Requirement Meta-model and Ontology-based Blogs Architecture for IS Requirements Elicitation

Heng-Li Yang, Chi-Lun Liu
MIS Dept., National Cheng-Chi University
yanh@nccu.edu.tw, tonyliu@ms4.hinet.net

Abstract

Collecting requirements from large number of online stakeholders is one of success factors of information systems on Internet. However, unstructured and fragmental requirements are difficult to be seamlessly integrated. Distributed stakeholders are difficult to deliver and discuss requirements in an effective and efficient way. This paper proposes a meta-model to structuralize requirements. This paper argues that stakeholders can use ontologies to fill in the meta-model to composite a requirement summary. Based on the proposed meta-model and ontologies, this paper proposes an improved blogs system to assist stakeholders to send, receive, and manage requirements and ontologies. A system architecture is depicted. Finally, a scenario as an illustrative example is also provided to explain how the proposed system architecture works.

1. Introduction

Most organizations have information systems (IS) that provide various services on the Internet for a large number of diverse on-line users on different locations. However, requirements are always changing (Nissen et al., 1996). Stakeholders may give a lot of viewpoints on a requirement or problem (Andrade et al., 2004). Hence most organizations have to face a large number of functional requirements for each problem.

Gathering requirements information has been recognized as the most difficult task in system development processes (Browne and Rogich, 2001). Two difficulties about gathering requirements are relevant in this research (Browne and Rogich, 2001, Davis, 1982): communication and content structuring problem. The communication problem results from complex dialogue patterns among stakeholders. The content structuring problem is due to variety and complexity of requirement contents without a formal structure. Therefore developing a tool to assist communication and content structuring in requirement gathering process is necessary.

Blog is a kind of mainstream communication tools (Rosenbloom, 2004; Young and Terrence, 2003). Many people like to post multi-media articles on their blogs in daily life. Blogs can automatically arrange articles according to date (Lindahl and Blount, 2003). Hence blog can be applied to assist requirement gathering. However, in nowadays, blog could not support stakeholders to build meta-model or meta-data to structuralize content. Although traditional blogs already have a partial metadata mechanism, it is limited to RSS (Really Simple Syndication) offering subscription function. Besides, if establishing conceptual models to organize blog article content, it could facilitate effective access to the collective knowledge (Cayzer, 2004).

A structured requirement not only needs a meta-model but also requires well-organized terms to fill in the meta-model. Ontology, which contains well-organized terms, provides a framework for sharing a meaning of symbols exchanged during communication (Maedche et al., 2003). Ontologies can be used to represent domain knowledge. Organizations can incrementally establish domain ontologies which can give meanings to requirements contents (Kaiya and Saeki, 2006).

This study argues that selected terms from evolving ontologies as a summary appending to blog articles can organize the contents and further facilitate communication among stakeholders in requirement elicitation process. The goal of this study is to propose a meta-model of requirement summary and the architecture of ontology-based blogs. The ontology-based blogs apply a free software package and adds new modules for assisting stakeholders to produce conceptual models and communicate with others. The proposed system architecture has three sub-goals: (1) assisting stakeholders to acquire relevant information

as a foundation to express their functional requirements; (2) assisting stakeholders to compose and deliver requirements; (3) assisting the administrator to manage received requirements and terms of ontologies.

2. Meta-model and ontologies

2.1. Meta-model of requirement summaries

The summary attached to a functional requirement document should contain important elements. According to use case diagram in unified modeling language (UML) (Fowler and Scott, 2003) and the implications for functional requirements from hermeneutics, figure 1 proposes the six elements to constitute the meta-model of a functional requirement summary. This meta-model have two parts: the functional part and the contextual part. The functional part including four elements can directly represent some roles use some functions to support some activities. Roles represent actors in use case diagram. Functions, indicating tools that are mentioned in hermeneutics (Rathswohl, 1991), can be systems, subsystems, or classes. Use cases can be separated into activities and data. Activities are tasks supported by functions. Data is a kind of materials which are involved in activities and manipulated by functions.

