Knowledge-Based Systems 21 (2008) 617–628

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

References to graphical objects in interactive multimodal queries

Daqing He [a,*], Graeme Ritchie [b], John Lee [c]

[a] School of Information Sciences, University of Pittsburgh, 135 North Bellefield Avenue, Pittsburgh, PA 15260, USA
[b] Department of Computing Science, University of Aberdeen, Aberdeen AB24 3UE, Scotland, UK
[c] School of Informatics, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, Scotland, UK

ARTICLE INFO

Article history:
Received 22 May 2007
Accepted 24 March 2008
Available online 31 March 2008

Keywords:
Source ambiguities
Intelligence multimodal interfaces
Constraint satisfaction problems
Reference resolution

ABSTRACT

Traditionally, interactive natural language systems assume a semantic model in which the entities referred to are in some abstract representation of a real or imagined world. In a system where graphical objects such as diagrams may be on the screen during the language interaction, there is possibility that the user may want to allude to these visual entities. Since graphical objects have their own properties (colour, shape, position on the screen, etc.) but may also represent items in a knowledge base which have other associated properties (price, geographical location, technical specifications, etc.), some systematic way is needed to enable such objects to be referred to in terms of either their screen properties or their associated attributes from the domain under discussion. In this paper, we present a formalisation for these arrangements, and show how our logical definitions can be used to generate constraints suitable for reference resolution within a natural language interpreter.

1. Introduction

For many years, the typical natural language (NL) query interface would assume a simple environment in which a symbolic representation of information about some domain (i.e., a knowledge base) was the sole source of responses to questions, and was the only place in which referents might be sought for phrases (for overviews of conventional NL query systems, see [41,12,5,4]). In more recent years, attention has been paid to more complex systems, in which graphical displays are integrated into the system handling the queries (see Section 3 below). Such an arrangement raises interesting questions about the semantic model underlying the interpretation of NL queries, since words and phrases may refer to knowledge base entities directly (as in traditional systems) or may denote graphical objects visually accessible on the screen. A query may even contain a mixture of these two broad classes of reference. To complicate matters further, the screen entities (icons, etc.) may be in a denotational relationship to the symbolic entries in the database, so that a user wishing to allude to a database object may have two routes for achieving this: a phrase which directly mentions the properties of the object as recorded in the database, or a phrase which uses the visual properties of the corresponding icon in order to single out (indirectly) the database object.

We focus on this basic situation: a database (or knowledge base) is queried in English where there may be images (icons) on the screen corresponding to database objects. We consider how simple referring noun phrases could be interpreted when there may be ambiguity about whether screen or database objects (or properties) are being referred to, or when a single phrase may contain a mixture of screen and database information.

In the remainder of the paper, we will first explain this scenario in more detail and propose our approach for handling it (Section 2). We then outline briefly some of the related work in this area (Section 3). After that, we will present our language-interpretation framework (Section 4), discuss our formalisation in general set-theoretic terms (Section 5) and in terms of constraint-resolution (Section 7), including a simple worked example (Section 9). In Section 10, we consider limitations and possible developments of the work.

2. Motivation

2.1. Example scenarios

Graphical displays inherently have spatial relations, but the applications that they represent may or may not. For simplicity, we shall start by considering applications in which there is no real spatial information in the subject matter (as opposed to the screen display, which cannot avoid being spatial). We shall return to the issue of domains with inherent spatial content later. The following example is merely illustrative, but is closely based on the simple scenarios which we have tested by implementation [22]. In considering this example, it should be borne in mind that we are not

* Corresponding author. Tel.: +1 412 624 2477; fax: +1 412 648 7001.
E-mail addresses: daqing@sis.pitt.edu (D. He), gritchie@csd.abdn.ac.uk (G. Ritchie), J.Lee@ed.ac.uk (J. Lee).

0950-7051/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2008.03.023

proposing it as a model of excellence in HCI, but rather as an artificial context for provoking the phenomena we are interested in while remaining simple enough for clarity. We imagine these kinds of phenomena as arising more plausibly in a system that would more closely approximate human–human dialogue, including the use of speech input. This would include cases where, for example, the user can introduce graphical items and state new interpretations for them, as perhaps a sketch system embedded in a natural language dialogue. A system supporting such discourse would, however, raise many issues that would obscure our focus in this paper. In a simple case such as we discuss here, one might well in practice use direct manipulation and forego natural language altogether; but our point is that even in such a simple case it is possible to exhibit the ambiguities we are interested in, notwithstanding that it would be straightforward to avoid them.

Consider a system which allows the user to browse through a database of used cars which are for sale, and which displays the information about the available vehicles in a simple graphical form (Fig. 1). Icons in the DISPLAY area represent individual cars, and characteristics of these icons convey attributes of the corresponding cars, as summarised in the table displayed in the KEY area. A user who is browsing through the cars 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 screen actions (e.g., *move*, *remove*, *add*) on the icons of those cars.

We assume that the system maintains some form of table, referred to here as the *mapping relation*, which shows which database entity corresponds to each of the icons on the screen, and we assume that this is a one-to-one mapping from screen into domain (i.e., every screen item has a unique corresponding domain entity, but some domain entities may have no associated screen item).

The system does not impose any explicit limits on how these commands or queries may be phrased, allowing the user some freedom in the choice of NL expressions.

Let us consider a case where the user asks about one particular car, as in Example (1), with graphical displays as in Fig. 1, where the bottom-right icon is displayed in green, and where the database contains no information about the colours of actual cars.

(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)

The adjective "*green*" in the phrase "*the green car*" indicates the colour of the *icon*, not the colour of the *car* being represented. However, the head noun "*car*" refers to possible entities in the database (or, depending on one's point of view, in the domain of discussion). So too does the entire NP "*the green car*", since cars have insurance groups, but icons do not. Hence when computationally calculating the potential candidate entities for the reference of "*the green car*", we cannot simply consider only the entities of type "*car*" which have the property "*green*".

