Ontology-Based Interpretation of Arrow Symbols for Visual Communication

Yohei Kurata and Max J. Egenhofer

National Center for Geographic Information and Analysis
and
Department of Spatial Information Science and Engineering
University of Maine
Boardman Hall, Orono ME 04469-5711, USA
{yohei,max}@spatial.maine.edu

1. Introduction

Arrow symbols are paramount for visual communication, as they are used multi-purposely to represent directions, movements, interactions, relations, orders, and so forth. With arrow symbols, people typically illustrate dynamic processes even in a static diagram (Monmonier 1990, MacEachren 1995), as the existence of arrow symbols encourages people to interpret causal aspects in a diagram (Tversky, *et al*. 2000). Since arrow symbols are versatile, it is up to the diagram readers to interpret the semantic roles of arrow symbols. People usually interpret these roles almost intuitively, but computers need to learn formalisms for this interpretation (Kurata and Egenhofer 2005a). Due to the lack of such formalisms, today's pen-based systems require their users to specify explicitly the meaning of every arrow in a diagram (Forbus and Usher 2002) or restrict the use of arrows to a small set of meanings (Alvarado and Davis 2001, Landay and Myers 2001, Kurtoglu and Stahovich 2002).

This paper introduces a new approach that aims at interpreting an arrow-containing diagram based on its structure as well as semantics associated with elements in the diagram. Such an interpretation method is not only essential for natural human-computer interactions on pen-based systems (Figure 1), but also will be useful for evaluating the ambiguity of arrow symbols produced by people or computers (especially visualization systems) and minimizing the risk of misunderstanding.

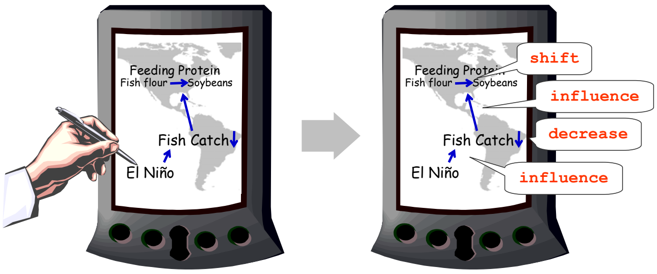


Fig. 1. A pen-based system, where users employ arrow symbols while sketching or annotating their ideas or plans on the screen as naturally as people often do on papers in face-to-face communication. When processing such diagrams, the system must interpret the semantic roles of the arrow symbols.

While the analysis of the syntactic structure of an arrow diagram (Section 2) has identified a small set of semantic roles that an arrow symbol may have in a diagram (Kurata and Egenhofer, 2005b), the structure alone leads at times to ambiguous or even invalid interpretations. To resolve these deficiencies we also consider the semantics of the relevant diagram components (Section 3). We focus here on correct interpretation of *actions*—and their distinctions from *associations* and *annotations*—by exploiting ontologies as general-purpose knowledge bases (Section 4). We conclude with a discussion of ongoing and future work (Section 5).

2. Syntax-Based Interpretations

Usually, the meaning of an arrow symbol is established when the arrow symbol *refers to* (i.e., originates from, points to, traverses, or goes along) other elements around the arrow symbol. The relative position of these elements and the arrow symbol are critical for such interpretations (Kurata and Egenhofer 2005a). The combination of arrow symbols and the elements to which the arrow symbols refer to is called *an arrow diagram* (Kurata and Egenhofer 2006). Then, the elements in an arrow diagram are called the *components* of the arrow diagram. The components are classified into the following five types (Figure 2):

- *objects* (O) take an action,
- *events* (E) occur in time and are characterized by a set of changes,
- *locations* (L) are positions in space,
- *moments* (M) are positions in time, and
- *notes* (N) are descriptions that modify other components and arrow symbols.