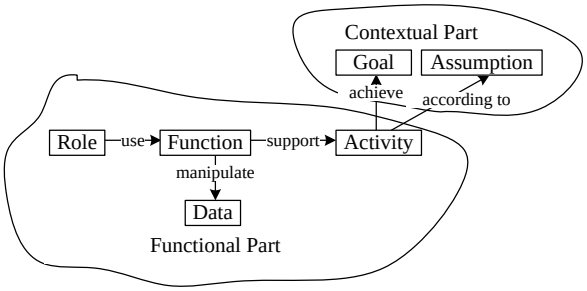

Figure 1. Meta-model of Functional Requirement Summaries

The contextual part has two other elements to represent contexts surrounding the functional part. An activity always has its preconditions and contexts (Fowler and Scott, 2003). Any activity should have its intended goal. Hence activity contexts can be twofold: goal and assumption. An activity inevitably has its own aim to be achieved. And behind an activity, there are possibly a number of assumptions. The information can assist stakeholders to recognize necessary and appropriate functional requirements caused by contexts.

This meta-model organizes the six elements and their relationships in requirement summaries. But the terms, which are used to fill in the elements, should be also well-organized. Ontologies are a feasible theory to organize terms in requirement summaries.

2.2. Ontologies for filling in the meta-model

Ontologies contain a set of shared vocabulary. The terms in ontologies can represent specific concepts. The semantic relationships between terms in ontologies can be antonymy, hypernymy, hyponymy, holonymy, meronymy, and synonymy. The stakeholders can use terms in ontologies to describe situations about IS. These shared terms in ontologies are used for representing and mapping to pieces of functional requirement summaries. In detail, stakeholders can choose existing common terms from the ontologies or build new terms to add notes on functional requirement articles or, in other words, to fill in requirement summaries. According to the proposed elements of functional requirements, there are six kinds of ontologies explained as follows.

1. Role: In business environment, roles mention different sizes and levels of organization units. For example, a company consists of groups, such as departments and project teams. A group usually consists of various positions which are responsible for specific works. And external users or consumers may have different types such as VIP members and common customers. Besides, aliases of formal role names may be built in the ontologies.
2. Function: This ontologies represent what functions constitute software systems. In general, software systems consist of several kinds of parts, such as subsystems, modules, business logics of web pages, or components. These parts have interconnections between them in requirements. For example, a component is reused by other components or called by a web page.
3. Activity: Activity ontologies describe tasks valuable for stakeholders. Activities always constitute of sub activities and so on. In other words, concepts in activity ontologies may have parent and child concepts. Besides, processes are kinds of activities that constitute of a sequence of stages. Synonyms for a same activity can be built.
4. Data: Using IS functions to support activities always deal with data. Data can be classified as input data and output data. Input data ontologies describe what field and what kinds of data should be input to functions. Output data ontologies describe what data should be output.
5. Goal: Behind activities, goals can explain why activities should be existed and necessary. Different

roles do some activities for some goals. For example, all company activities have to achieve high-level organization goals, such as earning profit and increasing sales volume. Departments and project activities have to achieve not only organization goals but also group goals, such as system analysis and design which is responsible by a system development team. Position activities are kinds of group activities possess specific position goals and have to achieve group and organization goals. When stakeholders wants a function supports an activity to achieve different goals, their requirements may have conflicts.

6. Assumption: Assumptions are always behind activities to guide how activity works and what kinds of functions is necessary. Assumptions possibly include organization and consumer cultures, government politics, laws, as well as strategies, management and control manners, and Standard Operation Procedures (SOP). Some assumptions appearing in requirements may result in conflict, such as a selfishness culture.

These terms in ontologies can not only uncover the static space but also should describe the dynamic space with time. Hermeneutics reveals that human memories can reflect situations of worlds include the future, present, and past simultaneously (Heidegger, 1962). It implies that the future is the oncoming circumstances which may partially be expected or planned by humans. The past may partially influence humans' standpoints at present. Therefore the concepts used to fill in the requirements should be added notes of time axis to represent which concepts are planned in requirements documents, which concepts describe current situations, and which concepts now become the history in the reality.

The proposed meta-model and ontologies transform unstructured requirements into structuralize and well-organized requirement summary content. The next step is to propose a system architecture to assist stakeholders to use the meta-model and ontologies to represent requirement summaries. The proposed system also use the meta-model and ontologies to assist stakeholders to communicate and collaborate with each others.

3. Ontology-based blogs architecture

The proposed ontology-based blogs include front-end and back-end systems. The aim of front-end system is to assist stakeholders to receive and post structure summaries for functional requirements. The aim of back-end system is to assist the administrators managing requirements and relevant ontologies.

3.1. Front-end system architecture

Figure 2 depicts the architecture of the front-end system. There are three layers in the architecture: interface, business logic, and storage. The front-end system has a web interface to provide stakeholders with a simple access way for using several modules about requirements communication and content structuring.

