Division of Informatics, University of Edinburgh

Institute for Communicating and Collaborative Systems

Resolving References to Graphical Objects in Multimodal Queries by Constraint Satisfaction

by

Daqing He, Graeme Ritchie, John Lee

Informatics Research Report EDI-INF-RR-0028

Division of Informatics **October 2000**
http://www.informatics.ed.ac.uk/

Resolving References to Graphical Objects in Multimodal Queries by Constraint Satisfaction

Daqing He, Graeme Ritchie, John Lee

Informatics Research Report EDI-INF-RR-0028

DIVISION *of* INFORMATICS
Institute for Communicating and Collaborative Systems

October 2000

Published in the Proceedings of the Third International Conference on Intelligent Multimodal Interfaces, pages 8-15, Springer Publishers Germany 2000.

Abstract :

In natural language queries to an intelligent multimodal system, ambiguities related to referring expressions – source ambiguities – can occur between items in the visual display and objects in the domain being represented. A multimodal interface has to be able to resolve these ambiguities in order to provide satisfactory communication with a user. In this paper, we briefly introduce source ambiguities, and present the formalisation of a constraint satisfaction approach to interpreting singular referring expressions with source ambiguities. In our approach, source ambiguities are resolved simultaneously with other referent ambiguities, allowing flexible access to various sorts of knowledge.

Keywords : source ambiguities, referring expressions, natural language processing, intelligent multimodal interfaces

Resolving References to Graphical Objects in Multimodal Queries by Constraint Satisfaction

Daqing He, Graeme Ritchie, John Lee

Division of Informatics, University of Edinburgh
80 South Bridge, Edinburgh EH1 1HN Scotland
{Daqing.He, G.D.Ritchie, J.Lee}@ed.ac.uk

Abstract. In natural language queries to an intelligent multimodal system, ambiguities related to referring expressions – *source ambiguities* – can occur between items in the visual display and objects in the domain being represented. A multimodal interface has to be able to resolve these ambiguities in order to provide satisfactory communication with a user. In this paper, we briefly introduce source ambiguities, and present the formalisation of a constraint satisfaction approach to interpreting singular referring expressions with source ambiguities. In our approach, source ambiguities are resolved simultaneously with other referent ambiguities, allowing flexible access to various sorts of knowledge.

1 Source Ambiguities

With the widespread use of multimodal interface, many systems integrate natural language (NL) and graphical displays in their interactions. In some systems, graphics on the screen represent entities or attributes of entities in the application domain. For example, Fig. 1 shows a system called IMIG, from [7], where icons in the DISPLAY area represent individual cars, and characteristics of the icons convey attributes of the corresponding cars. A table of how attributes of the real cars are represented is displayed in the KEY area. Each representation is called a *mapping relation*. A user who is browsing through the cars with a view to buying may use the POTENTIAL BUY area to collect the icons of cars that s/he is interested in. During the interaction, the user can ask about the cars on the screen, or perform actions (e.g. *move*, *remove*,*add*) on the icons of those cars.

Interesting ambiguities can occur during interactions with a system like this because it has no total control of what the user can enter through NL modality. An attribute used in a referring expression can be an attribute either from the display on the screen, or from the entities in the application domain. For instance, as the worst scenario, the colour attribute represented by the word "`green`" in (1 a) potentially can denote the colour of an icon on the screen or the colour of a car in the domain, which are different. Another example of the ambiguity is that the referent of a phrase can be either the entity in the domain or its icon on the screen. For example, the two uses of the phrase "`the green car`" in (1 a) and (1 d) refer to different entities: the first refers to a *car* (represented by a green icon on the screen), whereas the second refers to that green *icon*.

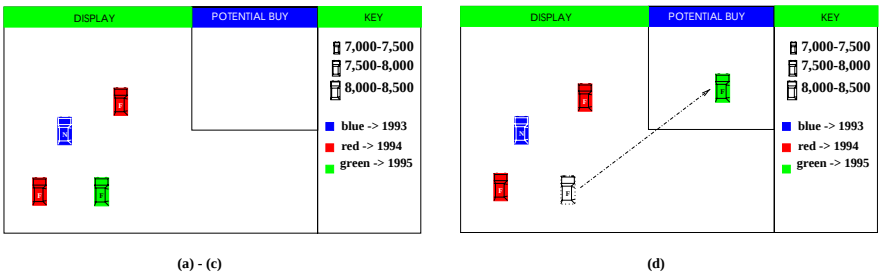

Fig. 1. Screen displays for (1)