Since the components are located either in front of the arrow's head, behind the arrow's tail, or along/on the arrow's body, an arrow symbol identifies three different areas where the components can be located, which are called *component slots*. Kurata and Egenhofer (2005b) considered that the type of components in each slot of an arrow symbol form a syntactic pattern. This pattern is described as ($[O|E|L|M|N]^*$, $[O|E|L|M|N]^*$, $[O|E|L|M|N]^*$), where three elements in parentheses indicate the type of components in tail, body, and head slot, respectively, and $[x]^*$ means a sequence of any number of x or empty, and $x|y$ means x or y but not both (Figure 2).

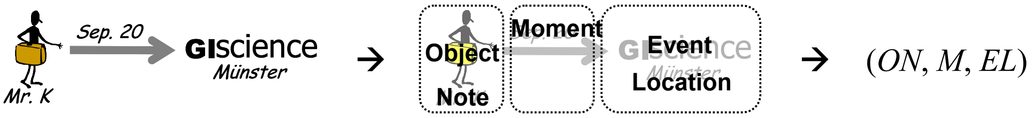


Fig. 2. An arrow diagram may have five types of components in three slots, which forms a syntactic pattern.

Studies of the syntactic patterns that underlie the combination of an arrow symbol and its related elements have identified a set of possible semantic roles (Kurata and Egenhofer, 2005b): *property*, *annotation*, *action*, and *association*. Arrow symbols for *properties* modify a component by itself (e.g., direction and vector). An arrow symbol for *annotation* connects two components such that one component is modified by another component (i.e., labeling). Arrow symbols for *actions* represent the motion of one component, which may trigger or be triggered by an interaction with another component. Finally, arrow symbols for *associations* illustrate the association of components by connecting them (e.g., order and asymmetric relation).

Kurata and Egenhofer (2005b) identified the syntactic requirements for assigning each semantic role to an arrow symbol, as well as the syntactic conditions for adding optional components to the arrow diagram. These two syntactic rules determine all syntactic patterns that may correspond to each semantic role. Consequently, possible semantic roles of an arrow symbol can be deduced, although not always uniquely, from the syntactic pattern of the arrow diagram. For instance, among 215 patterns of *simple arrow diagrams*, which contain at most one component in each slot, 81 patterns correspond to exactly one semantic role, and 52 patterns correspond to two or three roles (Figure 3). This means that in the 81 patterns of arrow diagrams the arrow's role is uniquely determined, whereas the 52 patterns of arrow diagrams need further information in order to narrow down the candidates.

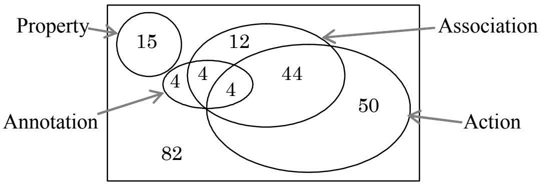


Fig. 3. The correspondence between 215 structural patterns of simple arrow diagrams and possible semantic role of arrow symbols in these diagrams (Kurata and Egenhofer 2005b).

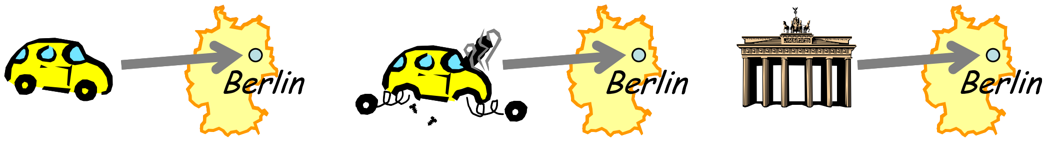
3. Beyond A Purely Syntax-Based Approach

In order to construct a meaningful arrow diagram, the components must have certain characteristics associated with them. For instance, Figures 4a and b illustrate the scenario "the car goes to Berlin" only if the car is movable. While such knowledge may depend on the instances depicted in the arrow diagram, in most cases it can be determined already at the class level. For example, cars are typically considered moveable, unless additional information (such as "a car without wheels") would negate a car's mobility. In general, to assign the previous four semantic roles to arrow symbols, the elements with the following characteristics are necessary:

- *Action* requires a component entity that moves, called the *mobile subject*.
- *Property* requires a component entity with a property that is related to orientation (and thereby represented by arrow's orientation), called the *orientation-related subject*.
- A*nnotation* requires an entity and a description that modifies the entity, called the *labeled-subject* and *label*, respectively.
- *Association* requires two entities that are linked by an associating rationale.