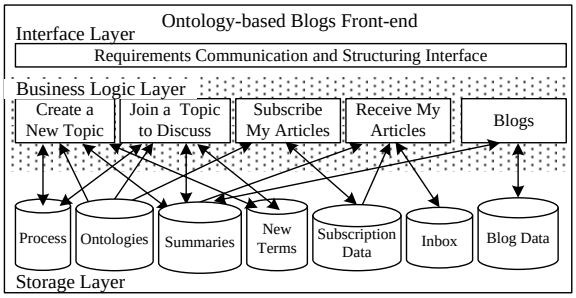

Figure 2. Proposed font-end system architecture

The business logic layer of the front-end system has five modules: crate a new topic, join a topic to discuss, subscribe my articles, receive my articles, and blogs. Overall, stakeholder can use create a new topic model and join a topic to discuss module to post their requirements. Stakeholder can use "subscribe my articles module" and "receive my articles module" to read relevant discussing information. The five modules are described as follows.

1. Create a new topic: Stakeholder can use this module to create a new discussion room about a specific topic of functional requirements. Creating a new topic has to indicate what the issue or topic is. There are two kinds of issues. The first issue may be concerned with the contextual part including goals and assumptions. If a stakeholder brings a contextual concept which may have been indicated by another, the module will remind the stakeholder that it may be redundant. If the contextual concept or its parent or child is connected to an activity and the activity or its child concept is connected to a function, the module can consequently remind relevant stakeholders that some new possible requirements about related activities and functions existed. The second issue may come from the functional part to bring new ideas. To describe a new general idea as an initial functional requirement, stakeholders can specify concepts in four elements, including the role, operated function,

supported activity, and manipulated data, to represent a concept mode of the idea. If a stakeholder is not sure what the concept of function could be mentioned, the stakeholder can optionally specify a role, activity, or data to find an existing function. If a stakeholder brings up an issue that already existed, this module can remind stakeholders. Besides, the module can indicate related requirement documents posted in the past and stakeholders can use these documents as reference materials.

2. Join a topic to discuss: Stakeholders can use this module to browse what topics are under discussions, and then join a specific discussion. There are two ways to join a discussion: offering a requirement, and giving a comment. Offering a requirement is to submit a concept model of a new requirement. This module can remind stakeholders that which subjects have been mentioned in this discussion room to facilitate generating functional requirements. If stakeholders submit a concept model which may be already appeared in this discussion room, this module will remind them. Giving a comment is to criticize existing requirements in this discussion room. For the concepts in the contextual part of requirement summaries, stakeholder can judge whether are true or not. For the concepts in the functional part, stakeholder can completely or partially agree with or oppose the concepts in summaries. And system analyst or designer can judge whether a requirement is fully or partially feasible or not. Furthermore, this module can detect conflicts if a function should support several activities in order to achieve different goals.
3. Subscribe my article: Stakeholders can use this module to subscribe articles that contain interesting concepts mapping to ontologies. This module lists concepts in ontologies and stakeholders can choose those interesting concepts to subscribe requirement articles that mention these concepts or child concepts. In default, the articles, which contain the functions used by a specific role, have been subscribed by stakeholders who play this role.
4. Receive my article: Each stakeholder can use this module to receive the subscribed articles. If new related articles appear, this module can also notify relevant stakeholders through e-mail.
5. Blogs: A blog software package would be integrated as a module in the proposed system. Each requirement summary ID should correspond to a blog article ID. Using trackback mechanism can link requirement summary and blog article.

Table 1 summarizes seven rules in the front-end business logic layer. In general, there are three kinds of proposed rules: prompting relevant requirements (rule 1, 3, 4, 7), suggesting possible relevant concepts for the proposing requirement (rule 2), and detecting requirement conflicts (rule 5, 6).

Table 1. Rules in front-end business logic layer

	If…	Then…
1	The concept of the new topic *j* appears in topic *k*	Topics *j* and *k* may be redundant
2	Concept *x* or its parent or child is connected to concept *y* in the existing requirement(s)	Concept *y* may be relevant to the requirement
3	The concept in proposed requirement *p* already appeared in the existing requirement *q*	Requirement *q* should be referenced while discussing *p*
4	All Concepts in requirement *p* already appeared in requirement *q*	Requirements *p* and *q* may be redundant
5	There is a opposing comment *c* for a concept in requirement *p*	There is a conflict between comment *c* and requirement p
6	Function *f* support any activities to achieve different goals	There is a conflict for designing function *f*
7	Concept *x* within new requirement *p* is subscribed by role *m*	Role *m* receive requirement *p*

The storage layer is a database and has 7 kinds of data used for this system: process, ontologies, summaries, new terms, subscription data, inbox data, and blog data. These data are explained as follows.