(1) **User:** What is the insurance group of the green car? (a)
System: `It is group 5.` (b)
User: Move it to the potential buy area. (c)
System: `The green car has been moved.` (d)

In the IMIG system [6], this distinction is made by storing entities and their attributes from the domain (i.e. *domain entities/attributes*) in the *world model*, and those from the screen (i.e. *screen entities/attributes*) in the *display model*. The *source* of an entity/attribute indicates whether it belongs in the domain or on the screen. The ambiguities mentioned above are hence termed *source ambiguities*.

Source ambiguities affect the interpretation of referring expressions, and hence of the input query. Also, they cannot be resolved in a simple way. A satisfactory resolution of source ambiguities seems to involve at least knowledge about the screen, about the domain and about the dialogue context.

He et al. [7] postulated a *described entity set* and an *intended referent* for each referring expression. The described entity set is an abstract construct which allows a more systematic account of the relationship between the linguistic content of the expression (roughly, its "sense") and the (intended) referent. The described entity set of a singular referring expression is either a singleton set containing a entity (domain or screen), or a two-element set containing a domain entity and a screen entity which are related to each other by the mapping relation (see above). That is, the described entity set contains the objects that might be the referent of the phrase, based purely on the descriptive content of the phrase, but with the source unresolved. The intended referent is what is called the *referent* in the literature. More about source ambiguities can be found in He et al. [5–7].

There are several relevant linguistic regularities, within and between referring expressions, related to the sources of the words [7]. An example is the *screen head noun rule*, which states that *if the head noun of a phrase unambiguously names a screen category and it does not have a classifier modifier, then the described entity set of the phrase contains only screen entities and the intended referent is a screen entity too.*

2 The Resolving Process as a CSP

The restrictions used in source disambiguation can come from many places, such as the input query, the context of previous interactions and the content of the screen and the domain [7]. Formulating these restrictions as a CSP is natural and allows flexible processing. Referent resolution has already been proposed as a CSP [4, 10], and we propose that these two processes can be viewed as an integrated CSP. Source disambiguation is necessary for finding the intended referent of a phrase; conversely, the success or failure of referent resolution provides feedback on any potential solutions to source disambiguation.

3 Related Work

Among previous work related to source ambiguities, only He et al. [6, 7] provided a systematic discussion and a set of terminology for handling source ambiguities. The CSP approach here improves upon those ideas, by being more motivated, rigorous and general.

Ben Amara et al. [2] mentioned the issue of referring to objects by using their graphical features, and indicated that, to handle this type of references, the graphical attributes on the screen should, like those in the domain, be available to referent resolution. Binot et al. [3] mentioned the referent of a phrase being a graphical icon on the screen. However, as these authors acknowledged, they did not have a solution to source ambiguities.

Andre & Rist [1] and McKeown et al. [9] both allow natural language expressions to allude to graphic attributes, so that text can refer to parts of accompanying illustrations. However, they worked on multimodal *generation* and treated graphics on the screen merely as the descriptions of the represented domain entities. They did not have a resolution mechanism for source ambiguities.

4 The Formalisation of the CSP

The formalisation of source disambiguation and referent resolution into a CSP (called *CSP-souref*) consists of two stages: the identification of the variables and the extraction of the constraints on the variables.

4.1 Sources and Referents: the Variables

There are two kinds of variables in our CSP, and all of them have a finite range. A variable ranging across *entities* (potential intended referents, for example) has a range containing the entities in the context model and the display model, together with any entities that correspond, through mapping relations, to them. A variable ranging across *sources* has the range {`screen, domain`}. Each value in a variable's range is a potential solution (a *candidate*) for that variable, and the constraint resolver proceeds by refining that initial set.

For the simplicity of computation, two different variables are used in CSP-souref if it is not clear that two sources/entities are conceptually identical. For example, described entities can be derived from many parts of a referring expression (i.e. adjectives, nouns and prepositional phrases). Before the exact described entities are identified, it is difficult to tell which parts give rise to the same described entities. Therefore, in CSP-souref, different parts of a referring expression contribute different variables. Once solutions are found for these variables it will be explicit which variables actually represent the same described entity.

