Some theoretical considerations on integration of ontologies

Hideaki Takeda and Toyoaki Nishida

Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara 630-01, Japan
phone: +81-7437-2-5261
fax: +81-7437-2-5269
takeda@is.aist-nara.ac.jp

keywords: foundation, integrity, knowledge engineering

Abstract

In this paper, we discuss integration of multiple ontologies in a formal way. First, we formalize ontology as combination of logical theories with modality. We introduce two types of integration. Combination aspect connects heterogeneous aspects in which aspect theories are simply merged. On the other hand, category aspect connects homogeneous aspects in which aspect theories are connected with possibility modality. In the above formalization, inter-aspect theories are defined syntactically but no semantical indication are given. Then we introduce abduction as a heuristics to find inter-aspect theories. Abduction can produce hypothesis which may integrate different aspects. To ensure integration of aspects in hypothesis, we use superposition of hypothesis which try to minimalize instances in hypothesis.

1 Introduction

Ontology is not only a key concept to realize intelligent systems systematically but also one to model our intelligent activities because it includes relations between the real world and our conceptual worlds. When ontology is regarded as an element of our intelligent activities, of interest is how ontology can be built. In this paper, we focus on integration of ontologies as a type of ontology building.

We first glance how multiple ontologies can exist and how they are expected to be integrated. Then we show our two methods for ontological integration. One is constructive approach in which ontologies are built as composite of ontological elements called aspects. The other is teleological approach in which integration is arisen as hypothesis.

2 Types of ontological integration

There are various definition of ontology depending on its purpose and application, e.g., [18][5][19]. We can pick up some common features on it, i.e., purpose-orientedness, partialness, agreement, correspondence to the target world, and consistency. Purpose-orientedness means that every ontology is built with some purpose. In general it seems that ontology is universal like discussion in philosophy, but ontology in computer science cannot be generated without purpose for use. Partialness means that ontology has its scope in the target world. Since ontology is theory of existence in philosophy, it should cover all of the world. In computer science, ontology is more limited so that it concerns only part of world, which is determined according to its purpose. Agreement means that ontology should be agreed by some community or society. As we mentioned above, ontology is not universal. but it should be shared. Correspondence to the target world means that ontology should not closed itself but should have connection to the target world. Consistency means that ontology itself should not contain inconsistency because ontology is a type of abstraction which requires operationalization.

In the above five features, the first three features imply that a single ontology is not sufficient so that multiple ontologies are introduced. Then we should consider how multiple ontologies can be cooperated to each other in order to state or solve problems.

There are practically various types of ontologies so that many possible combination of ontologies exist. One distinction of ontology is to divide it according to the subject of conceptualization, i.e., domain ontologies, process ontologies, and so on. If distinction among them is purely exclusive, integration of these ontologies is trivial because integration is just orthogonal.

The other distinction is general versus specific ontologies. Vertical integration or integration of general and specific ontologies is useful and practical. It is theoretically simple to integrate them because they can use some shared concepts as connection point between them. But as pointed out in Ref. [15], such ontologies often use different formalism which disturb vertical integration.

We focus on integration of ontologies of which subjects is the same and which are similar in generality.

We can furthermore find dimension to classify them, i.e., homogeneousness and heterogeneousness in domain. Homogeneous ontologies means that they represent the same domain as different styles. Heterogeneous ontologies are those with different domains. Integration of homogeneous ontologies is needed when different aspects to interpret the target world co-exist. For example, ontology for information gathering[9][8] can be different according to person or community, which can be homogeneous ontologies. Integration of heterogeneous ontologies is needed when more than two ontologies are required to solve some problem.

In this paper, we introduce *aspect* as component of ontology, i.e., ontology is built by combination of aspects. We define aspect and relations between aspects in a logical framework in Section 3. Here we can obtain what is relations between aspects, but do not know how relations are established. In Section 4, we show a way to establish their relations by abduction.

3 A Logical Formalization of Aspect

Since our basic policy is to define aspects constructively, we start from defining atomic aspect and then define more complicated aspects.