1. Process: Each new requirement discussion topic would initiate a new process. The processes data includes topic, executed stage, scheduled/actual start time and end time of stages. Stakeholders would use the module of "create a new topic" to propose issues as the first stage. Using the module of "join a topic to discuss" can execute the rest of two stages: stakeholders proceed to propose detailed requirements, and system analysts describe business blueprint.
2. Ontologies: "Ontologies" data includes trees (containing concepts, represented terms, and relationships between concepts). Besides, requirement summaries can make interconnections among trees. "Create a new topic" and "join a topic to discuss" module can assist stakeholders to search

and choose terms in ontologies to frame requirement summaries. “Subscribe my articles” module can assist stakeholder to subscribe concepts including the child concepts in ontologies.

3. Summaries: Each blog article has a corresponding functional requirement summary. “Summaries” data include role, function, activity, data, goal, and assumption terms that can map to ontologies. Stakeholders can use “create a new topic” and “join a topic to discuss” module to create summaries. The “receive my article” module would use summaries data to identify which articles are subscribed.
4. New terms: “New terms” data includes term name, established date, author, type specified by author (e.g., role, activity, and goal), and status. The status can be “waiting to be handled”, “already moved to ontologies”, and “dropped”. Stakeholders can create new terms when they use “create a new topic” and “join a topic to discuss” module.
5. Subscription data: It stores each stakeholder’s interested concepts. The requirement articles referring to these interested concepts are subscribed by stakeholders. Stakeholders can use “subscribe my article” module to access this subscription data.
6. Inbox data: The subscribed article IDs and subscriber IDs are stored here. “Receive my article” module uses summaries and subscription data to determine which articles are interested and store these article IDs to inbox data.
7. Blog data: It stored in a packaged blog system. In general, blog data mainly includes member login data, article categories, article text, and content management data.

3.2. Back-end system architecture

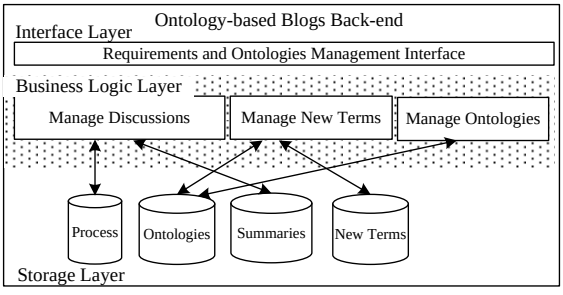

Figure 3. Back-end Architecture of the Proposed Blogs

Figure 3 depicts the back-end system, a management console. The administrators of the ontology-based blogs can use the interface layer to manage requirements and ontologies. The business logic layer includes three modules described as follows.

1. Manage discussions: Stakeholders can use this module to manage topics, discussion processes, and articles. Managing topics is to edit the topic titles and to delete garbage and useless topics. Managing discussion processes means controlling the progresses in discussion rooms. The administrators can decide to go to other stages according to discussion situations. And the module would suggest the possible and appropriate next stages. Lastly, the administrators can modify and delete requirement articles if necessary.
2. Manage new terms: The administrators have to manage new terms which are established by stakeholders and did not appear in ontologies. The administrators can use this module to add new terms into ontologies as synonym, hyponym, hypernym, holonym, meronym, or metonym. In addition, the administrators also have to set time properties of the added terms in summaries. There are three values can be filled in a time property: planned in requirements, current situations, and history in the past. “Planned in requirements” indicates that the terms describe some blueprints or expectations for the future and possibly become the reality later. “Current situation” means the terms depicting living realities in the present. And “history in the past” terms just describe previous things which are possible preserved in memories.
3. Manage ontologies: The administrators can use this module to create, modify, and delete terms or relationships among ontology trees. For example, they can describe some role concept “uses” some function concept(s), and some component “is used by” some web page(s).

There are four kinds of data in the storage layer of Figure 3. These data have been mentioned in Figure 2. The relations between the business logic layer and the storage layer are described as follows.

1. Process: The administrators can use “manage discussions” module to get past and current progresses from process data, or schedule the progresses and store new stage instances as new process data.
2. Ontologies: The administrators can use “manage new terms” module to move terms from “new terms” data to “ontologies” data. Administrators can use “manage ontologies” module to add, modify, and delete concepts directly.
3. Summaries: The administrators can use “manage discussions” module to modify and delete “summaries” data.