Suppose the user now makes the request indicated as (c), and gets the response from the system shown as (d). Here, the object being moved is clearly an *icon* – no cars are being driven through the streets. Nevertheless, the user uses "*it*" to refer to the icon, presumably referring anaphorically to the earlier phrase "*the green car*" which, as argued already, denotes not an icon but a car. Moreover, the system's response uses the phrase "*the green car*" to refer to the moved object, i.e., an icon.

Underlying the above comments is one of our central assumptions, namely that it makes sense to talk about an *intended referent* for a noun phrase in a particular context of use. This is the object which the phrase can be seen to denote when all possible linguistic and contextual information (including the requirements of actions) has been taken into account. (The term has been used in an analogous manner with respect to the generation of referring expressions [24]). In discussing our illustrative examples, we shall rely on intuition about the meanings of the sample sentences to decide the intended referents, always adopting the assumption that referents must meet all internal semantic constraints of the dialogues. One way of looking at our proposed framework is to say that the information which contributes to singling out an intended referent may not be straightforward, particularly if a phrase is viewed in isolation (e.g., during the reference resolution process), and we must devise a way of overcoming this.

We can imagine a more convoluted case than that above, where the system has been set up so that the user may ask about the colour of a car, while colour is still being used in the icons to code non-colour information about the cars (e.g., year of manufacture, as in the above example). It has to be admitted that this would not be a well-designed representation, but it serves here to illustrate a logical possibility (Ben Amara et al. [6] discuss such an arrangement). In this situation, a phrase such as "*the green car*" might well-denote a car which is green in real life (but perhaps with a blue icon) under one context, and represent another car which is represented by a green icon (but is not actually green) under another context. Therefore, it causes ambiguity between referents during the resolution process.

We shall use the term *source* to refer to the two possible contexts for interpretation: the screen (visual display) or the domain (the represented subject matter); that is, the system has to interpret linguistic elements relative to one or other of these sources. We shall refer

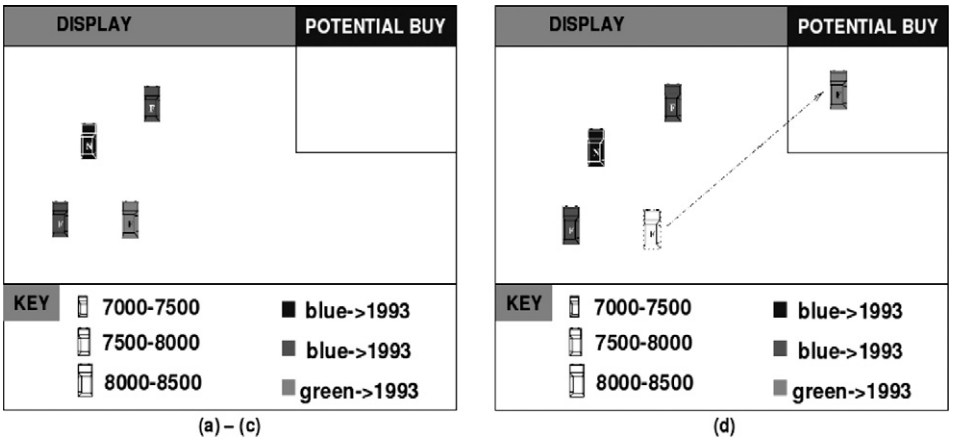

Fig. 1. Screen displays for (1).

to predicates or constants which belong to the domain as *domain predicates/constants*, and those which are associated with screen properties as *screen predicates/constants*. Where there is doubt about the source relative to which a word, phrase or sentence should be interpreted, we shall refer to this as *source ambiguity*.

Even where all the substantive terms in a phrase are derived from the same source, the phrase may refer to an entity in the other source. Consider the phrase "*the expensive car*". This, in the conventional account, has descriptive content based on the domain predicates *expensive* and *car*, and hence would conventionally denote a domain entity with these properties: something that is a car and is expensive. The phrase would indeed refer to such an entity in the sentence "*buy the expensive car*". However, it has a different reference in the sentence "*move the expensive car to the top of the screen*". In order that the screen operation *move* can make sense, the phrase must have as its intended referent the screen entity – *the icon* – that corresponds to the expensive car in the domain.

Even with a single utterance, these dual possibilities may arise. Consider the example "*move the expensive car to the top of the screen, and place an order for it*". The natural interpretation of the pronoun "*it*" allocates "*the expensive car*" as its antecedent. However, the referent of the pronoun "*it*" must, to make sense in "*place an order for it*" be a car, not an icon, while the referent of "*the expensive car*" must, for compatibility with "*move*", be a screen object. The two referents are linked by the mapping relation, but this is not the standard notion of coreference. We shall refer to such arrangements, where anaphor and antecedent denote entities linked by the mapping relation, as *quasi-coreference*.

The possibilities for source ambiguity in individual words may combine to create even more possible options. Sentence (a) in Example (2) is a relatively simple query. In a situation with source ambiguities for "*blue*" and "*small*" (but with "*car*" unambiguously a domain expression), its meaning could in theory be synonymous with any one of the four sentences (b) to (e) (where the subscripts attached to the words "*blue*", "*car*" and "*small*" indicate the source – screen or domain – in which the word is interpreted).[1]

(2) **User:** Is the blue car small? (a)
User: Is the blue_d car_d small_d ? (b)
User: Is the blue_d car_d small_s ? (c)
User: Is the blue_s car_d small_s ? (d)
User: Is the blue_s car_d small_d ? (e)

If the domain in question has a valid notion of space (unlike the car-selling domain discussed above), there is even more scope for source ambiguity. Expressions such as "*above*", "*next to*", "*to the right of*" could refer to relationships on the screen or in the domain. For example, imagine a system which allowed the user to plan the layout of a room by viewing and altering a visual display of the position of items. If the display was a plan view, "*above*" on the screen might mean "*to the right of*" in the actual room (cf. [7]).

