15

Metaphor and the Semantic Web

Jonathan Biguenet[1], Bogdan D. Czejdo[2] and John Biguenet[3]
[1] Tulane University, [2] Fayetteville State University, [3] Loyola University
USA

1. Overview

Metaphor, a comparison of two entities, is a fundamental expression of human intelligence, offering a simple formulation of the assertion of a relationship. Though simple in form, the metaphor demands a complex interpretation of the relationship, revealing subtle correspondences between the two entities compared. Such an interpretation hinges on general background information and specific context. The metaphor's purpose is to shed new light on the two entities through their comparison.

While a simile, employing *like* or *as* in the comparison, makes clear that the two entities are only approximations of each other, a metaphor seems to suggest shared identity. However, to say a camera is an eye is not to say they are the same thing. So we cannot formally represent the relationship as an equality. Metaphor obviously requires a very complex cognitive operation to grasp its meaning. Similarly, an intelligent agent, such as a computer, faces a complicated problem in processing the data conveyed by a metaphor.

We will present a model for processing the relationship expressed in a metaphor. Examining the objects specified in the metaphor together with their properties exposes the layers of relationships implicit in the comparison. This examination is extended to other objects related by general background knowledge or specific context to the original paired entities. Through selection, rejection, and emphasis of the properties of all these objects, we establish a new complex relationship representing the metaphor. Ideally, this complex relationship contains all necessary information for the computer to understand the metaphor.

Human interpreters of a metaphor draw from their knowledge and the text itself; in our model, a computer would process a metaphor using knowledge databases expressed as Semantic Markups on the Web. If the Semantic Web is to support metaphor processing, these markups should define objects as well as their properties and relationships. Such a development would facilitate better use of the knowledge contained on the Web.

2. Introduction

Beginning with modern automatons and especially since Karel Čapek introduced the word *robot* in *R.U.R.*, his 1921 play, such mechanical creatures usually have been modeled on human beings, often in terms of form as well as function. "A robot is a human being" remains an influential, though debatable, metaphor.

Metaphor, a common device for the classification of objects and the assertion of newly recognized relationships, poses an array of challenges for natural language processing. A close analysis of the statement "a robot is a human being" quickly reveals the difficulties posed in the processing of the metaphor by a software agent.

The Semantic Web (Berners-Lee et al., 2001) offers the possibility of solutions to these challenges through its capacity for annotation. In this paper, we present the analysis of a common metaphor and its modeling, which can be a basis for annotating metaphorical statements.

Previous research (Booch et al., 1999; Czejdo et al., 2003; Czejdo et al., 2006; Delteil and Faron, 2002; Dori et al., 2003; Lehrnan, 1999; Sowa, 1984) demonstrates the need for appropriate models for the interpretation of complex statements such as metaphor. Here, we use the notation of Universal Modeling Language (UML) (Booch et al., 1999; Rumbaugh et al., 1999) for metaphor modeling. Such modeling has been employed to depict object-relationship systems (Rumbaugh, 1993). Concise graphical notation, availability of simple tools, and the possibility to use informal specifications if necessary make UML modeling very useful for defining ontologies. Therefore, UML diagrams can still play an important role even though new notations such as OWL (Smith et al., 2002) and new tools such as Protégé (Gennari et al., 2002) and SWOOP (Kalyanpur et al., 2005) allow for manipulation of ontologies.

In this paper, we first discuss the complexities of interpreting metaphor. We then examine the modeling of a specific metaphor as an example of this process. Continuing with an analysis of the inverse of the exemplary metaphor, we argue that metaphors are not commutative and thus illuminate one aspect of their complexity. The paper concludes that in order for the Semantic Web to interpret metaphors accurately, it must accommodate complex annotations.