4. New terms: The administrators can use "manage new terms" module to move new terms into ontologies or decide to drop garbage new terms.

4. Illustrative scenario

Online retailer company α has a shopping website ψ to service customers. Due to the volume of trade on website ψ grows up substantially; suppliers are willing to collaborate with company α more closely then before. On the ontology-based blogs (OBlog for short) of website ψ, supplier *S* creates a new topic and starts to describe an issue. Supplier *S* selects "Company α" from role ontologies to fill in the role field and selects "Website ψ" from function ontologies to fill in function filed in the requirement summary. Supplier *S* discovers that there is no appropriate term in activity ontologies and inputs a new term of "Collaborative Activities with Suppliers" in activity filed. Afterwards, supplier *S* posts the original requirement summary for creating a new topic:

company α can use website ψ to support collaborative activities with its suppliers

Then supplier *S* writes a blog article to explain the summary: "your website should allow your company to work closely with its suppliers".

After supplier S submits the requirement, OBlog's "Receive my article" module sends an e-mail to notify the stakeholders whose role is company α (e.g., a public speaker or chairman to represent the company) and its child roles as well as the stakeholders who subscribed articles mentioning company α and its child roles. On the OBLog, some stakeholders join this topic and give comments to agree that this requirement is interesting. But they also consider that proposing a concrete requirement is necessary.

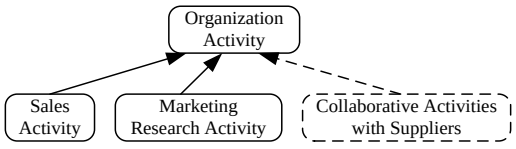

Figure 4. Adding "Collaborative Activities with Suppliers" into Activity Ontologies

Administrator *A* of the OBlog uses "manage new term" module to discover the new activity term "Collaborative Activities with Suppliers" and setup it as a child of "Organizational Activity" in the activity ontologies. Figure 4 shows a part of the activity ontologies. Administrator *A* also uses "manage discussions" to decide the progress move to "propose more detailed requirements" stage.

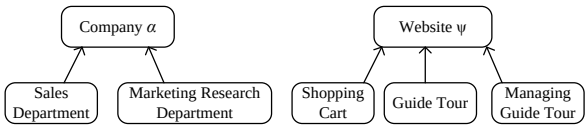

Figure 5. A Part of Role and Function Ontologies

"Join a topic to discuss" module shows Company α's child (e.g., sales department, and marketing department) and Website ψ's child (e.g., shopping cart, guide tour, and managing guide tour) as clues to remind stakeholders to propose a more concrete and detailed requirement. Figure 5 shows a part of role and function ontologies. Sales representative *G* proposes a concrete requirement. "Sales department" and "Managing Guide Tour" are chosen from role and function ontologies, and "Special Sales with Suppliers" in activity field are established by sales representative *G*. This requirement states: "Listing supplier's brand name in guide tour to hold special sales with suppliers may be a feasible way for realizing the original idea".

Administrator *A* moves the new "Special Sales with Suppliers"' term into activity ontologies, and set it as the common child of both "Collaborative Activities with Suppliers" term and "Sales Activity" term. The activity ontologies are updated as Figure 6.

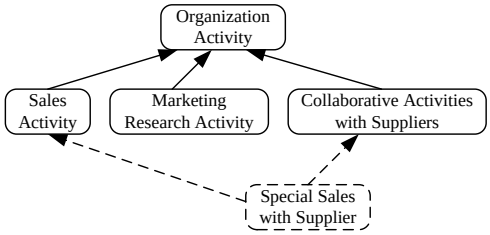

Figure 6. Adding "Special Sales with Supplier" into Activity Ontologies

Within several days, a lot of stakeholders receive notification about *G*'s requirement from the OBlog and give comments to support the requirement from sale representative *G*. Hence administrator *A* promises information systems department will further analysis the requirements of this topic.

According to the requirements form *S* and *G*, system analysis *H* creates a "Managing Brand Guide Tour" function as a child of "Managing Guide Tour" function. And system analysis H also creates "Brand List" data. The original requirement summary is revised as:

Sales Department (r) can use managing brand guide tour (f) to support special sales with supplier (a) in order to increase sales volume (g)

Administrator *A* updates ontologies as Figure 7.