To summarise, there are various interesting phenomena illustrated by examples like those discussed above:

(i) Predicates (such as the meaning of "*green*") may refer to visual screen properties, or to properties of objects in the database denoted by the screen objects.
(ii) Phrases (such as "*the green car*") may in certain situations contain both words referring to screen properties/items and words referring to properties/items in the domain under discussion, thus not uniformly describing any entity in either source.
(iii) A phrase which directly describes an entity in one source may, in context, denote the corresponding entity in the other source.
(iv) A pronoun may refer back to an antecedent which strictly does not denote the object that the pronoun refers to, although the two referents may be systematically related.
(v) There may even be uncertainty as to whether a particular word or phrase is to be interpreted with respect to screen information or domain information.

We do not claim that source ambiguities would always be there when the utterances as a whole is considered in context. In fact, we believe that source ambiguities can be resolved with the help of context. However, as with most reference ambiguities, a program trying to compute referents in a near-compositional fashion (i.e., figuring out the meaning from the words expressed in the text) might encounter source ambiguities before the whole meaning of the utterance is clarified. This is why we are working on a computational mechanism to resolve them.

2.2. Limitations of previous representations of ambiguities

Previous approaches to semantic ambiguity assume that the various possible meanings are all within the domain, and hence a choice can sometimes be made on the basis of domain information. This approach is neatly typified by van Deemter who presents a formalisation in which a word may be associated with several different logical expressions, with the correct expression being selectable on the basis of the sort (or type) of the variable(s) involved [48]. For example, an ambiguous word *American* might map to one of the following unambiguous expressions (notation from van Deemter):

$American_1$: $\lambda x{:}Country(DeparturePlace(x)) = USA$*(for a flight)*
$American_2$: $\lambda x{:}Country(Manufacture(x)) = USA$*(for an aircraft)*
$American_3$: $\lambda x{:}Country(PlaceOfBirth(x)) = USA$*(for a person)*

Disambiguation then results from the sort of the value of x: *flight*, *aircraft*, or *person*.

This approach is not general enough for our two-source situation. First, as stated in Section 1, the typical NL query interface contains only a symbolic representation (constants and predicates) from the application domain. No information about the screen is included. It is not difficult to systematically augment such domain information with screen objects and predicates (indeed, that forms part of our own solution), and this might allow a disambiguation approach like van Deemter's. However, this would not handle the "mixed" phrases, where some predicates within the NP contain domain information and others embody screen information. More importantly, we need not only a method of disambiguation, but a method of determining a referent set for an NP. Given the types of example we have outlined above, there needs to be some method for reference resolution which takes account of the two-source subtleties.

2.3. Our approach

Our goal is to translate combinations of locally ambiguous expressions into unambiguous representations so that the language interpreter can identify the right referents for input phrases. We tackle this by representing source information explicitly (so that it can be manipulated, and even reasoned about), and by partitioning the set of abstract entities (predicates and values to which they might apply) into two disjoint classes: *domain entities* and *screen entities*.

Using this, we define some declarative relations between predicates and their potential referents (Section 5), thereby specifying the set of candidate referents for an NP where there might be two sources involved. We also show how these definitions allow

[1] We acknowledge that some combinations could be ruled out on pragmatic grounds – is the user likely to ask about the size of a screen icon? – but they do exist as logical possibilities.

a systematic expression of these relationships as constraints (Section 7), so that processing of reference information can proceed in a general manner.

Although, we implemented our ideas within a prototype NL query system called IMIG [22], the details of that environment are not of interest here. The central idea is a formal mechanism for mapping an NP to a set of candidate referents; other issues of parsing, dialogue, integrating pointing actions, generating answers, etc., are outside the scope of the current article. All that we need to assume here is an environment which supplies certain information (see Section 4 for a summary of the knowledge bases needed).

To keep the problem to a manageable level, we concentrated on English language phenomena involving singular NPs. As is often the case in computational work on reference resolution (e.g., [52,36]) or on the generation of referring expressions (e.g., [13,24]), we have considered only NPs where the various modifiers represent a conjunction of properties which the referent(s) must meet, without allowing for relative properties (e.g., "*small elephant*" vs. "*large flea*") or predicates which are not properties of the referent (e.g., "*fake gun*"). In order to concentrate on the referential issues, we have not gone into any depth on other aspects of sentence interpretation. Our test implementation processes whole sentences, but the mechanisms are conventional and not of interest here.

3. Related work

3.1. General metonymy

The type of indirect reference which we are examining here is a particular kind of *metonymy*, in the traditional sense: 'something associated with an idea is made to serve for the expression of that idea' [43]. Such phenomena have been discussed by Nunberg [39], considering examples such as naming an author to refer to their books ("*Plato is on the top shelf*"), mentioning a publication in place of the publisher ("*The newspaper fired John*"), using the name of a bird/animal for its flesh ("*We ate chicken in bean sauce*") or a having a restaurant dish denote the customer ordering it ("*The ham sandwich is sitting at table 20*"). Nunberg argues that such usages are best captured by positing a pragmatic *referring function* which relates the explicitly mentioned item to the one actually intended. He then discusses some of the ways in which referring functions can be used, and the factors that allow language users to determine the appropriate referring function. He also notes the phenomenon which we have called quasi-coreference. Fauconnier [14] takes up Nunberg's ideas and summarises the central point thus:

> *Identification principle:* If two objects (in the most general sense), a and b, are linked by a pragmatic function F (i.e., $b = F(a)$), a description of a, d_a, may be used to identify its counterpart b.

We have not attempted to formalise the full range of effects which Nunberg and Fauconnier discuss. The main obstacle to a computational implementation which can handle general metonymic references is that detection and use of the pragmatic (referring) function may be difficult. This is because, as Koch [29] points out, metonymy has many types, and their interpretation depends on various information including speech rules, language rules, discourse rules, and even trade knowledge. Our analysis, and our small prototype implementation, are based on a particular situation where that function (our "mapping relation") is well-defined, and indeed directly provided by the multimedia setup. It is to be hoped that our methods may be generalised in some way to handle more complex and subtle forms of metonymy.