We assume a first-order language L_E, and predicate *aspect* of L_E. L is a first-order language which is the same to L_E except predicate *aspect* is removed.

3.1 Aspect Theory

First we define *atomic aspect*, aspect which does not depend on any other aspects.

Definition 1 *An atomic aspect A with a consistent theory $T(A)$ of a first language L and with a unique name $aspect(A)$ satisfies the following formula.*

$$aspect(A) \leftrightarrow T(A)$$

$aspect(A)$ is an identifier of aspect which has a similar effect to the second argument of predicate `ist` in Ref. [7], and a modal operator in Ref. [14].

Then, we introduce L_E^m and L^m as modal extension of L_E and L respectively. Here we assume domain of individuals are always the same regardless of possible worlds. In the following discussion, we assume this language L_E^m.

A combination aspect is simply defined as follows.

Definition 2 $T(A_{COM}(A_1, \ldots, A_n))$*, aspect theory of combination aspect for aspect* $A_1, \ldots, A_n$*, is a consistent theory defined as follows;*

$$T(A_{COM}(A_1, \ldots, A_n)) = aspect(A_1) \wedge \ldots \wedge aspect(A_n) \wedge I(A_1, \ldots, A_n)$$

where $I(A_1, \ldots, A_n)$ *is a set of formula of language* $L^m \cup \{aspect(A_1), \ldots, aspect(A_n)\}$*, which means inter-aspect theory among* $A_1, \ldots A_n$*.*

Apparently, it would cause unexpected results if some of aspect theories share predicates. We ideally assume that if the same predicates appear in some aspects, there should share some concepts in them[1]. In such cases, it should be represented by category aspects.

On the other hand, a category aspect is more complicated because it does not imply that both of aspect theories are *always* true. In order to represent a category aspect, we introduce modal operators $\Box$ (necessity) and $\Diamond$ (possibility) and assume S4 modal system. Then a category aspect for two aspects is define as follows.

[1] Of course, it is too strict in practise. In programming approach we allows the same predicates in different meanings.

Definition 3 $T(A_{CAT}(A_1, \ldots, A_n))$*, aspect theory of category aspect for aspect* $A_1, \ldots, A_n$*, is a consistent theory defined as follows;*

$$T(A_{CAT}(A_1, \ldots, A_n)) = \Diamond aspect(A_1) \wedge \ldots \wedge \Diamond aspect(A_n) \wedge I(A_1, \ldots, A_n)$$

where $I(A_1, \ldots, A_n)$ *is a set of formula of language* $L^m \cup \{aspect(A_1), \ldots, aspect(A_n)\}$*, which means inter-aspect theory among* $A_1, \ldots A_n$*.*

$I(A_1, \ldots, A_n)$ is again an inter-aspect theory among $A_1, \ldots, A_2$.

Since we can use combination and category aspects as component of aspects, we can define hierarchical aspects using combination and category aspects. In other words, An aspect A is represented $A = f(A_1, \ldots, A_n)$ where $A_1, \ldots, A_n$ are aspects and function f is composed by A_{COM} and A_{CAT}.

3.2 Inter-aspect Relations

Then we can define relations between aspect, **inclusion** and **strict inclusion**.

Definition 4 *An aspect A is* ***included*** *in aspect B if and only if* $aspect(B) \vdash \Diamond aspect(A)$*.*

Definition 5 *An aspect A is* ***strictly included*** *in aspect B if and only if* $aspect(B) \vdash aspect(A)$*.*

Note that there are two reasons for these relations, i.e., one is composition or category relations among aspects and the other is logical implication. Strict inclusion corresponds *weaker-than* relation in Ref. [14].

Similarly, relations between formula and aspect are defined.