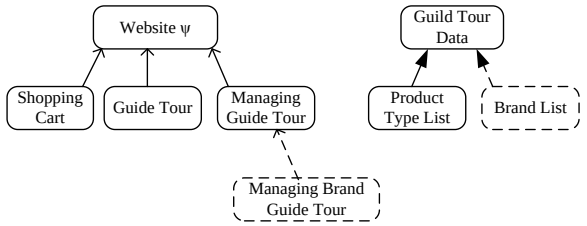

Figure 7. Adding "Managing Brand Guide Tour" and "Brand List" into Function and Data Ontologies

At this time, marketing researcher *I* considers that "Managing Brand Guide Tour" function also can support "Product Placement" activity and depict the requirement summary:

Market research department (r) can use managing brand guide tour (f) to support product placement (a) in order to build brand image (g)

However, the "join a topic to discuss" module detects *H*'s and *I*'s requirement may have a conflict because "Special Sales with Suppliers" and "Product Placement" have to achieve diverse goals: "Increasing Sales Volume" and "Building Brand Image". Figure 8 shows the goal ontologies. This module queries goal ontologies and finds that 'Company Growth' is their common parent. Therefore, the module reminds stakeholders to consider the "company growth" goal to resolve this conflict.

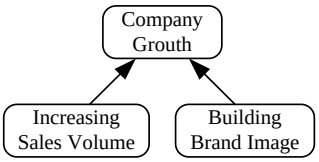

Figure 8. A part of Goal Ontologies

Subsequently, marketing researcher *I* decides to concede, and the sales department commit to help the marketing research department for building brand image in the future.

5. Conclusion

This research proposes a meta-model of functional requirements, and suggests six kinds of ontologies. The terms in these ontologies can be used for filling in the meta-model. Based on the meta-model and ontologies, this study proposes the architecture of ontology-based blogs and explains how this system assist stakeholders to post, receive, and manage requirements. In the meantime, we are undertaking to put this system in practice. A system prototype would be implemented first. And a real company has been contacted to establish their ontologies and agree to try this system as requirement elicitation tool in the future.

6. References

[1]J. Andrade, J. Ares, R. García, J. Pazos, S. Rodríguez, and A. Silva. A Methodological Framework for Viewpoint-Oriented Conceptual Modeling. *IEEE Transactions on Software Engineering*, 30(5):282-294, 2004.

[2] G. J. Browne, and M. B. Rogich. An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques. *Journal of Management Information Systems*, 17(4):223-249, 2001.

[3] S. Cayzer. Semantic Blogging and Decentralized Knowledge Management. *Communications of the ACM*, 47(12):47-52, 2004.

[4] G. B. Davis. Strategies for Information Requirements Determination. *IBM Systems Journal*, 21(1):4-30, 1982.

[5] M. Fowler and K. Scott. *UML Distilled: a brief guide to the standard object modeling language, Third Edition*. Addison-Wesley Professional, 2003.

[6] M. Heidegger. *Being and Time. J. Macquarrie & E. Robinson, Trans*. Harper & Row, 1962.

[7] H. Kaiya, and M. Saeki. Using Domain Ontologies as Domain Knowledge for Requirement Elicitation. In *14th IEEE International Requirements Engineering Conference (RE'06)*, September 2006.

[8] C. Lindahl, and E. Blount. Weblogs: Simplifying Web Publishing. *IEEE Computer*, 36 (11): 114-116, 2003.

[9] Maedche, B. Motik, L. Stojanovic, R. Studer, and R. Volz. Ontologies for Enterprise Knowledge Management. *IEEE Intelligent Systems*, 18(2):26-33, 2003.

[10] H. W. Nissen, M. A. Jeusfeld, M. Jarke, G. V. Zemanek, and H. Huber. Managing Multiple Requirements Perspectives with Meta-models. *IEEE Software*, 13(2):37-48, 1996.

[11] E. J. Rathswohl. Applying Don Ihde's Phenomenology of Instrumentation as a Framework for Designing Research in Information Science. In *Information Systems Research: Contemporary Approaches and Emergent Tradition*, H. -E. Nissen, H. K. Klein & R. Hirschheim (Editors), Elsevier Science, 1991.

[12] Rosenbloom. The Blogosphere. *Communications of the ACM*, 47(12):31-33, 2004.

[13] Jr. Young, and E. Terrence. Blogs: is the New Online Culture a Fad or the Future? *Knowledge Quest*, 31(5):50-51, 2003.