3.2. Intelligent multimodal studies

Intelligent multimodal systems have been studied since the 1980's. Some important research topics include theoretic and empirical studies of multimodal integration [11,2,40], multimodal reference resolution models [27,25,26,42,28,10], interactive multimodal systems [37,38,1], and multimodal reference in NL generation [3,35]. However, none of these projects explores the phenomena we are considering. For example, although researchers like Johnston and Pineda have developed methods that are capable of handling multimodal reference [27,25,26,42], none of them considered source ambiguities. We suggest that the key to handling source ambiguity problems is viewing the on-screen icons as different from, but related to, their corresponding domain entities. Without this, a model cannot resolve expressions involving source ambiguities. However, we acknowledge that the source ambiguity issues have been touched upon briefly by Ben Amara et al. [6], Binot et al. [7], and Chai [9], although none of them give a name to the ambiguities.

Ben Amara et al. [6] discuss natural language expressions using the graphical features of objects to refer to those objects. Their scenarios involve querying and displaying a local area network. One of their examples relates to source ambiguities – the user may enter one of the following sentences to remove a workstation shown as a purple icon:

(i) "*remove the stellar workstation*"
(ii) "*remove the purple workstation*"
(iii) "*remove the purple icon*".

They suggest that the second command is ambiguous because it is not clear whether it is the workstation in the application domain or the icon of the workstation that is purple.

The MMI[2] system is a toolkit connected to a knowledge base system [7]. It supports various forms of input from natural language, command language, graphical display, direct manipulation and gesture. The system has different modules to handle input from different modals, and then combines the processed inputs into a logical expression scheme developed for the task.

Binot et al. [7] enumerate some examples of natural language references using graphical spatial relations in an application domain that is the same as that of Ben Amara et al. Two such examples are "*add a PC to its left*" and "*remove the left workstation*". They suggest that the spatial relations in both cases are ambiguous as to whether the relations are in the domain or in the graphical representation. They do mention the possibility that the referent of a phrase might be a graphical icon, but acknowledge that they do not have a solution to, in our terminology, source ambiguity.

Chai [9] presents a semantic-based multimodal interpretation framework called MIND (Multimodal Interpretation for Natural Dialog). The framework combines semantic meanings of unimodal inputs, and forms an overall understanding of users multimodal inputs. The interpretation of the inputs integrates contextual information from domain, temporal and anaphoric aspects of the conversations, and visual and spatial aspects of the screen display. In this framework, the visual properties of certain entities (e.g., colour, highlight) can be used in the referring expressions. For example, when the system highlights two houses with colours, the users may use phrase "*the green one*" to refer to one of them. Chai states that the colour "*green*" has to be "further mapped to the internal color encoding used by graphics generation" in order to resolve the reference. However, she does not go further to examine the potential ambiguities and resolution mechanism if the colour "*green*" exists in both the domain and the visual display.

Binot et al. [7] briefly suggest that the ambiguities can be resolved by using the context of the utterance, or by asking the user

to disambiguate explicitly. In addition, the resolution process must have access to a representation of the spatial geometry of the domain. However, they do not explore the question of source ambiguity in depth, nor provide any systematic way to model and resolve the problems.

4. Meaning representation with source annotation

We have adopted a fairly conventional meaning representation language (MRL), with some extensions to allow for explicit processing of source information (and hence source ambiguity). Our MRL is closely based on the *procedural semantics* stated in [53–55] (or, more accurately, the reconstruction of Woods' ideas in [15,16]). That is, it uses logic-like constructs but is ultimately defined in terms of procedural operations upon the stored data, treating the knowledge base (for a particular domain) as providing a fairly conventional collection of objects and relations, much as in the standard Tarski semantics for first-order logic (although the language itself is not first-order). What is novel about our language is that it provides mechanisms for source information (in our sense) to be explicitly marked on predicates and constants, and for this information to be left unspecified where there is source ambiguity. That is, each predicate and constant symbol has a subscript which can be *d*, *s* or is omitted where the source is 'unspecified'.

As well as constants and variables, we also allow (as a term in an expression) the special function $\mathscr{C}_W(Y)$, which maps an item Y to its corresponding entity under the mapping relation (its subscript W indicates the source of the argument it requires, not the source of its result). This $\mathscr{C}_W(Y)$ is a particular instance of the pragmatic referring functions presented by Nunberg and Fauconnier (Section 3.1 above).

Predicates or arguments with different sources cannot be mixed: a screen-predicate cannot be applied to a domain-constant. Although there may be a screen-predicate $green_s$, and a domain-predicate $green_d$, and these may both (intuitively speaking) be representing the colour green, they are different predicates in our framework.

The exact notation used for the MRL is not central to our discussion here. A knowledge representation scheme from a different tradition (e.g., semantic nets, deductive databases) might well be equally effective. The central points are that source information must be represented explicitly, that under-specification (ambiguity) of source must be possible, and that there is some notion of terms having referents. Although our MRL does have quantifier facilities, we shall not discuss issues involving quantifiers and their scoping.

In the course of the various semantic processing, we assume the presence of the following knowledge sources (an example of these knowledge bases is in Fig. 2 below):

General model. This stores general knowledge that is independent of any particular domain.

World model. This stores specific facts about items in the domain.

Display Model. This contains information about what is currently on the screen.

Context model. This acts as a basic discourse memory, so that recently mentioned entities can be referred back to (cf. [47,46,50]).

Mapping model. This is the table, mentioned earlier, which associates each icon with exactly one database (domain) entity.

As Ben Amara et al. [6] note, to handle source ambiguities the graphical attributes on the screen should be available to referent resolution, just as those in the domain knowledge base are. The Display Model ensures that there is information about all the entities/attributes/relations in the visual display.

5. The source disambiguation problem

5.1. Sense and reference

Although source ambiguity can occur in various lexical categories (e.g., adjectives such as "*green*", verbs such as "*move*"), we have focussed our attention on its effect on, and interplay with, the resolution of references, conveyed by noun phrases.

