussions and results may also be coded subsequently through **selective** coding technique if deemed necessary.

6. Generating Seed Ontology Through GO Methodology

Following is an example of seed ontology creation by applying first two stages of GO methodology, shown in Figure 3 to illustrate the process of extracting the entities and their relationship and ontologization. A small data set consisting of a paragraph with three sentences is used as corpus as the purpose is only to illustrate the application of GO methodology and not to build an ontology. This paragraph, given below is taken from Boss, Kirsch, Angermeier, Shingler, & Boss (2009) paper on user behavior and information security. The assumption is that papers from well-reputed journals of a domain are more rigorously peer-reviewed and ontology extracted from such publications would lead to greater acceptability by the domain experts. In the example the GO Methodology is applied manually.

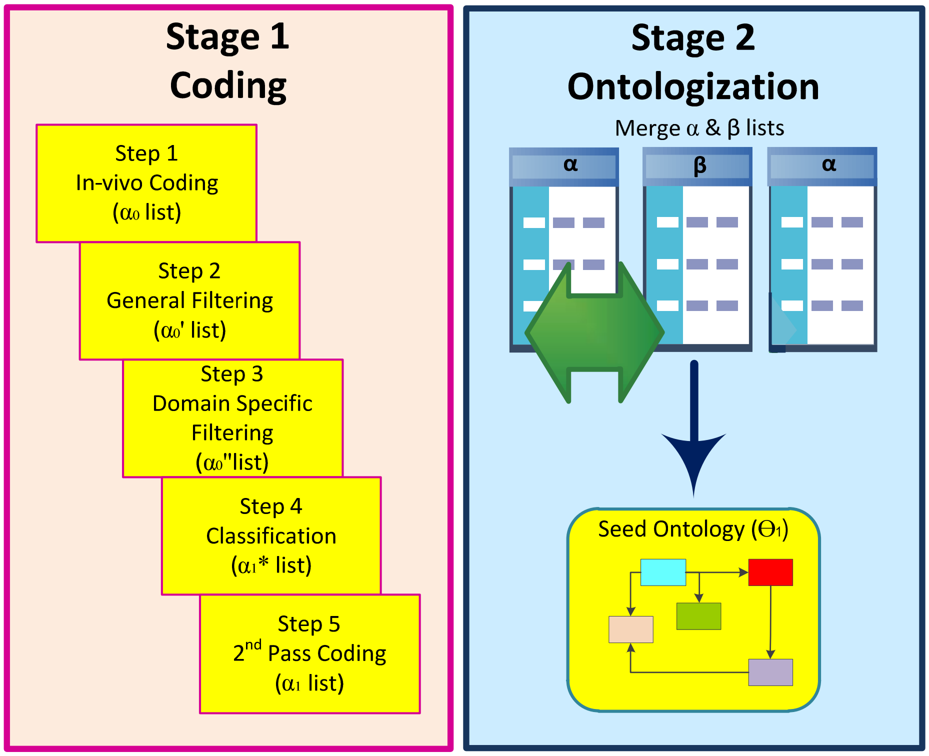

Figure 3. **Stage one and two of GO Methodology with detailed steps.**

Original Text:

> Information security has become increasingly important to organizations. Despite the prevalence of technical security measures, individual employees remain the key link – and frequently the weakest link – in corporate defenses. When individuals choose to disregard security policies and procedures, the organization is at risk.

6.1. Stage 1: Coding

Step 1: In-vivo coding

In this step all possible candidates (nouns) are coded (extracted) from the text for eventually selecting appropriate and relevant entities. The list is called alpha naught (α_0) list. **In-vivo** codes (i.e. nouns), highlighted in the original text are shown below:

Information security has become increasingly **important** to **organizations**. Despite the **prevalence** of **technical security measures**, **individual employees** remain the **key link** – and frequently the weakest **link** – in **corporate defenses**. When **individuals** choose to **disregard** **security policies** and **procedures**, the organization is at **risk**.