Definition 6 *A formula f is* ***included*** *in aspect A if and only if* $aspect(A) \vdash \Diamond f$*.*

Definition 7 *A formula f is* ***strictly included*** *in aspect A if and only if* $aspect(A) \vdash f$*.*

These definitions mean that there are two types of interpretation of aspect theories. One is represented as *strict inclusion* which is traditional way of inter-theory relation. The other is *inclusion* which takes account of all alternatives of theories. By having two types of interpretation, we can deal with both strictly a single representation and variety of representations.

Theorem 1 *If aspect A is strictly included in aspect B, then A is included in aspect B.*

Another relation is **compatibility** which is criteria two aspects are related to each other[2].

Definition 8 *Aspect A and B is* ***compatible*** *if one of the following condition is satisfied;*

1. *A and B is the same aspect,*
2. *there exists aspect C which has both A and B as component,*
3. *there exist compatible aspect A' and B' are components of A and B respectively.*

Definition 9 *Formula f is* ***compatible*** *with aspect A if and only if there exists aspect B in which f is and B is compatible with A.*

Compatibility assure neither consistency nor translatability between aspect theories, but denotes existence of connection between aspects.

3.3 Inter-aspect Theory: Aspect-level Relations

Characteristics of category aspect varies according to its inter-aspect theory. One type of category aspect is *compact*.

Definition 10 *If $I(A_1, \ldots, A_n)$ of a category aspect satisfies the following formula, it is called* ***compact category***.

$$I(A_1, \ldots, A_n) \vdash \Box(aspect(A_1) \vee \ldots \vee aspect(A_n))$$

Intuitively, all of theories can be true and either of them should be true at any time in compact category aspects. It means that aspect $A_1, \ldots, A_n$ is sufficient to define the category.

Theorem 2 *If a compact category aspect A which has exactly two components A_1 and A_2, and A_1 is strictly included in A_2, then A_1 is strictly included in A.*

[2] Term *compatible* is borrowed from Ref.[6].

Actually relation between A and A_1 is stronger, i.e., $aspect(A) \vdash \Box aspect(A_1)$. We can say that any formula in A_1 is *rigid* in A by defining *rigidness* as follows;

Definition 11 *A formula f is rigid in aspect A if and only if $aspect(A) \vdash \Box f$.*

Theorem 3 *If a compact category aspect A which has exactly two components A_1 and A_2 and A_1 is strictly included in A_2, then any formula which satisfies A_1 is rigid in A.*

Rigidness in ontology is discussed in Ref.[6].

The other type of category aspect is *exclusive*.

Definition 12 *If $I(A_1, \ldots, A_n)$ of a category aspect satisfies the following formula, it is called* ***exclusive category****.*

$$I(A_1, \ldots, A_n) \vdash \bigwedge_{k=1,\ldots,n-1} \bigwedge_{l=k+1,\ldots,n} \neg\Diamond(aspect(A_k) \wedge aspect(A_l))$$

3.4 Inter-aspect Theory: Object-level Relations

We also describe relations between formulae in different component aspects by inter-aspect theories. We call such relations *object-level* relations, while we call relations described in the above subsection *aspect-level* relations. For example, p in aspect A should imply q in aspect B can be written as follows;

$$\Diamond(aspect(A) \wedge p) \rightarrow \Box(aspect(B) \rightarrow q)$$

More generally a rule "If f_1 in A_1 is true, then f_2 in A_2 should be true" is described as;

$$\Diamond(aspect(A_1) \wedge f_1) \rightarrow \Box(aspect(A_2) \rightarrow f_2)$$

If p in A and q in B should be equivalent to each other, then

$$(\Diamond(aspect(A) \wedge p) \rightarrow \Box(aspect(B) \rightarrow q)) \wedge (\Diamond(aspect(B) \wedge q) \rightarrow \Box(aspect(A) \rightarrow p))$$

Theorem 4 *If a proposition p is equivalent in all component aspects of a compact category aspect A, p is rigid in A.*

4 Integration of Aspects by Abduction

In the previous section, we formally defined aspects and their relations. But crucial point is how to establish inter-aspect theories. Some aspects have relations to each other by definition, i.e., aspects can be dependent on other aspects. But there are relations between them not generically but coincidentally. In this section, we adopt abduction as heuristics to find coincidental relations.