Normally, elements with these characteristics are represented in the arrow diagram in a component element, such that the drawer's intension is correctly communicated to the diagram reader (although these elements may be omitted if they are contextually evident). The use of arrow symbols, therefore, encodes such characteristics as *mobility* on some components in an arrow diagram. Not all components carry the necessary characteristics, however. For instance, a huge construction such as the *Brandenburg Gate*, or a geographic location like *Berlin*, are typically not considered mobile; therefore, Figure 4c does not capture an *action* (i.e., its arrow symbol does not represent a component's movement), since the arrow symbol for *action* has to impose the characteristic of *a mobile subject* on the *Brandenburg Gate* or on *Berlin*. In general, an arrow symbol does not qualify for a semantic role *r* if the use of an arrow symbol for *r* has to impose an impossible characteristic on one of the components.

Diagram drawers and readers usually share the knowledge about possible characteristics of components and, therefore, they can communicate arrow diagrams without misunderstanding. Lack of such knowledge results in the failure to remove impractical interpretation, such as "the Brandenburg Gate goes to Berlin" (Figure 4c), which has no problem from a syntactic viewpoint. This ambiguity indicates the need for common-sense knowledge about the world for more human-like diagram interpretations by computers. For this purpose, we employ knowledge extracted from *ontologies* (Guarino 1998), which are formal models of the people's conceptualizations about the world. A large variety of ontologies has been developed to model and formalize the knowledge shared in different domains. An ontology typically consists of vocabularies, properties and operations associated with each vocabulary, and relations between the vocabularies. Those relations usually include subsumption and metonymy relations (i.e., *is-a* and *part-of* relations), thereby establishing hierarchical structures between vocabularies.



(a) (b) (c)

Fig. 4. Arrow diagrams with the same syntactic pattern (O, –, L), illustrating (a) an action, (b) no action due to the immobility of the broken car, and (c) no action due to the (immobile) Brandenburg Gate.

4. Ontological Knowledge for Mobile Subjects

In this section, we show that the knowledge about possible characteristics of components is computationally derivable from an ontology. For generality, we use an upper ontology, which is typically used as a foundation of various domain ontologies. Here we demonstrate how to derive the knowledge about the possibility of entities to be a *mobile subject* from *WordNet* (Fellbaum 1998), which is a semantic lexicon for the English language and a well-known upper ontology.

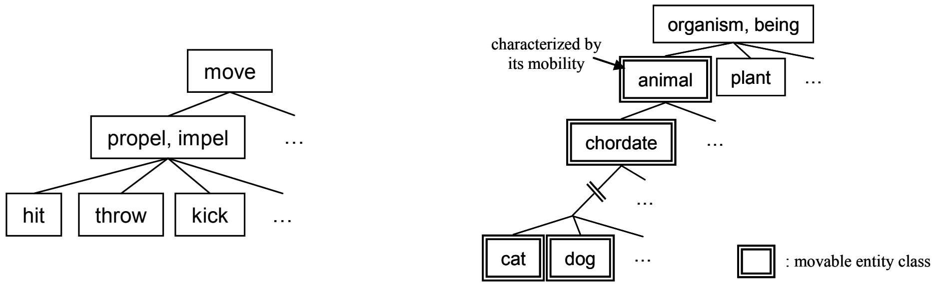
A component must be movable if it is considered a *mobile subject*. Mobility is often employed in the definition of an entity class as one of its essential characteristics. For instance, WordNet defines *animal* as "living organism characterized by voluntary movement." This definition clearly indicates that, *animals* are movable. The mobility of a class is also determined from the operations associated with the class. For instance, a *ball*, which is defined as "a round object that is hit or thrown or kicked in games," is associated with such operations as *hit*, *throw*, and *kick*. Since *hit*, *throw*, and *kick* are subclasses of the verb *move* (Figure 5a), the *ball* is considered movable.