Result of **in-vivo** coding (α_0 List) is given in Table 1. It is alphabetically ordered to make it easy to process is manually.

α_0 List - Based on Occurrence	α_0 List - Alphabetically Order
Information security	Corporate defenses
Important	Disregard
Organizations	Important
Prevalence	Individual employees
Technical security measures	Individuals
Individual employees	Information security
Key link	Key link
Link	Link
Corporate defenses	Organizations
Individuals	Prevalence
Disregard	Procedures
Security policies	Risk
Procedures	Security policies
Risk	Technical security measures

Table 1. Result of in-vivo coding

Step 2: General filtering

Initial alpha naught (α_0) list is examined to filter out any common nouns that may not be of much use as entities. The resultant list is called alpha naught prime (α_0') list.

In general filtering two words '***important***' and '***prevalence***' have been **excluded** from the **all inclusive** (α_0) list since they are general use words and may not be of much used as entities. The resultant list is called alpha naught prime (α_0').

Step 3: Domain specific filtering

Alpha naught prime (α_0') list is examined to exclude nouns found to be unrelated to the topic or domain of interest. This examination is done with sensitization to the domain specific information taken from existing literature, including any ontologies. The intent is to remain focused and to be more effective while avoiding system overload with trivialities. The sensitization is dependent upon the purpose of the ontology and will help determine the boundary. The resulting list is domain specific and is called alpha naught double prime (α_0'') list.

It is important to note that the automated methods of picking important terms employing term-frequency (*tf*) and inverse-document-frequency (*idf*) may result in the most significant terms based on statistics. Such lists are always dependent upon expert scrutiny for relevance checking. A complete framework for such ontology development has been given by Abulaish *et al*. (2011). In contrast we have adopted a manual method where experts themselves make the decision, therefore it is likely to result in a more meaningful and relevant list as discussed elsewhere.

Excluding nouns found to be unrelated to the topic or domain of interest results in a list called alpha naught double prime (α_0'') list. In this step it was found that '**key link**' and '**link**' are not related to the domain of interest. Similarly, singular noun, '**organization**', can provide the required understanding in reference to the context, as its plural form. Therefore, these are filtered out.

Step 4: Classification

Alpha naught double prime (α_0'') list is now analyzed to find and categorize the nouns into entities, attributes, and classes (*type-of* and *part-whole*). Similar meaning codes are consolidated into single codes. This is also done with consideration to domain sensitization. This initial cataloging is called alpha one star (α_1^*) classification list.

In alpha naught double prime (α_0'') list the terms are taken directly from the text as it is and might include both British and American spellings which would initially count as separate terms. For example **organisation** (British spelling) and **organization** (American spelling) would be two different terms. In the alpha one star (α_1^*) classification list these two terms would be consolidated into one term.

The initial cataloging is called alpha one star (α_1^*) classification list. In reference to information security, '**corporate defenses**' are about securing information and can be generalized as '**information security**'. Similarly, '**individual employees**' and '**individuals'** can be expressed as '**individual'**. The word '**procedures**' in the text refers to security procedures and can be generalized as '**security policies**'.

In this example '**corporate defenses**' can be taken as defensive measures in effect at an organization. In this sense, '**procedures'**, and '**security policies**' can both be *part-of* it.

Step 5: Second pass coding

The 2nd Pass Coding is for validation of classification. Also, plural terms are converted to singular terms, unless this significantly changes the meaning. This can be called manual stemming[1]. In this process initial classification alpha one star (α_1^*) list is compared with the original text to validate the classification. The validated classification list is denoted as (α_1) list.

The validated list of classified entities (α_1) is given in Table 2. During the 2nd Pass Coding it was found that '**security policies**' can form *part-of* '**technical security measures**'.