3. Metaphor and Its Complexities

A metaphor is a direct comparison between two objects or states. A simile, another form of comparison, uses *like* or *as*. So "a robot is like a human being" is a simile and, perhaps, easier to interpret than the metaphoric form of the comparison. In the simile, the robot is merely an approximation of a human, sharing a limited number of attributes. In a metaphor, differences are either de-emphasized or masked by the superimposition of one of the two components of the metaphor upon the other. In a simile, such differences are acknowledged. As we have argued elsewhere (Czejdo et al., 2006; Biguenet et al., 2005), "*like* signals simply a congruence between the two entities compared; its use eliminates the possible confusion of the metaphor: that the two entities share identity. But the metaphor, while demanding a similar cognitive operation, declines to limit explicitly the relationship between the two entities compared to mere approximation. Instead, the metaphor hints at shared identity. In mathematical terms, we are tempted to express the metaphor as A equals B. The difficulty the computer faces in decoding such a statement's intended meaning is obvious." This potential for confusion becomes clear when the inverse of the metaphor is stated: "A robot is a human being" is not equivalent to "A human being is a robot."

The two elements of the metaphor's construction usually share only specific characteristics. In reducing the scope of the comparison to only those elements that are shared, the metaphor can efficiently express an essential relationship. But in doing so, it obscures difference. In fact, when applied to a human being, the adjective "robotic" is negative, suggesting the loss of intentionality and even freedom.

4. Extending UML Diagram to Capture Metaphoric Constructs

The UML diagrams, as previously mentioned, are well suited to the initial phase of knowledge extraction when all details are not yet clear. This feature is extremely important for systems of metaphor processing that require many phases of refinement.

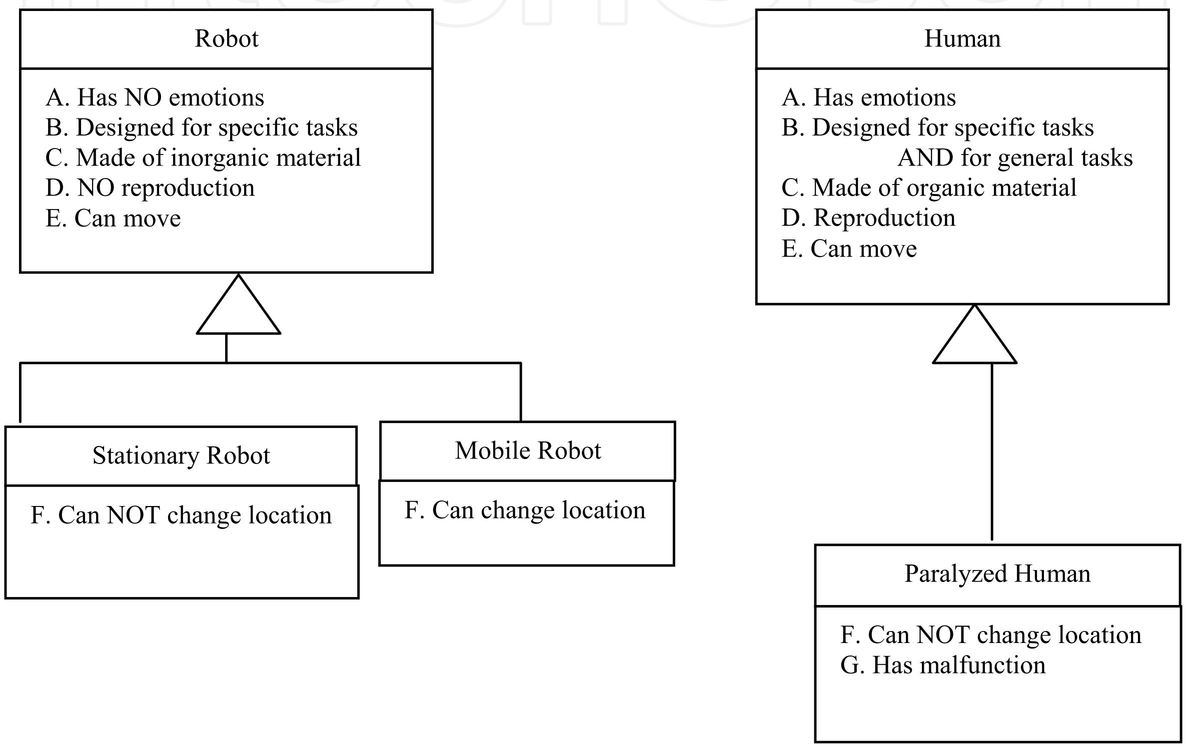

Fig. 1. Common knowledge about robots and humans represented by a UML class diagram