Mobility is inherited from upper classes to lower classes. Consequently, any subclasses of *animal*, such as *dog* and *cat*, and any subclasses of *ball*, such as *soccer ball* and *tennis ball*, are also considered movable (Figure 5b). This is a great benefit of using an ontology for the judgment of the components' mobility.

If the superclass of class c is movable, but c itself is characterized by its immobility, c is immovable and immovability would be inherited to c's subclasses as well. For instance, a car without wheels is immovable (Figure 4b), although its superclass, a car, is movable. A truck without wheels, as a subclass of the car without wheels, is then immovable as well.

In general, class c qualifies as a *mobile subject* if

- c is characterized by its mobility in its definition,
- c has an operation *move* (in the sense of either a transitive or an intransitive verb) or a subclass of "*move*" (Figure 5a), or
- c's parent qualifies as *mobile subject* and c's definition does not characterized it a immobile (Figure 5b).



(a) (b)

Fig. 5. (a) Hierarchy of an operation "*move*" and its subclasses and (b) hierarchy of *animal* and its super/subclasses with inheritance of mobility.

A difficulty arises when determining the lack of mobility (i.e., immobility), since immobility is less recognized as an essential characteristic of an entity class than mobility. A realistic solution is to adopt the closed world assumption (Reiter 1987), that is, to assume that lack of knowledge about its mobility indicates its immobility. For instance, the *Brandenburg Gate* is considered immovable, because the *Brandenburg Gate* and its super classes (*memorial/monument*, *structure/construction*, *artifact*/*artifact*, and so forth) are not characterized by their mobility and have no operation related to *move*. Such inferences rely on the completeness of the ontology and may lead to unexpected consequences. For example, from WordNet one would misjudge a *cloud* in the sky to be immovable due to the lack of knowledge about its mobility. Since this problem arises from the incompleteness of WordNet, the use of another ontology may actually reveal the mobility of a cloud. Indeed, *Dictionary.com*, defines a *cloud* as "a large moving body of things in the air or on the ground," which clearly indicates the cloud's mobility. Such discrepancies among ontologies imply it may be required to employ and mine multiple ontologies for error-free judgments about the mobility of components.

Strictly speaking, mobility/immobility of an entity may be subject to direction and field. For example, an elevator can move only vertically and an aircraft carrier can move only on a (large) water surface. Consequently, Figure 6a cannot be interpreted as *an elevator goes to the right* and Figure 6b cannot be interpreted as *an aircraft carrier goes to an airport's landing field*. Such interpretations require the information about the entities' direction-dependent and field-dependent mobility, which is often derivable from the ontology as well. For instance, WordNet defines *elevator* as "a lifting device consisting of a platform or cage that is raised and lowered mechanically." The two operations associated with the elevator, *raise* and *lower*, indicate its movable direction. Similarly, the mobility of an aircraft carrier on the water surface is derivable from the definition of its superclass, a *vessel*—"a craft designed for water transportation."

(a) (b)

Fig. 6. Arrow diagrams with a restriction on their interpretations: (a) due to the elevator's direction-dependant mobility the diagram cannot be interpreted as *the elevator goes right* and (b) due to the carrier's field-dependent mobility the diagram cannot be interpreted as *the aircraft carrier moves to the landing field.*

Mobility of an entity may be determined by the mobility of its container and foundation, if they exist. For example, like the *Bandenburg Gate*, the *Liberty of Freedom* is typically immovable, although it once traveled across the Atlantic Ocean by ship (i.e., in a mobile container). Similarly, a *house* is typically immovable, but it may move by a landslide of the site (i.e., the foundation becomes a moveable surface). Containers and surfaces are image schemata (Johnson 1987), and their roles in modifying otherwise immobile items is a topic for future research.