As a simplification in this preliminary exploration of source ambiguities, we are not considering plural referring expressions. Hence, the solution for an entity variable is a single entity – there are no sets of entities to be considered as referents. Also, because an entity has exactly one source, the solution for a source variable contains only one candidate.

Our resolving process has to be general enough to cover not only phrases that have referents, such as definite noun phrases, deictic phrases and pronouns, but also phrases that do not have referents, such as indefinite phrases. Entity variables corresponding to the former type of phrase require solutions to be computed, but variables for the latter type have no such requirement. Hence this distinction must be marked on the variables to allow the resolver to recognise when there is unfinished business.

4.2 Constraints and Preferences

Not all the attributes/relations related to the variables can be used as constraints in a CSP. Rich & Knight [12] pointed out that only those that are locally computable/inferrable and easily comprehensible are suitable constraints. We distinguish between obligatory *constraints*, which must be met by any candidate, and heuristic *preferences*, which can be ignored if necessary to find a solution.

All the restrictions that occur within CSP-souref can be stated naturally as unary or binary constraints/preferences, so we have adopted the simplifying assumption that there will be no more complex forms of restrictions.

Constraints on Sources are attributes/relations involving only source variables. They can be derived from the following origins of knowledge:

1. *Domain knowledge about the sources of some particular attributes.* For example, the word "`icon`" in the IMIG system is always related to an entity from the screen. A source variable related to this type of word would be given a (unary) constraint stating that "`the solution of variable` A `must be the source` a", written as $must\_be(A, a)$.

2. *The screen head noun rule* (see section 1). For example, suppose variable $S1$ represents the source of the described entity related to the word "`blue`" in the phrase "`the blue icon`" and variable $S2$ represents the source of the intended referent of the same phrase. The (unary) constraints generated from the rule can be written as $must\_be(S1, screen)$ and $must\_be(S2, screen)$.

3. *Semantic preconditions of relations.* For example, the possessive relation "`of`" in "`the top of the screen`" requires the variable $S4$, which represents

the source of a described entity of "`the top`", to have the same value as the variable $S5$, which represents the source of the intended referent of "`the screen`". The (binary) constraint can be written as $same\_source(S4, S5)$.

Preferences on Sources are from the *heuristic rules* mentioned in [5, 7]. For example, the words "`red`" and "`yellow`" in "`delete the red car at the right of the yellow one`" are preferred to have the same source because their semantic categories share a common direct super-category in the sort hierarchy, which satisfies the *same type* rule. This can be written as the binary preference $prefer\_same(S7, S8)$ where $S7$ and $S8$ represent the two source variables related to the two words respectively.

Constraints on Entities are another set of constraints involving at least one entity variable. They come from the following origins of knowledge, the first two of which were introduced by Mellish [10], while the last two are specific to CSP-souref.

1. *Local semantic information that derives from the components of a referring phrase.* In CSP-souref, the components are adjectives, nouns and prepositional phrases. For example, the noun "`car`" provides a constraint on its entity variable to be a car. Constraints from this origin are unary constraints.

2. *Global restrictions that derive from a sentence or the whole dialogue.* In CSP-souref, global restrictions mainly refer to the "preconditions" of operations that make the operations "meaningful". For example, the intended referent of the direct object of operation `move` should be movable. Constraints from this origin are unary or binary.

3. *Relations between an entity variable and a source variable representing the source of the entity.* The relations can be written as binary constraints that "`the two variables represent an entity and the source of that entity`".

4. *Restrictions between the described entities and the intended referent of the same phrase.* The restrictions state that the intended referent and the described entities of a referring expression are either the same entity or the entities linked with a mapping relation. They help CSP-souref to restrict the entity variables to have the same solution if the variables represent the same entity.

Preferences on Entities come from various information, including the heuristic about the relation between the salience of a dialogue entity and the likelihood of that entity being the referent. As stated in Walker [13], the more salient a dialogue entity is, the more likely it is to be the referent. Another origin of the preferences is the spatial distance between an entity and the position indicated by a pointing device. Work in the multimodality literature (such as Neal & Shapiro [11]) usually assumes that the nearer an entity to the pointed position, the more likely it is to be the entity that the user wants to point out. CSP-souref prefers the candidate nearest to the pointed position.