In the UML diagram used as an example in our paper describing a Robot and a Human as shown in Fig. 1, we define a Robot here (in a limited way) as an object that has no emotions, is designed for specific tasks, is made of inorganic material, cannot reproduce, can move, and can change location. We define Human here (in a limited way) as an object that has emotions, is designed for specific and general tasks, is made of organic material, can reproduce, can move, and can change location. Both Robot and Human are broken down into subclasses. For Robot, we have the two categories, Stationary Robots and Mobile Robots. A Stationary Robot we define as an object that cannot change its location. A Mobile Robot, on the other hand, can change its location. If a Mobile robot malfunctions, we call it a Malfunctioning Robot. A Malfunctioning Robot cannot change its location. A Paralyzed Human is one who cannot change location either.

The underlying UML diagram contains the knowledge about a specific subject or many subjects (e.g., about robots and humans). The database described by the UML diagram can therefore be used to answer both simple and complex queries in these subject areas. A simple query, similar to "Describe Robot," will include all properties and functions of the class (object). Among various complex queries, the comparison queries are some of the most useful for this project, allowing for an extended comparison of classes. Such a comparison of classes requires the comparison of their properties and functions as well as of all relationships.

"Compare a robot with a human" is an example of a comparison query. The comparison would result in the identification of all common properties and functions. The common subclasses and relationships need to be identified as well. Some additional processing is necessary since the common relationship type is a super-class. Because all attributes from a super-class can be inherited, they are also indicated as common. Additionally, we have a common relationship indicated by "has."

5. Metaphor Modeling by UML Diagram

A natural-language metaphor can contain quite a complex meaning not only stressing the similarity between two classes (or objects) but also simultaneously emphasizing or de-emphasizing some properties, functions, and relationships of the pair. Modeling the metaphor described in section 2, "The robot is a human," by a UML diagram demonstrates these features of metaphor.

As previously mentioned, the metaphor hints at shared identity and symmetrical structure, but scrutiny of the inverted form of the metaphor shows that such symmetry does not exist. Obviously, the simplistic initial reading of the metaphor must yield to a more precise definition of metaphorical structure. The metaphor, in terms of UML modeling, is rather a non-symmetrical special subclass/superclass relationship that allows for modifying properties, functions, and relationships of one concept by another. We will call this special subclass/superclass relationship "is a metaphor" and indicate it by arrows on the diagram. Returning to our example, we can create a new relationship called "is a metaphor" between Robot and Human. The direction of the arrow from Robot to Human in Fig. 2 indicates that "Robot is a Human."

Determining the definition of a metaphor in the UML diagram is a complex process. It requires not only insertion of a new type of relationship called "is a metaphor" but also identification of the list of modified properties, functions, and relationships with respect to the given metaphor, as well as determination of the type of modification of each listed property, function, and relationship. The modification can include emphasis, de-emphasis, or superimposition. Superimposition does not erase the real values but only temporarily masks them with new values. We refer to such properties, functions, and relationships as "superimposed." In Fig. 2, we will mask the property "designed for specific tasks" by the superimposed property "designed for specific tasks AND for general tasks." The masking may serve various goals. One of these goals is to achieve an "amplification" of a property. In our case, the new property "designed for specific tasks AND for general tasks" is superimposed, replacing temporarily the previous property "designed for specific tasks."

We must also identify any component metaphors created on the basis of another metaphor using the process of "propagation." In our example, we have a component metaphor between "malfunctioning mobile robot" and "paralyzed human" derived from our initial metaphor by its propagation to the appropriate subclasses. Fig. 2 also reflects that specification. Once the derived metaphor is stated, all of the same steps discussed above must be taken to define it.