Very broadly, the problem of reference resolution in the type of system we are considering (computer programs for interpreting natural language with respect to a finite knowledge base) can be viewed as follows: for each referring noun phrase, the system must compute a set of knowledge base identifiers, the *referents* of that phrase. This perspective goes back at least as far as Woods [53] and Winograd [52]. For a singular noun phrase, this set would be a singleton.

In order to organise this mapping from language to sets of identifiers, it is customary to interpose intermediate stages of representation, with a corresponding decomposition of the processing into stages. After some form of syntactic processing, there is normally some symbolic representation of the "meaning" of the phrase. Usually, this semantic structure fulfils a *descriptive* role, and the referent(s) are exactly those knowledge base items which meet the description [30]. For example, a phrase such as "*the green car*" would, traditionally, have a semantic structure which specifies that the referent should be of type *car* and should have the property *green*. This arrangement can be seen as a computational version of the traditional philosophical distinction between *sense* (a descriptive unit) and *reference* (the actual item denoted) [17]. Philosophers have tended to treat the reference of a phrase as being an object in the real world, but we, like most implementers of natural language query systems, have to regard referents as belonging within a symbolically represented world of knowledge-base entities.

```
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),
   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
```

Fig. 2. Relevant content of knowledge bases.

The overall computation of a referent can be viewed as depending on three sorts of information (see [45]) for a fuller discussion:

Descriptive information. This originates, as outlined above, in the "sense" or semantic structure of the phrase, and the referent must conform to it.

Semantic compatibility. The enclosing linguistic unit may supply further limits; for example, the referent of a noun phrase object must meet any selection restrictions specified for the dominating verb.

Miscellaneous inferrable constraints. These are derived from other contextual information, including domain knowledge.

Any NL interpretation system which evaluates referents assumes the first of these. It is also quite normal for a relatively simple NL system to implement a simple form of semantic compatibility. Dealing with miscellaneous contextual information is more complicated. Whatever arrangement is chosen, it is commonplace for the last two types of information (semantic compatibility and domain inference) to be seen as *narrowing down* a short-list of candidate referents, each of which already meets the first criterion: being described by the sense of the phrase, viewed in isolation. It is this which makes the phrases we have been examining problematic. That first step, the defining of candidates based on descriptive information, is less straightforward where source information (and source ambiguity) has to be considered.

What we will do now is show how a systematic definition can be given of available candidates for these multiple-source phrases, thereby allowing conventional candidate-pruning methods to proceed.

It is important to note that these are abstract set-theoretic definitions of various semantic entities and relationships which we are positing to capture the regularities cleanly; they are *not* definitions of a computational sequence of stages whereby referents are computed. The computation will be outlined later (Section 7).

5.2. From descriptive content to intended referent

In the kind of system we are considering, where words may correspond to different sources (domain or screen), the descriptive content of a referring expression in a sentence may appear to describe an entity in one source, but when considered in a larger context, e.g., with other components of the same sentence, it may actually refer to the corresponding entity in the other source. All this indicates that an extra level of representation (and processing) may be needed. The new representation is more specific than the descriptive content (sense), but is not necessarily the ultimate referent object. We define a set of item(s) which are systematically related to the descriptive content (sense) of a phrase (in the current context), taking into account the mapping relation. This set then gives rise to a (very) shortlist of candidates which includes the actual referent in context, the *intended referent* (cf. [39]).

Although all these definitions would have to be generalised to cover plural noun phrases, the concepts are made clearer by considering only singular noun phrases here.

The first building-block formalises the notion of "an entity and possibly its counterpart under the mapping relation":

Definition 1. A **described entity pair** DE is a set with one of the following forms:

(i) a singleton set containing either a domain constant or a screen constant;
(ii) a two-element set containing a domain constant and a screen constant related by the mapping relation.

Here, we take 'constant' to be the class of entity which is a potential symbolic referent. As stated in Section 2.3, we assume that the semantic content of a singular, specific, referring noun phrase can be stated as a collection (set) of predicates, which, in the simpler cases, can be interpreted as a conjunction. We also assume that there is a level of meaning representation at which specific predicates (such as `expensive`, `cheap`) have their meaning tied unambiguously to be either *domain* or *screen* predicates (as in our MRL). For example, the meaning of a phrase such as "*the cheap car*" can be approximated by a conjunction of the properties `cheap` and `car`, with both these being domain properties.

However, there could be phrases like "*the green car*" with "*green*" denoting a screen attribute instead. This makes the two predicates "*green*" and "*car*" coexist in a conjunction but one describes the screen icon and the other denotes the domain object. It is for these type of phrases we include the second clause in the definition of described entity pair, and it is this complex relationship that leads to the following definition:

Definition 2. The semantic content DC of a noun phrase **descriptively covers** a described entity pair DE iff

(i) every predicate in DC is true of at least one element in DE, with source taken into account (i.e., domain-predicates can be true only of domain-constants, and screen-predicates can be true only of screen-constants); and
(ii) for every element in DE, there is a predicate in DC which is true of the element, with source taken into account.

The second clause is to ensure that there are no extraneous elements in the described entity pair-simply enforcing the first clause would allow a collection of predicates to "descriptively cover" any set whatsoever as long as there was a subset of it which the predicates all described.

We can now put in place the additional level of representation (between semantic structure and referent) that we mentioned earlier.

Definition 3. A **described entity pair assignment** DEA is a mapping which allocates to each noun phrase NP a described entity pair DE such that the semantic content of NP descriptively covers DE.

For example, suppose that `car2` is a domain constant that is classified (in the domain KB) as a car_d, `icon3` is a constant on the screen with $green_s$ screen colour attribute, that `car2` and `icon3` correspond via the mapping relation, that "*green*" is interpretable only as a screen predicate, and "*car*" is interpretable only as a domain predicate (i.e., these lexical items have no source ambiguity in isolation). Then the semantic content of "*the green car*" consists of two predicates $\texttt{green}_s$ and $\texttt{car}_d$, and a described entity pair assignment is possible which allocates to this phrase the described entity pair {`icon3`, `car2`}. Notice that this relationship between phrases and entities is independent to the linguistic context because it does not consider the surrounding sentence structures (if any).