α_1^* **List**	**Validated List (α_1 List)**
Disregard	Disregard
Individual	Individual
Information security	Information security
Organization	Organization
Risk	Risk
security policies	*Technical security measures*
Technical security measures	

Table 2. Validated classified list

6.1.1. Stage 2: Ontologization

Ontologization stage is about establishing relationships between validated entities and presenting them in a graphical form. Possible relationships between validated nouns (α_1) are established by reviewing the original text to find verbs relating these nouns to each other. This results in a list of possible relationships between classified nouns. The relationships list is called beta one (β_1) list. Both the validated entities (α) list and relationships list (β) are subscripted with numbers denoting the versions. Thus, 1 stands for initial version while subsequent versions are denoted as 2, 3, and so on. This results in different versions of α and β lists i.e. (α_2, β_2), (α_3, β_3), …, (α_n, β_n), where n denotes nth version.

[1] Stemming process reduces inflected/derived terms to their stem (base or root word) – Taken from Information Retrieval/Linguistic Morphology.

Although mentioned separately, beta (β) lists do not exist independently. Practically, relationships are appended to the pair of nouns between whom the relationship exists.

The alpha and beta lists are merged together to form the initial ontology (seed ontology) listing the entities and their corresponding relationships. This initial ontology is represented as theta one (θ_1).

For this example, with a valid consolidated list of entities (α_1) available, the next stage is to find relationships from the text that can be appended to the pair of nouns (entities) between whom the relationship exists. The relationships (β_1) between the entities in α_1 list are given in Table 3.

Entity (α_1)	**Relationship (β_1)**	**Entity (α_1)**
Information security	Has Link to	Individual
Information security	Has Part	Technical security measures
Information security	Provide Security To	Organization
Individual	Can	Disregard
Disregard	Causes	Risk
Risk	Is to	Organization

Table 3. Entities and relationships between them

The alpha and beta lists are merged together to form the initial ontology (seed ontology). All the entities and relationships given in Table 3 are used to form this ontology. This seed ontology represented by theta one (θ_1) is shown graphically in Figure 4.

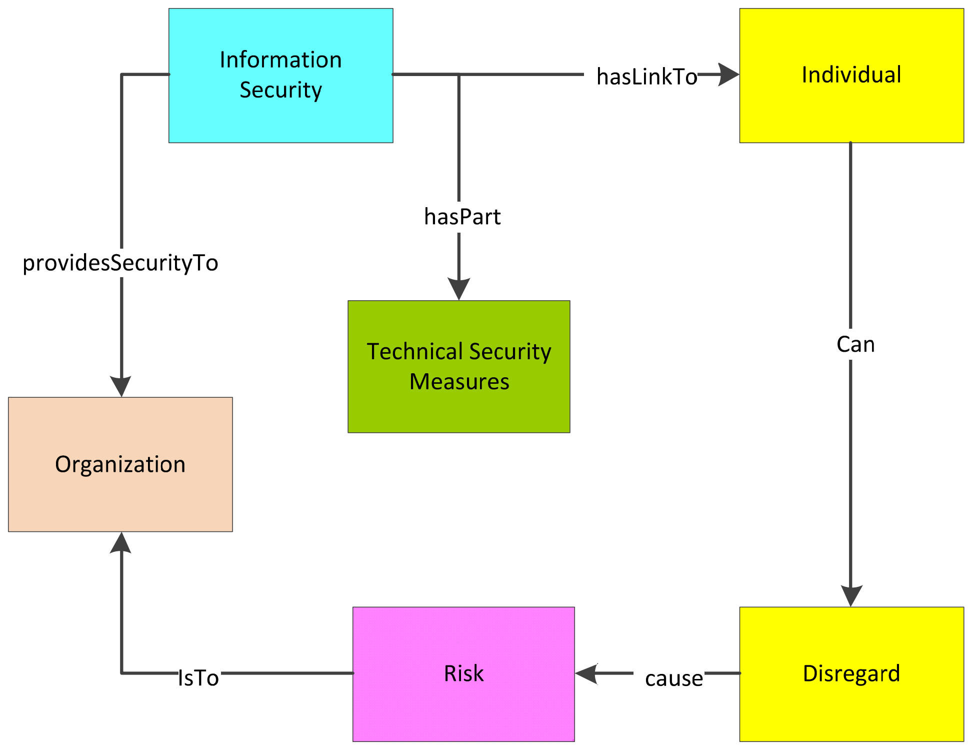

Figure 4: Initial user behavior information security ontology derived from sample text using GO methodology