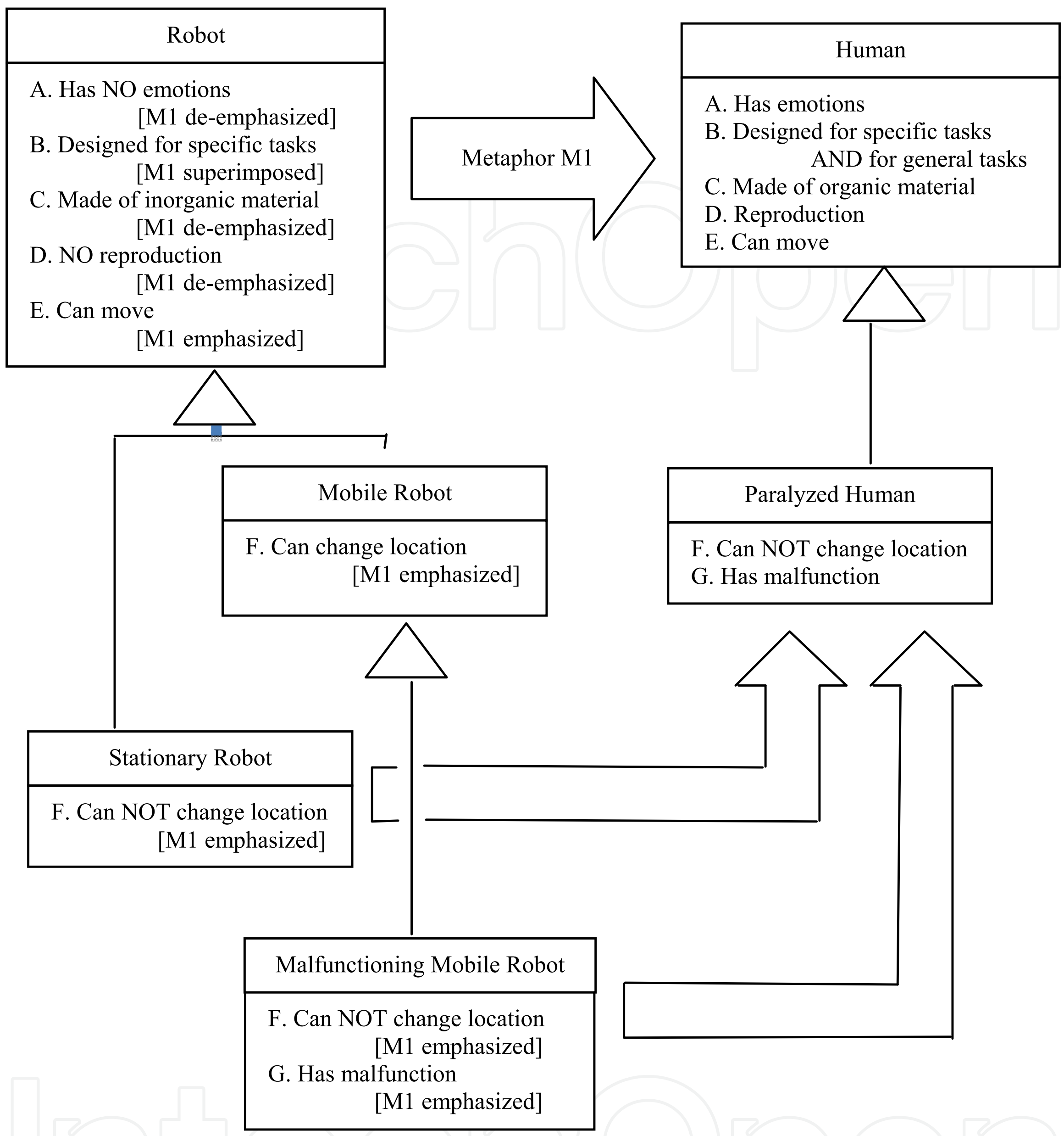

Fig. 2. Representation of a metaphor using UML diagrams

Let us expand our initial UML diagram to include components of robots and human beings, as shown in Fig. 3. A human has five senses. An eye is one of them. A robot has at least one sensor (e.g., camera). Included components of a robot are sensors, with a camera as a specific sensor type. Included components of a human are sense receptors, with an eye as a specific receptor. An eye captures an image and transmits an image to the human brain. A camera captures an image and records an image to digital memory.