4.1 Generic Integration vs. Coincidental Integration

As we mentioned above, there are two types of relation between aspects, i.e., generic and coincidental relations. Generic relation means that an aspect is dependent on other aspects originally. In such case, their relations should be described with their definition so that aspect-level relation in the previous section is appropriate.

But some relations are not so general, i.e., more individual relations between aspects. We call it coincidental relation. For example, think about designing new objects, in particular in creative design Creative design does seldom happen within a single existing aspect, but with a new aspect which is new combination of existing aspects. New combination means that designers find new way to combine aspects, i.e., new relations among aspects. To design objects creatively is, thus, to find new relations among aspects that have not been recognized yet.

Suppose that a screw is introduced in a design from structural aspect and a stopper of linear movement from kinematics aspect. Then a designer decides to use the screw as the stopper. In this case an inter-aspect relation between the kinematics aspect and structural aspect is arisen. The designer is not sure that this relation, i.e., "screw as stopper" is really true before precise estimation of geometry, but he tries to keep it unless it turn out false. If screw as stopper is a general idea not but a special case in a special situation, it can possibly be added to ontological relations between kinematics and structural views. In this paper we formalize this process by abduction with multiple aspects.

4.2 Research on Abduction

C.S. Peirce introduced abduction as the third kind of reasoning in logic in addition to deduction and induction.

One of important characters of abduction he argued is that direction of inference in abduction is opposite to that in deduction. For example, he demonstrated abduction as follows [16];

> The surprising fact C is observed,
> But if A were true, C would be a matter of course;
> Hence, there is reason to suspect that A is true.

Many logical formalizations for abductive reasoning have been proposed recently, for example Levesque[10], Poole[17], Cox[1], and Finger[4], but their definitions for abduction are basically similar, i.e., abduction for an observation O with a theory T is to find a hypothesis A which consists of (ground instances of) possible hypotheses and satisfies both that $A \cup T \vdash O$ is hold and that $A \cup T$ is consistent. This definition is logically sound and suitable to represent the character of abduction mentioned above.

Unfortunately, this definition of abduction fails to capture another important character of abduction. Abduction is *ampliative* reasoning, while deduction is merely *explicative* reasoning. In ampliative inference the conclusion introduces new ideas into our store of knowledge, but it it does not follow from the premises with necessity [3]. In explicative inference the conclusion explicates what is stated in the premises and follows form the premises necessarily.

Hypotheses generated by the above definition are *definitely* all what can deduce the given observation with the given theory, and ampliativity is realized just by enumeration of multiple hypotheses.

This *clear and definite* abduction is unattractive in design because of complexity and quantity of object structures and knowledge. Since it translates ampliative ability of abduction into enumeration of multiple hypotheses, it would generate an enormous number of hypotheses. We need the other way to interpret ampliative ability of abduction.

The problem lies in the following two issues. One issue is that they put abduction into a traditional problem solving scheme. Abduction should include not only problem solving but also problem formation to some extent.

Although abduction may generate hypotheses by using reasoning like *reversed deduction*, it does not imply that the whole process of abduction is such reasoning. The other issue is lack of structures in hypotheses and the background theory. They assume simple and uniform structures that hide crucial problems in abduction like composition of hypotheses. For the former issue, we propose abduction as a process which includes finding theory used in deduction-style inference. For the latter, we use structuralized theory and hypotheses according to aspects.

In the following discussion, a problem given to abduction to solve is called an *observation*. It represents facts in the target world and it is what we should find explanation for. Knowledge which is used to find explanation is call a *background theory*. A *hypothesis* is an idea conjectured by abduction.

4.3 Definition of Abduction with Multiple Aspects

We first define abduction as the following way.