The use of an 'assignment' in Definition 3 is intended to capture the notion of a "reading" or an "interpretation", so that we can discuss the phenomena we are interested in without complications from other forms of ambiguity. These other factors may give rise to alternative described entity pair assignments.

Definition 4. Given a described entity pair DE, the **potential referents** of DE is the two-element set: DE $\cup$ {y | there is an x in DE which is related to y by the mapping relation}.

This is to capture the fact that a phrase like "*the cheap car*" might be allocated a described entity pair containing only one (domain) constant but might (as in an earlier example) refer in context to the (screen) entity which corresponds to that constant under the mapping relation. That related constant would in such examples not be part of the described entity pair, as it (being in the other source) would not be described by the semantic content of the noun phrase.

Table 1 shows some simple examples, where we assume that the mapping relation links (`car1`, `icon1`) and (`car2`, `icon3`).

Definition 4 defines only the *potential* referents, because the actual choice of referent will depend upon various factors, such as the surrounding linguistic context.

Definition 5. Given a described entity pair assignment DEA, an **intended referent assignment** based on DEA is a mapping IRA which allocates to each noun phrase NP an element of the potential referents of DEA(NP).

An intended referent assignment IRA(NP) is the mapping which finally allocates a referent to a noun phrase NP. Again, where there is ambiguity, there might be more than one IRA for a given phrase.

These definitions allow phrases with different described entity pairs to have the same potential referents, and thus their intended referents could be the same, as for example, the phrases "*the* $cheap_d$ car_d" and "*the* $blue_s$ $icon_s$" in Table 1.

Definitions 1–5 provide a chain from the descriptive content of a phrase to the intended referent of the phrase. That is, we have a formally defined shortlist of candidate referents, while taking into account the dimension of "source": the candidate intended referents IR for a phrase *N* are exactly the potential referents of a described entity pair *DE* which is descriptively covered by the predicates in the semantic form of *N*.

There is actually more information extractable from the NP than just a set of potential referents: there are also restrictions on these referents, in terms of the predicates which must apply to them (such as $cheap_d$ and $blue_s$), and what their sources might be. We shall outline below (Section 7) how all this locally extractable information can be very simply encoded as constraints, thereby allowing these definitions to interact with more general contextual or linguistic information in a process which results in the selection of the intended referent from the potential referents (i.e., the role of the IRA).

6. Anaphora

The phenomenon that we called (Section 2.1) *quasi-coreference* (a pronoun being related to an antecedent with a different source), is captured by the following two definitions.

Definition 6. Noun Phrases N1 and N2 are **coreferential** iff they have the same intended referents.

Definition 7. Noun Phrases N1 and N2 are **quasi-coreferential** iff their intended referents correspond under the mapping relation.

We have not specifically investigated anaphoric phenomena within our framework, but we have ensured that those mechanisms which we have developed do cater for the possibility of quasi-coreference.

Table 1
Representations for examples in the text

Phrase	Semantic predicates	DE	Potential referents
"*the green car*"	{$green_s$, car_d}	{`car2`, `icon3`}	{`car2`, `icon3`}
"*the cheap car*"	{$cheap_d$, car_d}	{`car1`}	{`car1`, `icon1`}
"*the blue icon*"	{$blue_s$, $icon_s$}	{`icon1`}	{`car1`, `icon1`}

6.1. Syntax salience information

Pronoun resolution often makes use of syntactic regularities (e.g., [31,23]) and salience factors (e.g., [47,46,8,32,19]). In certain applications, it may be helpful to impose some form of compatibility between the semantic content of the anaphor and antecedent phrases (e.g., [49]). All these are generally used to *find candidate antecedent phrases* for an anaphor.

Our simplified way of finding candidate antecedents is to allow referring expressions (including anaphors) take their referents either from the Display Model (items on the screen) or the Context model (items recently mentioned), or the counterparts (under the mapping relation) of the items in these two models, but we have not imposed any other salience or syntactic constraints on this process (although such restrictions could be added). The inclusion of counterparts allows quasi-coreference. (It might seem that this prevents items not already on the screen ever being mentioned in the dialogue, but we do allow quantified commands to extract items from the knowledge base. For example, "*Display all expensive cars*" is not treated as involving a referring expression, but would cause the corresponding icons of certain domain entities displayed on the screen, and hence in the Display Model.)

6.2. Referential inference

Some form of inference (perhaps limited) is sometimes used to assess whether the interpretation in which the denotations of the phrases are the same is plausible or even possible. This is typically the ultimate arbiter for *evaluating or eliminating* possibilities.

Although we have not looked at real inference (beyond intra-sentential semantic constraints), our constraint-extraction process automatically allows quasi-coreference to occur; see Section 7.2 below.

There is no explicit linking of anaphor to antecedent phrase; the linkage is only implicit, in that both phrases are assigned the same (or corresponding) referents.

7. Formulation as a constraint problem

7.1. Constraint satisfaction and reference resolution

Constraint satisfaction, as a way of sifting through candidates when there may be various requirements to be met, has a long history in artificial intelligence [51,34]. Its use for reference resolution was first proposed by Mellish [36]; see also [20]. In a computer system for understanding natural language within a small finite world model (the situation we are considering), there is a limited set of possible referents for any given referring phrase. However, there may be interrelationships between these phrases (for example, if they occur in the same sentence) which impose relationships between their possible referents. Hence the choice of referent for each phrase cannot be made in isolation, but must take into account the choices being made for other related phrases. Constraint satisfaction is a formal search procedure (or more accurately, a family of search methods) designed for just this kind of situation. It has the advantage of generality, so that inferred contextual information can be handled by the same uniform process as semantic compatibility restrictions.