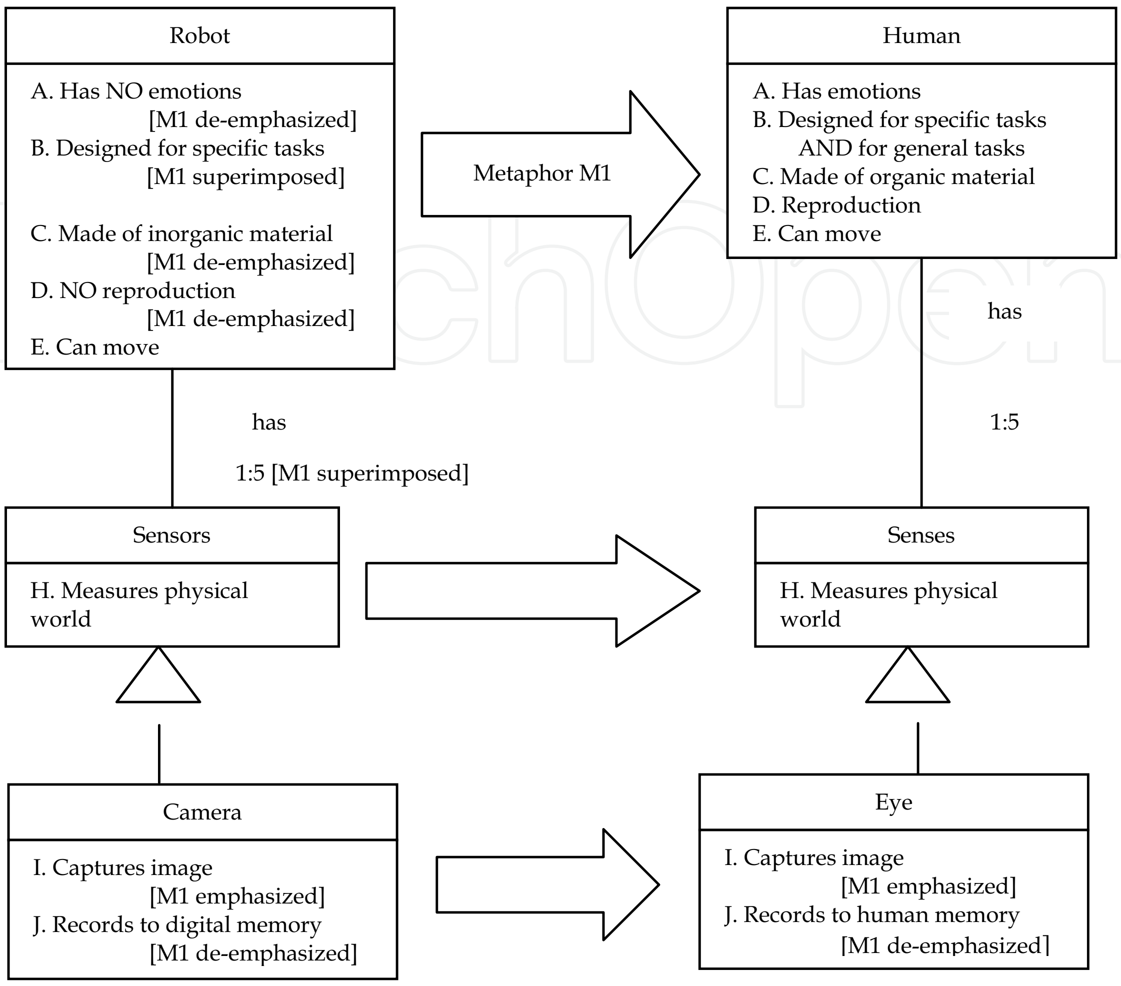

Fig. 3. Relationships between objects formed by extending our metaphor example

In Fig. 3, we have also another component metaphor between camera and eye derived from our initial metaphor by its propagation to the related ("has" relationship) classes. Fig. 3 reflects that specification.