Definition 13 *An* ***explanation*** *of an observation O with a set of aspects T_B is $\langle A, T\rangle$, a tuple of an explanatory hypothesis A and an explanatory theory T which satisfy the following conditions;*

- $T \subseteq T_B$,
- $T \cup A$ *is consistent,*
- $T \not\models O$,
- $A \cup T \models O$, *and*
- *there are no $E \subset A \cup T$ that stratifies $E \models O$.*

We can say that a hypothesis A explains an observation O by an explanatory theory T. In this paper, we restrict both observations and hypotheses to ground formulae, i.e., no variables are appeared in them. Furthermore observations are given as a set of literals (atomic formulae or negation of atomic formulae).

This definition may seem identical to the definition in Section 4.2, but an explanation is not a hypothesis but combination of a hypothesis and an explanatory theory, and the whole background theory is not required to use in abduction. As we have mentioned, theory used in abduction is a theory which consists of part of aspect theories.

Then we should discuss how integrated and creative abduction is realized in this framework. The key idea is that minimalization of hypotheses and explanation with given constraints. The first approach is to minimalize hypotheses and the second is to minimalize explanations.

4.4 Superposition in Hypotheses

According to structure in explanatory theory, we can divide an explanatory hypothesis as follows;

$$A = A_{TH} \cup A_I$$

Here, A_{TH} is derivative hypothesis that can be derived from the background theory and the observation. A_I is connective hypothesis that integrates members of the derivative hypothesis. A derivative hypothesis A_{TH} alone can satisfy derivativeness of the observation O, i.e.,

$$A_{TH} \cup T \models O.$$

Since the hypothesis is generated from combination of different aspect theories, it may be merely a set of hypotheses each of which is generated from an aspect theory. To ensure integration of the hypothesis, we need the connective hypothesis which combines parts of the derivative hypothesis together. We realize this connective hypothesis as superposition of hypothesis.

4.4.1 Instantiation of Entities

An observation is a description about entities, and a hypothesis is another description about entities appeared in the observation. But in synthesis one should consider not only entities presented in the observation, but also entities needed to solve the given problem. We can call these entities instantiated entities.

Introduction of new entities should be careful because it changes the degree of integration of explanation. It is one of important criteria to create and evaluate hypotheses.

Suppose $\boldsymbol{a} = \langle a_1, ..., a_k \rangle$ a tuple of constants appeared in the observation, $\boldsymbol{i} = \langle i_1, ..., i_m \rangle$ a tuple of instantiated constants appeared in the hypothesis A, and $\boldsymbol{x} = \langle x_1, ..., x_n \rangle$ a tuple of variables.

Suppose that there are no constants in the explanatory theory.

We can get $A(\boldsymbol{x})$ by substituting each constant in A, $A(\boldsymbol{x})$ itself can explain the observation too, i.e.,

$$\forall \boldsymbol{x} A(\boldsymbol{x}) \cup T \models O.$$

Since we need hypotheses of ground formulae, we find a substitution θ to all variables in $A(\boldsymbol{x})$ so that $A(\boldsymbol{x})\theta = A$ [11].

We can also represent O as $O(\boldsymbol{y})\gamma_a$ where $\gamma_a = \{a_1/y_1, ..., a_k/y_k\}$ is a substitution. Then,

$$\forall \boldsymbol{x} A(\boldsymbol{x}) \cup T \models \boldsymbol{y}\gamma_a\, O(\boldsymbol{y}).$$

The fact that the observation is given as $O(\boldsymbol{y})\gamma_a$ not as $O(\boldsymbol{y})$ indicates that terms which satisfy every predicate in O should be restricted to constants used in the substitution γ_a. It means that $A(\boldsymbol{x})\theta \cup T \cup O$ should be minimal with respect to each predicate in O. Minimality with respect to a predicate is that the extension of the predicate (a set of tuples which satisfy the predicate) is minimal [2]. The extension of a predicate in O for $A(\boldsymbol{x})\theta \cup T \cup O$ should be the same to the extension for O. This restriction results as a substitution θ_a for $A(\boldsymbol{x})$, which abductive procedures with the resolution principle can find. But $A(\boldsymbol{x})\theta_a$ can have free variables still. Then these free variables in $A(\boldsymbol{x})\theta_a$ are assigned either to instantiated constants or to constants in O. Here θ_s stands for a substitution from variables to variables, θ_i for a substitution from variables to instantiated constants. Then $A = A(\boldsymbol{x})\theta_s\theta_i\theta_a$. θ_s represents identification between different terms, i.e., the way which entities in hypotheses should be identified.