A constraint problem consists of a number of *variables*, each with an associated *value-set* to which its value must belong,[2] and a number of *constraints* between the variables.

[2] Value-sets are usually called "domains", but as we are already using the word "domain" for another technical concept, we have opted for this expression.

In a situation where there is just one "source", the candidate set for each noun phrase could be straightforwardly narrowed down to all the objects which were described by the meaning of the noun phrase, in keeping with the Fregean tradition. This, as argued above, has to be amended to cover the examples we are concerned with, as this descriptive relationship does not always hold when there are separate sources. However, if we can provide a suitably tractable definition of how to frame the content of an NP as restrictions on its referents, the constraint satisfaction method will once again be applicable. We have laid the basis for that in Section 5: our constraint satisfaction system relies on our formal definitions in deciding the set of potential referents for each NP, and any inherent restrictions on those objects.

7.2. Extracting the constraints

Once the semantic representation of an input sentence has been constructed (in our MRL formalism), a process scans this formula (and the lexicon) to determine which variables and constraints should constitute the CSP. That is, it builds up (quite separately from the semantic representation of the input text) a set of CSP variables and a set of constraints (predicates) linking these variables. In our problem, there are just two types of CSP variables. A variable ranging across *entities* (possible intended referents, for example) has a value-set containing the entities in the Context model and the Display model, together with any entities that correspond to them through the mapping relation (see Section 6). A variable ranging across *sources* has the value-set {`screen`, `domain`}.

A noun phrase such as "*the blue car*" will, in its semantic representation, involve two predicates, $blue_d$ and car_d. In setting up the CSP, the system creates one entity-variable for each predicate in the phrase, and inserts a constraint that the predicate must be true of that variable (more precisely, must apply to any binding for that variable). In the worked example that we give in Section 9, these variables appear as *DE*11 and *DE*12, and the predicate constraints as *blue*(*DE*11) and *car*(*DE*12). This ensures that, no matter how many CSP variables are created in this way for a given phrase, the set of entities that they are bound to will be 'descriptively covered' by the semantic predicates, in the sense of Definition 2 earlier. (These variables will be referred to here as "described entity" variables, as this captures their function fairly well.) Also inserted are variables for the sources of the entities, and the appropriate *have_source* relations – see the worked example for details.

The constraint-builder also inserts constraints between every pair of variables in this set using the predicate `related`. This (symmetric) predicate is true of two entities if either they are the same entity or they correspond under the mapping relation (in either direction). Stipulating that the entities bound to the phrase's variables meet this constraint, in all possible pairs, ensures that the set of entities bound to these variables forms a 'described entity pair' in the sense of Definition 1 earlier. (Formally, `related` is an equivalence relation whose equivalence sets are the maximal described entity pairs possible under the current mapping relation.)

The `related` predicate is used to constrain the possible intended referents. For a referring noun phrase, a single variable is set up for its intended referent, IR. A `related` constraint is established between this IR variable and at least one of the described entity variables. This means that the IR variable can be bound only to an entity in the 'potential referents' (Definition 4). That is, the entity for the IR variable is either in the described entity pair or corresponds under the mapping relation to an element of that described entity pair.

Since the set of values allowed for entity variables is restricted (as noted earlier) to entities in the Display Model and the Context Model (and their counterparts under the mapping relation), the flexibility of the `related` relation automatically allows quasi-coreference as well as conventional coreference. Hence our formal model (Section 5) is enforced indirectly via these instances of the `related` constraint.

It is a crucial part of the constraint expressions for an input sentence that the sources of each item are explicitly represented (with constraints linking them to their items), so that constraints can be stated on the sources, for example, that two sources must be the same, or that a particular source must be `screen`.

There is a further small technical point which is necessary to follow the example that we give in Section 9. It is central to our way of expressing the various relations between referent variables that the relation `related` is transitive. However, a basic constraint-solving algorithm of the type we have adopted does not automatically compute transitive closures of relations. We have therefore chosen to an algorithm for inserting *all* the `related` links in initialising the constraints for the solver. In particular, the link from an IR variable to the DE variables for that phrase need (logically) be made only once, but the algorithm links the IR variable to all the DE variables for that phrase.

8. Heuristics for source disambiguation

8.1. Constraints and preferences

When designing constraint-satisfaction methods, it is commonplace to have two sorts of rules for computing semantic properties or relationships: *constraints* which eliminate candidates, and *preferences* which express some form of ordering or priority on possible solutions. If a value does not satisfy a constraint, it is rejected, whereas a preference does not lead to rejection. The constraint propagation aims at getting rid of any value in a candidate set that does not satisfy a constraint, but may not narrow down the possibilities to unique values. In such a situation, preferences can be used to select a particular value from the available candidates, thus initiating a new constraint propagation to narrow down other candidate sets accordingly. If this still leaves some sets over-populated (having more candidates than is required), further preferences, if available, can be applied. Note that preferences can be viewed as weak constraints. Their purpose is to steer the resolution process to a more suitable direction without actually committing to it. How often preferences are needed really depends on the complexity of the input sentences and the states of the knowledge bases.

The initial information extracted from noun phrases NP (see Section 7.2) is all in the form of constraints. However, we have formulated a small number of generalisations about references within sentences in a two-source setup, and these regularities can contribute further information for the reference resolution process, if stated as constraints or preferences. There is space here to present only a few representative heuristics; a more detailed discussion can be found in [22]. All of these rules are relatively tentative, and our central proposals about two-source references do not crucially depend on the correctness of their content, but they illustrate the way in which further observations about source and references can be used uniformly.

8.2. Constraints

Our illustrative example of a linguistic constraint relies on a small amount of syntactic information about the NP. For this, we adopt a fairly conventional analysis of NP structure in English (e.g., [21]).

In a phrase like "*the blue nissan car in the corner of the screen*", "*car*" is the *head noun*, and all the other substructures constitute *modifiers*, either *pre-modifiers* (before the head noun) or *post-modifiers* (after the head noun). Where a noun such as "*nissan*" is used as a modifier just before a head noun in an adjective-like role, it is a *classifier*.