6. Inverse Metaphor Modeling by UML Diagram

As we stated before, the metaphor is a non-symmetrical special subclass/superclass relationship that allows for modifying properties, functions, and relationships of one concept by another. The inverse metaphor might be similar to the initial (base) metaphor by stressing the compatibility of two classes (or objects), but it might be significantly different by emphasizing or de-emphasizing different properties, functions, and relationships. Let us model the inverse metaphor "The human is a robot" by a UML diagram. We can create a new relationship "is a metaphor" between Robot and Human in the direction from the Human to Robot as shown in Fig. 4.

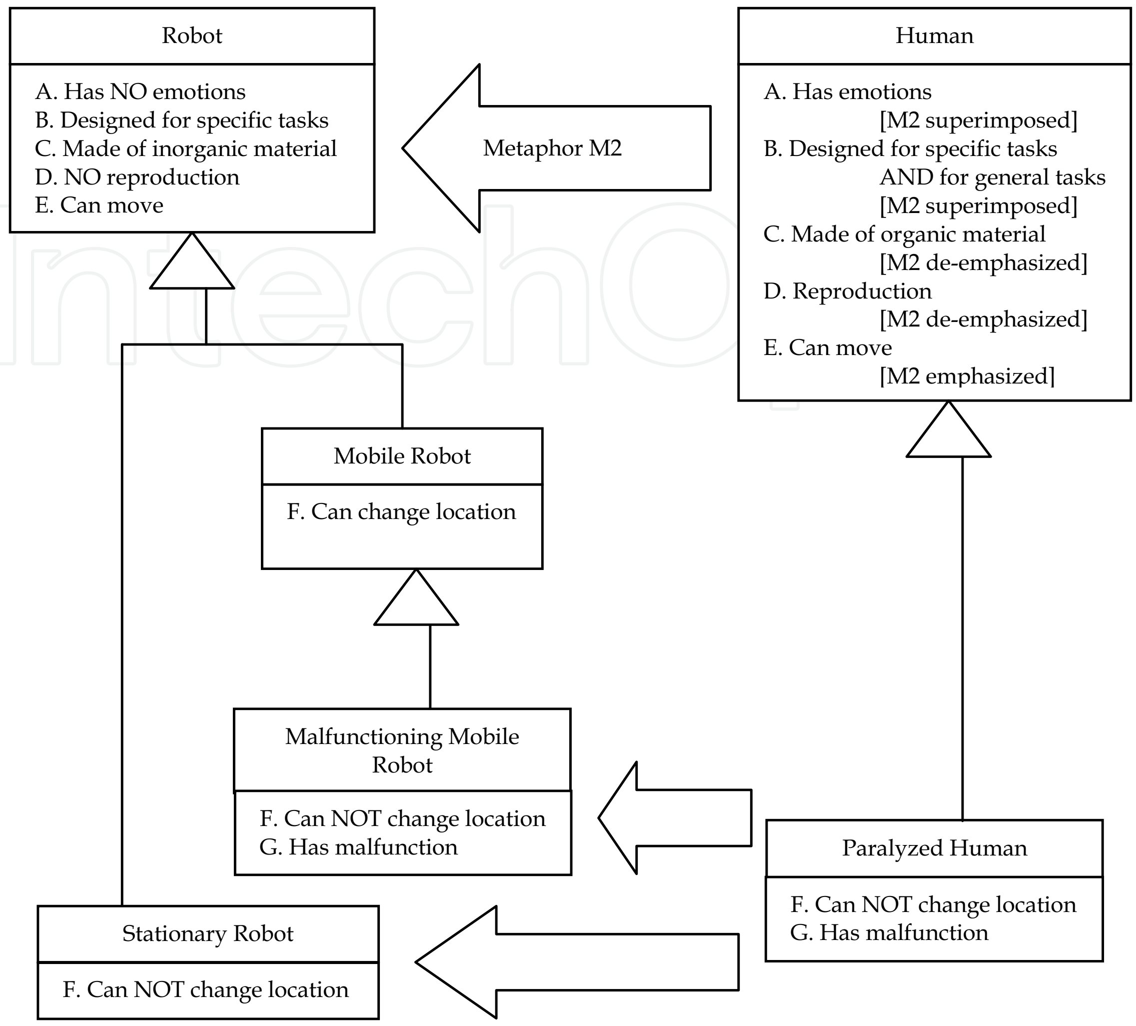

Fig. 4. Representation of a metaphor using UML diagrams

Let us look more closely at properties, functions, and relationships to be "superimposed." In our example in Fig. 4, we will identify the property "has emotions" to be "superimposed" by the new property "has no emotions" to indicate its new meaning for the metaphor. In this case, the masking served the goal of minimizing a property. If the property is positive, then decreasing its value may result in a negative metaphor. The propagation of this "negative" metaphor to the component classes ("has" relationship) could create a component metaphor "the eye is a camera," adding strength to the initial metaphor. It is interesting to note that propagation of the metaphor "A human is a robot" to the appropriate subclasses would make this metaphor negative, and perhaps even cruel, by suggesting that a paralyzed human is merely a malfunctioning mobile robot.

7. Summary

Metaphor obviously requires a very complex cognitive operation to grasp its meaning. Similarly, an intelligent agent, such as a computer, faces a complicated problem in processing the data conveyed by a metaphor. We have presented a model for processing the relationship expressed in a metaphor. Examining the objects specified in the metaphor together with their properties exposes the layers of relationships implicit in the comparison. Through selection, rejection, and emphasis of the properties of all these objects, we establish a new complex relationship representing the metaphor. Ideally, this complex relationship contains all necessary information for the computer to understand the metaphor. In our model, a computer would process a metaphor using knowledge databases expressed as Semantic Markups on the Web. If the Semantic Web is to support metaphor processing, these markups should define objects as well as their properties and relationships. Such a development would facilitate better use of the knowledge contained on the Web.