6.2. Comparison of GTM Usge in Vaious Methodologies

There is a difference in how GTM has been used and applied in GO Methodology as compared to its use and application by Kuziemsky *et al*. (2007) and Urban (2009). The difference between GO Methodolgy and that of Kuziemsky *et al*. (2007) is given in Table 4.

S No.	Characteristics	Kuziemsky *et al*.'s (2007)	GO
1	Coding Technique	Open, Axial and Selective	In-vivo and Selective
2	Purpose	Better understanding of domain	Presenting state-of-the-art in domain.
3	Information Sources	Practice experience of health care professional, patients' health management charts, and research literature	Research papers from journals

Table 4. Comparison of GT methodology applied by Kuziemsky *et al*. (2007) and GO (Nabi & Asif, 2014).

While it was pointed out by Urban (2009) that information may be better analyzed for greater understanding if GTM is used, Kuziemsky *et al*. (2007) actually used GTM for enhancing their understanding of the domain through the use of **open**, **axial** and **selective** coding techniques. However, **in-vivo** and **selective** coding techniques are used in GO for bringing forth the state-of-the-art in a particular area.

6.3. Overcoming Limitations of Existing Ontology Engineering Methodologies Through GO Approach

Following are the ways in which the four limitations of existing ontology engineering approaches as mentioned in Section 3.2 are addressed by GO Methodology, as mentioned our previous paper.

Limitation of Computer-Computer Interaction

GO Methodology is designed to enhance the understanding of a domain and conveying that to other human beings. The ontology thus developed uses simple natural language to improve understanding even by domain experts not proficient in mathematical or philosophical logic. At the same time it is ensured that the intended meaning is not lost. Thus, the limitation that ontology caters primarily for computer-computer interactions is taken care of by **GO.**

Limitation of Personal Understanding

To ensure that the intended meaning of the original author is not lost in the personal perspective and understanding of the ontologist during the process, **in-vivo** text-coding technique is used. Further, it is ensured that all entities can be back-tracked and located in the original text.

Limitation of Human Intervention

A methodology for ontology development that is fully automatic without any human interaction is yet to be achieved practically. So GO method uses **in-vivo** and **selective** coding of only the principal sections (namely **abstract** and **conclusion**) to reduce the amount of text required to be coded. Thus, reducing the effort of human expert in ontology development.

Limitation of Evolution of Ontology

Ontology development using GO methodology takes care of evolution as it is derived from published research, where the publication process ensures continued evolution and enhancement of knowledge.

6.4. Advantages of GO Methodology:

Not only has the GO methodology potential to overcome the limitations as mentions above, but it also has a few advantages:

1. Using the published research to derive the ontology helps in promptly comprehending state-of-the-art in a domain. Further, if combined with the **FocalPoint;** a proposed mechanism of continual evolution of an ontology by Nabi *et al.* (2013), it can take care of continued evolution knowledge in a fast changing domain. This can also help in scaling up the ontology development process.

2. It can help in finding new vistas of research by helping researcher know what is already done and what still needs to be researched.

3. It can provide common lexicon not only to enhance understanding but also a mechanism to resolve any existing misunderstanding among researchers.

7. Limitations and Future Research

Current paper illustrates only the first two stages of step on of GO Methodology starting from sample text, extracting relevant entities and their relationships and concluding at creating an initial ontology also called seed ontology. The process of saturation and enhancement are more applicable when a real ontology is developed and shall be addressed in the future research.