4.3 An Example

To help in understanding the formalisation, we use input command (2) as an example. Variables `DE**` and `Sde**` represent a described entity and its source,

Table 1. All the constraints raised from (2)

Con1	$must\_be(Sde12, domain)$.	Con2	$must\_be(Sde21, screen)$.
Con3	$must\_be(Sir2, screen)$.	Con4	$same\_source(Sir1, Sir2)$.
Con5	$has\_feature(DE11, blue)$.	Con6	$has\_feature(DE12, car)$.
Con7	$has\_feature(DE21, screen)$.	Con8	$has\_feature(IR1, removable)$.
Con9	$has\_feature(IR2, position)$.	Con10	$source\_entity(Sde11, DE11)$.
Con11	$source\_entity(Sde12, DE12)$.	Con12	$source\_entity(Sde21, DE21)$.
Con13	$source\_entity(Sir1, IR1)$.	Con14	$source\_entity(Sir2, IR2)$.
Con15	$same\_or\_corres(DE11, DE12)$.	Con16	$same\_or\_corres(DE11, IR1)$.
Con17	$same\_or\_corres(DE12, IR1)$.	Con18	$same\_or\_corres(DE21, IR2)$.

and `IR*` and `Sir*` represent an intended referent and its source. `DE11, Sde11, DE12, Sde12, IR1` and `Sir1` are identified from the phrase "`the blue car`", and variables `DE21, Sde21, IR2` and `Sir2` are from the phrase "`the screen`". `Sde11, Sde12, Sde21, Sir1` and `Sir2` range over {`screen, domain`}, and `DE11, DE12, DE21, IR1` and `IR2` range over a set containing two classes of entities (see section 4.1 above). The first are all the entities from the context and display models, and the second class are the entities related to entities in the first class through mapping relations.

(2) "`remove the blue car from the screen`"

Table 1 lists all the constraints extracted from (2). `Con1` to `Con4` are constraints on sources where `Con1` and `Con2` are from the origin 1, `Con3` is from the screen head noun rule and `Con4` is from the semantic preconditions. The remaining constraints are constraints on entities. `Con5` to `Con7` come from local semantic information, `Con8` and `Con9` are from the global restrictions, `Con10` to `Con14` are from the origin 3 and `Con15` to `Con18` are from the origin 4.

5 Resolving CSP-souref

A binary CSP can be viewed as a constraint network, whose nodes and arcs are the variables and the constraints, respectively. We have adopted Mackworth's network consistency algorithm (AC-3) [8], which achieves node and arc consistency. This is because the algorithm seems to be the most cost-effective way of resolving CSP-souref.

We use the example (2) to explain the process of constraint satisfaction for CSP-souref. Assume the relevant knowledge bases contain:

```
THE WORLD MODEL
   car(car1),car(car2),blue(car1),have_source(car1,domain),red(car2),
   car(car3),white(car3),have_source(car2,domain),have_source(car3,domain)
THE DISPLAY MODEL
   icon(icon1), red(icon1),have_source(icon1,screen),removable(icon1),
```

Table 2. The candidate sets of variables in Table 1 during constraint satisfaction. {*} represents {car1, car2, icon3, icon1, icon2, screen1}

variable	initial candidate set	after NC	after AC-3
Sde11	{screen, domain}	{screen, domain}	{domain}
Sde12	{screen, domain}	{domain}	{domain}
Sde21	{screen, domain}	{screen}	{screen}
Sir1	{screen, domain}	{screen, domain}	{screen}
Sir2	{screen, domain}	{screen}	{screen}
DE11	{*}	{car1, icon3}	{car1}
DE12	{*}	{car1, car2}	{car1}
DE21	{*}	{screen1}	{screen1}
IR1	{*}	{icon3, icon1, icon2}	{icon1}
IR2	{*}	{screen1}	{screen1}

```
   icon(icon2),red(icon2),have_source(icon2,screen),removable(icon2),
   icon(icon3),blue(icon3),position(screen1),have_source(icon3,screen),
   removable(icon3), screen(screen1),have_source(screen1,screen).
THE MAPPING MODEL
   corres(car1, icon1),corres(car2, icon2),corres(car3, icon3)
THE CONTEXT MODEL:
   car1, car2, icon3
```

Table 2 shows the results after achieving node and arc consistency respectively. In this example, each variable has only one candidate in its candidate set when network consistency is achieved. This candidate is the solution.