For example, suppose

$$\begin{aligned}\{ \quad & is\_alive(x) \wedge has(x,y) \wedge wing(y) \wedge is\_feather(y) \rightarrow bird(x), \\ & has(x,y) \wedge wing(y) \wedge is\_big(y) \rightarrow fly(x)\}\end{aligned}$$

as T and $\{bird(a),\ fly(a)\}$ as O. If there are no ideas to identify entities, both

$$\begin{aligned}A_1 = \quad \{ \quad & is\_alive(a),\ has(a,b),\ wing(b),\ is\_feather(b),\ has(a,c), \\ & wing(c),\ is\_big(c)\}\end{aligned}$$

and

$$A_2 = \{is\_alive(a),\ has(a,b),\ wing(b),\ is\_feather(b),\ is\_big(b)\}$$

can be hypotheses. The former seems redundant, but both hypotheses are minimal because $A_1 \not\supset A_2$ and $A_1 \not\subset A_2$. The difference is the way how to introduce entities in hypotheses.

4.4.2 Minimality of Entities in Explanation

One of criteria to integrate hypotheses is minimality of entities. Domain circumscription [12] can be used to achieve minimality of entities in explanations. Domain circumscription finds models that have minimal domains to hold given formulae. In this case $A(\boldsymbol{x})\theta_a \wedge T \wedge O(\boldsymbol{a})$ is a formula to circumscribe. But using domain circumscription without any restrictions will make undesirable results. For the above example, we can get

$$\{is\_alive(a),\ has(a,a),\ wing(a),\ is\_feather(a),\ is\_big(a)\}$$

as a hypothesis with domain circumscription. This hypothesis seems unnatural, because we have knowledge about what kind of entities can be unified or not. In this case, entities which can satisfy $wing(x)$ and $bird(x)$ should be different, while entities which can satisfy $wing(x)$ can be unified to each other[3].

Superposition is identification between entities, but it is specified by two propositions which have entities to be identified.

Although it is impossible to describe all possible unifiable entity relations in knowledge[4], we can postulate at least consistency of aspect theories. Relations among predicates in an aspect are all what are written in the aspect theory. If two proposition have predicates in the same aspect, they are not allowed to identify unless these predicates are the same.

Suppose

$$T_1 = \{is\_alive(x) \wedge has(x,y) \wedge wing(y) \wedge is\_feather(y) \rightarrow bird(x)\}$$

$$T_2 = \{part(x,y) \wedge lift\_force\_device(y) \rightarrow fly(x)\}$$

[3]It is not a matter of course. If there are more than two entities which satisfy the same predicate, each of such predicates can be associated to different entities.

[4]It is the frame problem to enumerate all combinations among predicates [13].

$$T = T_1 \cup T_2$$

$$O = bird(a) \wedge fly(a)$$

where T_1 and T_2 are aspect theories. We can get a hypothesis

$$A = \{ \ is\_alive(a),\ has(a,b),\ wing(b),\ is\_feather(b),\ part(a,c),\ lift\_force\_device(c)\}.$$

If we assume superposition

$$\{has(x,y),\ part(x,y)\}$$

and

$$\{wing(x),\ lift\_force\_device(x)\},$$

then the hypothesis is

$$A' = \{ \ is\_alive(a),\ has(a,b),\ part(a,b),\ wing(b),\ is\_feather(b),\ lift\_force\_device(b)\}.$$

Notice such superposition is also a hypothesis, and validity of the superposition is examined by deduction and further abduction from the whole or part of the hypothesis A'. In particular, part of the hypothesis which includes identified entities is important in further abduction and deduction in order to realize how the superposition is feasible. In this example, it is $\{wing(b),\ is\_feather(b),\ lift\_force\_device(b)\}$.