8. References

Berners-Lee T., Hendler, J., Lassila, 0.: The Semantic Web. Sci Am 284(5). (2001) 34-43

Biguenet, J., Czejdo, B, Biguenet, J.: "Creating Knowledge Databases from UML Diagrams", *Proceedings of Concurrent Engineering Research and Applications Conference*, Dallas (July 2005).

Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison Wesley, Reading, MA (1999)

Czejdo, B., Mappus, R., Messa, K.: The Impact of UML Diagrams on Knowledge Modeling, Discovery and Presentations. Journal of Information Technology Impact, Vol.3 (1). (2003) 25-38

Czejdo, B., Biguenet, J., Biguenet, J.: "Metaphor Modeling on Semantic Web", *Proceedings of Conceptual Modeling ER2006 Conference-Workshops*, Tucson (Oct 2006).

Delteil, A., Faron, C.: A Graph-Based Knowledge Representation Language. Proceedings of the 15th European Conference on Artificial Intelligence (ECAI), Lyon, France (2002)

Dori, D., Reinhartz-Berger, I., Sturm, A.: OPCAT—A Bimodal CASE Tool for Object-Process Based System Development. Proceedings of the IEEE/ACM 5th International Conference on Enterprise Information Systems (ICEIS 2003), Angers, France (2003) 286-291. www.ObjectProcess.org

Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubezy, H. Eriksson, N. F. Noy, S. W. Tu.: The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. (2002)

Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B., Hendler, J.: Swoop - a web ontology editing browser, Journal of Web Semantics 4 (1). (2005)

Lehrnan, F. (ed): Semantic Networks in Artificial Intelligence. Pergamon, Oxford, UK (1999)

Rumbaugh, J.: Objects in the Constitution: Enterprise Modeling. Journal of Object Oriented Programming. (1993)

Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. Addison-Wesley, Reading, MA (1999)

Smith, M.K., McGuinness, D., Volz, R., Welty, C.: Web Ontology Language (OWL) Guide Version 1.0. W3CWorking Draft (2002)

Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading, MA (1984)

Semantic Web
Edited by Gang Wu

ISBN 978-953-7619-54-1
Hard cover, 310 pages
Publisher InTech
Published online 01, January, 2010
Published in print edition January, 2010