5. Ongoing Work and Expected Conclusion

Arrow symbols are almost ubiquitous in visual communication. They are essential for representing, analyzing, and communicating spatio-dynamic information. To equip computers with human-like ability to understand arrow-containing diagrams will promote collaborations between people and computers on spatial tasks. This paper reported on our ongoing work about the development of a comprehensive computational method for interpreting semantic roles of arrow symbols in arrow-containing diagrams. We introduced a new approach for removing impractical interpretations with the aid of an ontology. The validity of interpretations relies on whether components in an arrow diagram can fulfill certain expected characteristics. Among such characteristics, this paper demonstrated how to determine the possibility of an entity to be a *mobile subject* from a general-purpose ontology. We are seeking the further use of ontologies for deriving knowledge about other possible characteristics of components and applying this knowledge to removing impractical interpretations as well as refining the interpretations.

References

Alvarado, C. and Davis, R. (2001) Resolving Ambiguities to Create a Natural Sketch Based Interface. in: Nebel, B. (ed.) *17th International Joint Conference on Artificial Intelligence (IJCAI-01)*. Seattle, WA, 1365-1374.

Fellbaum, C. (ed.) (1998) *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Forbus, K. and Usher, J. (2002) Sketching for Knowledge Capture: A Progress Report. in: Gil, Y. and Leake, D. (eds.): *7th International Conference on Intelligent User Interfaces*. San Francisco, CA, 71-77.

Guarino, N. (1998) Formal Ontology in Information Systems. in: Guarino, N. (ed.) *FOIS '98*, Trento, Italy, 3-15.

Johnson, M. (1987) The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason. University of Chicago Press: Chicago.

Kurata, Y. and Egenhofer, M. (2005a) Semantics of Simple Arrow Diagrams. in: Barkowsky, T., Freksa, C., Hegarty, M. and Lowe, R. (eds.) *AAAI Spring Symposium on Reasoning with Mental and External Diagram: Computational Modeling and Spatial Assistance*. Menlo Park, CA, AAAI Technical Report SS-05-06, 101-104.

Kurata, Y. and Egenhofer, M. (2005b) Structure and Semantics of Arrow Diagrams. in: Cohn, A. and Mark, D. (eds.) *COSIT '05*. Ellicottville, NY, Lecture Notes in Computer Science 3693, 232-250.

Kurata, Y. and Egenhofer, M. (2006) Topological Relations of Arrow Symbols in Complex Diagrams. in: Barker-Plummer, D., Cox, R. and Swoboda, N. (eds.) Diagrams'06, Menlo Park, CA.

Kurtoglu, T. and Stahovich, T. (2002) Interpreting Schematic Sketches Using Physical Reasoning. in: Davis, R. Landay, J. and Stahovich, T. (eds.) *AAAI Spring Symposium on Sketch Understanding*. Melon Park, CA, AAAI Technical Report SS-02-08, 78-85.

Landay, J. and Myers, B. (2001) Sketching Interfaces: Toward More Human Interface Design. *IEEE Computer*. 34(3):56-64.

MacEachren, A. (1995) *How Maps Work*. Guilford Press: New York, NY.

Monmonier, M. (1990) Strategies for the Visualization of Geographic Time-Series Data. *Cartographica*. 27(1):30-45.

Reiter, R. (1987) On Closed World Data Bases. in: Gallaire, H. and Minker, J. (eds.) *Logic and Data Bases*. Plenum Press, New York, NY, 55-76.

Tversky, B., Zacks, J., Lee, P. and Heiser, J. (2000) Lines, Blobs, Crosses and Arrows: Diagrammatic Communication with Schematic Figures. in: Anderson, M., Cheng, P. and Haarslev, V. (eds.) *Diagrams 2000*. Edinburgh, UK, Lecture Note in Artificial Intelligence, 1889, 221-230.