Although superposition of hypothesis is merely individual relations between aspects, it can be possible inter-aspect relations. For example, a possible relation is that $wing(x) \wedge is\_feather(x)$ in Aspect T_1 is equivalent to $lift\_force\_device(x)$ in Aspect T_2.

5 Summary

In this paper, we discussed integration of multiple ontologies in a formal way. First, we formalized ontology as combination of logical theories with modality. We introduced two types of integration. Combination aspect connects heterogeneous aspects in which aspect theories are simply merged. On the other hand, category aspect connects homogeneous aspects in which aspect theories are connected with possibility modality. In the above formalization, inter-aspect theories are defined syntactically but no semantical indication are given. Then we introduced abduction as a heuristics to find inter-aspect theories. Abduction can produce hypothesis which may integrate different aspects. To ensure integration of aspects in hypothesis, we used superposition of hypothesis which try to minimalize instances in hypothesis.

Our attempt in this paper is still unmature both in conceptualization and formalization because it can cover only some viewpoints of integration of multiple ontologies. One future direction is to make the formalization concrete to be able to apply real applications. The other direction is to expand domain of the formalization, in particular, integration of different granularity of ontologies is interesting.

Acknowledgement

This research was partially funded by the "Modeling of Synthesis" project of the Japan Society of for the Promotion of Science's Research for the Future Program under the contract number, JSPS-RFTF 96P00701.

References

[1] P. Cox and T. Pietrzykowski. Causes for events: their computation and applications. In *Lecture Notes in Computer Science 230*, pages 608–621. Springer-Verlag, Berlin, 1986.

[2] M. Davis. Notes on the mathematics of non-monotonic reasoning. *Aritifical Intelligence*, 13, 1980.

[3] K. Fann. *Peirce's Theory of Abduction.* Martinus Nijhoff, The Hague, The Netherlands, 1970.

[4] J. Finger and M. Genesereth. RESIDUE: A deductive approach to design synthesis. Technical report stan-cs-85-1035, Stanford University, 1985.

[5] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University, August 1993.

[6] N. Guarion, M. Carrara, and P. Giaretta. Formalizing ontological commitments. In *Proceedings of AAAI-94*, pages 560–567, 1994.

[7] R. V. Guha. *Contexts: A Formalization and Some Applications.* PhD thesis, Department of Computer Science, Stadford University, Stanford, CA, 1991. (Available as Report No. STAN-CS-91-1399-Thesis).

[8] M. Iwazume, K. Shirakami, K. Hatadani, H. Takeda, and T. Nishida. IICA: An ontology-based internet navigation system. In *Working notes for AAAI96 Workshop on Internet-Based Information Systems*, pages 65–71, 1996.

[9] M. Iwazume, H. Takeda, and T. Nishida. Ontology-based information capturing from the internet. In *Proceegins of the fourth International Conference on the International Society of Knowledge Organization*, pages 261–272, 1996.

[10] H. Levesque. A knowledge-level account of abduction. In *Proceedings of IJCAI-89*, pages 1061–1067, Detroit, 1989.

[11] J. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, Berlin, 1984.

[12] J. McCarthy. Circumscription — a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[13] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4:463–502, 1969.

[14] P. P. Nayak. Representing multiple theories. In *Proceedings of AAAI-94*, pages 1154–1160, 1994.

[15] N. F. Noy and C. D. Hafner. The state of the art in ontology design. *AI Magazine*, 18(3):53–74, 1997.

[16] C. Peirce. *Collected Papers of Charles Sanders Peirce*, volume 5. Harvard University Press, Cambridge, MA, 1935.

[17] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[18] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.

[19] G. van Heijst, A. T. Schreiber, and B. J. Wielinga. Using explicit ontologies in kbs development. *International Journal of Human-Computer Studies*, 45:183–292, 1997.