One of the limitations of the current application of the GO Methodology is the extraction of adjectives along with nouns. Although the methodology proposes extraction of nouns only. Thus, there is a need to look into this issue and resolve it.

Some of the other issues and limitation of GO Methodology as discussed in Nabi & Asif 2014 are:

- That different ontologies can emerge if different codes/categories are extracted by different ontologists. To overcome this the intended meaning of a researcher must be adhered to. If the original researcher is not available, prominent researchers of the domain may be consulted to come to a consensus. If multiple point of views still exist then all the views may be incorporated in the ontology. This will help keep the ontology validated.
- An in-built limitation of GO methodology is that it cannot be applied to unstructured text. Use of naïve Bayes classifier to provide an option to use unstructured text can be taken up in future.

In conclusion we would like to mention that the actual use and acceptability of a new proposed methodology is perhaps the criteria to judge its efficacy of GO and its importance to community. Thus, we have provided a step-by-step example to illustrate the application of GO Methodology for creating seed ontology. In future we intend to illustrate the application of GO Methodology to enhance and evolve an ontology.

8. References:

Abulaish, M., Nabi, S. I., Alghathbar, K., & Chikh, A. (2011b). SIMOnt: A Security Information Management Ontology Framework. In J. J. Park, J. Lopez, S.-S. Yeo, T. Shon, & D. Taniar (Eds.), *Secure and Trust Computing, Data Management and Applications* (Vol. 186, pp. 201–208). Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-22339-6_24

Basile, C. (2011). *Security Ontology Definition* (Deliverable No. D3.2). Europe: European Community. Retrieved from http://www.posecco.eu/fileadmin/POSECCO/user_upload/deliverables/D3.2_SecurityOntologyDefinition.pdf

Boss, S. R., Kirsch, L. J., Angermeier, I., Shingler, R. A., & Boss, R. W. (2009). If someone is watching, I'll do what I'm asked: mandatoriness, control, and information security. *European Journal of Information Systems*, *18*(2), 151–164. doi:10.1057/ejis.2009.8

Bryant, A., & Charmaz, K. (2007). Grounded theory in historical perspective: An epistemological account. *The SAGE Handbook of Grounded Theory*, 31–57.

Casellas, N. (2011). *Legal Ontology Engineering* (Vol. 3). Springer Berlin / Heidelberg.

Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What are ontologies, and why do we need them? *Intelligent Systems and Their Applications, IEEE*, *14*(1), 20–26.

Charmaz, K. (2006). *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. Pine Forge Press.

Ćosić, J., Ćosić, Z., & Bača, M. (2011). An Ontological Approach to Study and Manage Digital Chain of Custody of Digital Evidence. *Journal of Information and Organizational Sciences*, *35*(1), 1–13.

Fensel, D. (2001). *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, Berlin.

Glaser, B. G., & Strauss, A. L. (1967). *The discovery of grounded theory: strategies for qualitative research*. Transaction Publishers.

Gómez-Pérez, A., & Benjamins, R. (1999). Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods. Retrieved from http://oa.upm.es/6468/1/Overview_of_Knowledge.pdf

Gruber, T. R. (1991). *The role of common ontology in achieving sharable, reusable knowledge bases*. Knowledge Systems Laboratory, Computer Science Department Stanford University. Retrieved from http://reference.kfupm.edu.sa/content/r/o/the_role_of_common_ontology_in_achieving_127677.pdf

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, *43*(5), 907–928.

Guarino, N. (1995). Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human Computer Studies*, *43*(5), 625–640.