However, preferences sometimes have to be used to find the solution even after network consistency has been achieved. A preference selects a candidate for one of the remaining variables that have not found their solutions. With this piece of new information, network consistency can be achieved again in a new state, which might reach the solution. However, this could also sometimes lead to an inconsistency, where backtracking is necessary to get rid of the inconsistency. In IMIG, backtracking usually starts with removing the preference just applied.

We did an evaluation involving human subjects for the usefulness of our approach, which includes examining CSP-souref using the heuristic rules, source and context information. The statistical outcome shows that the functions provide significant help in making the test dialogues free from misunderstanding [5].

6 Conclusion and Future Work

In this paper, we have presented a framework for dealing with source ambiguities. By using a constraint satisfaction method, we integrated source disambiguation with referent resolution, and provided a unified mechanism to handle various restrictions on source ambiguities or referent ambiguities. In addition, the sequence of applying the restrictions is much more flexible in our approach.

Future work lies in the following directions: 1) mapping relations are accessible during the resolution, but they are assumed not to be mentioned by the user in the dialogues. For example, the sentence "`which car is represented by a blue icon?`" is not considered, although it could be used in an interaction. 2) the heuristics used in CSP-souref are mainly based on our intuition and experiments on a very small number of test dialogues. This restricts the applicability of these heuristics and imposes difficulties in further developing the resolution model. It would be beneficial to have a multimodal dialogue corpus. Even just a small one that contains only the dialogues related to our research would facilitate further exploration on source ambiguities.

Acknowledgements: The first author (Daqing He) was supported by a Colin & Ethel Gordon Studentship from the University of Edinburgh.

References

1. André, E., Rist, T.: Referring to World Objects with Text and Pictures. In *Proceedings of COLING'94* Kyoto Japan (1994) 530–534.
2. Ben Amara, H., Peroche, B., Chappel, H., Wilson, M.: Graphical Interaction in a Multimodal Interface. In *Proceedings of Esprit Conferences*, Kluwer Academic Publisher, Dordrecht, Netherlands (1991) 303–321.
3. Binot, J., Debille, L., Sedlock, D., Vandecapelle, B., Chappel, H., Wilson, M.: Multimodal integration in MMI^2: Anaphora Resolution and Mode Selection. In Luczak, H., Cakir, A., Cakir, G. (eds.): *Work With Display Units–WWDU'92* Berlin, Germany (1992).
4. Haddock, N.: *Incremental Semantics and Interactive Syntactic Processing.* PhD thesis, University of Edinburgh (1988).
5. He, D.: References to Graphical Objects in Interactive Multimodal Queries. PhD thesis, University of Edinburgh (2000).
6. He, D., Ritchie, G., Lee, J.: Referring to Displays in Multimodal Interfaces. In *Referring Phenomena in a Multimedia Context and Their Computational Treatment, A workshop of ACL/EACL'97* Madrid Spain (1997) 79–82.
7. He, D., Ritchie, G., Lee, J.: Disambiguation between Visual Display and Represented Domain in Multimodal Interfaces. In *Combining AI and Graphics for the Interface of the Future, A workshop of ECAI'98* Brighton UK (1998) 17–28.
8. Mackworth, A.: Consistency in Networks of Relations. *Artificial Intelligence* **8** (1977) 99–118.
9. McKeown, K., Feiner, S., Robin, J., Seligmann, D., Tanenblatt, M.: Generating Cross-References for Multimedia Explanation. In *Proceedings of AAAI'92* San Jose USA (1992) 9–16.
10. Mellish, C.: *Computer Interpretation of Natural Language Descriptions.* Ellis Horwood series in Artificial Intelligence. Ellis Horwood, 1985.
11. Neal, J., Shapiro, S.: Intelligent Multimedia Interface Technology. In Sullivan, J., Tyler, S. (eds.) *Intelligent User Interfaces* ACM Press New York (1991) 11–44.
12. Rich, E., Knight, K. *Artificial Intelligence.* McGraw-Hill, New York, 2 Ed (1991).
13. Walker, M.: Centering, Anaphora Resolution, and Discourse Structure. In Walker, M., Joshi, A., Prince, E. (eds.): *Centering Theory in Discourse.* Oxford University Press (1997).