Guarino, N. (2002). Ontology-driven conceptual modelling. In *Proc. of the 21st International Conference on Conceptual Modeling, LNCS* (Vol. 2503). Retrieved from http://www.cs.ioc.ee/adbis2005/downloads/OntologyDrivenCMTallin.pdf

Guarino, N. (2012, July 16). *Introduction to Applied Ontology and Ontological Analysis*. Lecture at IAOA Summer School on Ontological Analysis, room A206, “Fabio Ferrari” Hub, University of Trento,Trento, Italy.

Harter, A. G., & Moon, B. M. (2011). Common Lexicon Initiative: A Concept Mapping Approach to Semiautomated Definition Integration. In B. M. Moon, R. R. Hoffman, J. Novak, & A. Canas (Eds.), *Applied Conceptf Mapping: Capturing, Analyzing, and Organizing Knowledge* (1st ed., pp. 380, pp 133–134). CRC Press, Taylor & Francis Group.

Hepp, M. (2007). Ontologies: State of the Art, Business Potential, and Grand Challenges. In M. Hepp, P. D. Leenheer, & A. D. Moor (Eds.), *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications* (pp. 3–22). Springer.

Jasper, R., & Uschold, M. (1999). A framework for understanding and classifying ontology applications. In *Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW* (Vol. 99, pp. 16–21). Retrieved from http://folk.ntnu.no/alexanno/skole/WebInt/Articles/Articles.pdf

Kuziemsky, C. E., Downing, G. M., Black, F. M., & Lau, F. (2007). A grounded theory guided approach to palliative care systems design. *International Journal of Medical Informatics*, *76 Suppl 1*, S141–148. doi:10.1016/j.ijmedinf.2006.05.034

Martin, P. Y., & Turner, B. A. (1986). Grounded Theory and Organizational Research. *The Journal of Applied Behavioral Science*, *22*(2), 141 –157.

doi:10.1177/002188638602200207
Nabi, S. I., & Asif, Z. (2014). Grounded Ontology - A proposed methodology for emergent ontology engineering. *Business Review*, *9*(2), 119 – 128.
Nabi, S. I., Asif, Z., Iradat, F., Arain, W., & Ghani, S. (2013). FocalPoint- Proposed Grounded Methodology for Collaborative Construction of Information Systems Security Ontologies. *Information - An International Interdisciplinary Journal*, *16*(3 (A)), 2063–2074.
Noy, N. F., & McGuinness, D. L. (2001). *Ontology development 101: A guide to creating your first ontology*. Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880. Retrieved from http://liris.cnrs.fr/alain.mille/enseignements/Ecole_Centrale/What%20is%20an%20ontology%20and%20why%20we%20need%20it.htm
Orlikowski, W. J. (1993). CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development. *MIS Quarterly*, *17*(3), 309–340.
Saldana, J. (2009). *The Coding Manual for Qualitative Researchers*. Sage Publications Ltd.
Smith, B. (2003). Ontology. In L. Floridi (Ed.), *The Blackwell guide to the philosophy of computing and information* (pp. 153–166). Wiley-Blackwell. Retrieved from doi:10.1002/9780470757017.ch11
Sowa, J. (2013, April 27). Ontologies for human-human interactoin. *IAOA-member*. Reply. Retrieved from http://ontolog.cim3.net/forum/iaoa-member/
Strauss, A. L. (1987). *Qualitative Analysis for Social Scientists*. Cambridge University Press.
Strauss, A. L., & Corbin, J. M. (1998). *Basics of qualitative research: techniques and procedures for developing grounded theory* (2nd ed.). Sage Publications.
Urban, R. J. (2009). Blended Methods for Ontology Development. Presented at the 2009 ALISE Annual Conference, Denver, CO, USA: Association for Library and Information Science Education (ALISE). Retrieved from http://www.alise.org/index.php?option=com_content&view=article&id=196#51
Urquhart, C., Lehmann, H., & Myers, M. D. (2010). Putting the “theory” back into grounded theory: guidelines for grounded theory studies in information systems. *Information Systems Journal*, *20*(4), 357–381. doi:10.1111/j.1365-2575.2009.00328.x