The mapping relation between screen entities and domain entities is not used symmetrically in linguistic expressions. For example, in our sample car-sales application, it would be natural to say that "*a square represents a car*", but not to say that "*a car represents a square*". This asymmetry restricts the use of head nouns. Intuitively, a screen entity, such as a `square`, can be referred to by a head noun naming its screen source, such as "$square_s$", or by a word naming the source of the domain entity it represents, such as "car_d" (where subscripts *s* or *d* indicate the source we are postulating.) However, a domain entity such as a `car` cannot be referred to by a head noun naming the corresponding screen source; that is, it is very odd to refer to a `car` as a "$square_s$". We can summarise this pattern by Rule 1.

RULE 1. If the head noun of a phrase unambiguously names a screen predicate, then the intended referent of the phrase must be a screen entity.

Sentences which violate this rule sound unacceptably strange; for example, in our example application, "*buy_d the $icon_s$" and "*is the $square_s$ $cheap_d$?". (Of course, "*buy the icon*" would be acceptable in an application where the domain of discussion was icons rather than cars, but that would be "buy_d the $icon_d$".)

Where the head noun unambiguously denotes a screen object (as in Rule 1), it seems that the modifiers must denote only screen attributes. For example, while phrases like "*a red_s $icon_s$*" and "*a $small_s$ $circle_s$*" are meaningful expressions, phrases like "**the $expensive_d$ $icon_s$*" and "**every $large_d$ $square_s$*" (where non-screen modifiers combine with a screen head noun) do not sound felicitous.

In contrast, if the head noun is from the domain (so not covered by Rule 1), there is no strong restriction on modifiers. So, phrases like "*the $expensive_d$ car_d in_s the $corner_s$*", "*the red_s car_d*", "*a $small_d$ $saloon_d$*" are all acceptable.

However, there are exceptions. A domain word acting as the classifier component can sometimes be so closely bonded to the head noun denoting a fairly general screen class that the whole phrase is still meaningful in that combination; for example, "*the $Nissan_d$ $icon_s$*". The reason could be that the classifier component is not functioning as a modifier in this case, but helping to form a compound head noun. We have not found any clear generalisations regarding classifier modifiers.

If, as suggested above, the modifiers are screen words/phrases, it follows they can describe only screen entities; hence, Rule 2:

RULE 2. If the head noun of a phrase *N* unambiguously names a screen predicate then, when computing a described entity pair assignment for *N*, non-classifier modifiers of *N* can be true only of screen entities.

Both our rules so far cover situations where the head noun unambiguously names a screen predicate, so we can combine them as the *screen head noun rule*:

RULE 3. (*Screen head noun rule*) If the head noun of a phrase unambiguously names a screen predicate, then:

- the intended referent of the phrase must be a screen entity,
- the non-classifier modifiers must be true of screen entities (in defining possible described entity pairs for the phrase).

8.3. Preferences

As stated, we have devised a small number of preferences relating to allocation of source to words or phrases. It is recognised that at present we have relatively little data to work from in creating these, and they are based mainly on intuitive analogies with other linguistic contexts. It is possible that study, e.g., of dialogues in design contexts would provide further guidance; but it is also possible that the preferences depend on various factors that vary by context, making such a comparison misleading. It is not easy to say how far, if a system of the kind we propose were to gain widespread use, our preferences might have to be extended to cope with new kinds of phenomena, or indeed with the ways in which users of the system might adapt or change their existing linguistic preferences. The proposals here are intended merely to be illustrative of the kinds of preferences that we expect to be appropriate for this type of system, rather than absolutely accurate empirically.

8.3.1. Semantic types

We organise semantic entities into a hierarchy of types, where near the top there are three major classes: those which are solely domain types (e.g., *car*), those which are solely screen types (e.g., *icon*) and those which appear in both (e.g., *colour*). As well as being useful in various ways during semantic processing, this can be used to help determine the source of words in cases where the word is ambiguous between screen and display (for example, with colours or sizes).

The first observation is based on our intuitions together with observations made in the literature about the construction of referring expressions which in some way parallel a preceding phrase. Ritchie [44] suggests that where an adjective pre-modifies an anaphoric "*one*", it is more acceptable if it is directly comparable to the corresponding adjective in the antecedent phrase:

- "*That is a red bottle and this is a blue one.*"
- "*This is a wine bottle and that is a whisky one.*"
- *??* "*That is a red bottle and this is a whisky one.*"

Levelt, in reviewing psycholinguistic studies, observes that "*the speaker apparently contrasted the new referent object with the previous one*" [33], and Gatt [18] argues that when describing more than one object, there is a natural tendency to use the same "perspective", to increase coherence.

In our illustrative application, we suggest that in a phrase such as "*is the upholstery in the red car blue?*" it is most natural to assume that the two colour words "*red*" and "*blue*" are either both screen attributes or both domain attributes; to mix sources here would be slightly perverse (though not impossible – this is a preference, not a constraint, and could in practice be overridden by pragmatic considerations).

We have therefore formulated the following rule:

RULE 4. (*Same Type Rule*) If two words in the same sentence have the same semantic type (i.e., if there is a concept in the hierarchy which directly dominates both), then they *probably* have the same source.[3]

8.3.2. Structural parallelism

A closely related proposal is that where two phrases in a sentence have analogous structure, corresponding elements in those

[3] We have used the wording "probably" to emphasise that this is a preference rather than a constraint.

structures will be from the same source. For example, in the sentence "*Delete the small car to the right of the expensive car*" the pre-modifiers "*small*" and "*expensive*" are in comparable positions, and we suggest that they therefore are likely to have the same source.

RULE 5. (*Same Position Rule*) If two words are at the same position within two phrases, and the two phrases have the same structure, then the two words *probably* have the same source.

The criteria for defining two phrases as having the same structure are: *they have the same type of head noun, and either they both have post-modifiers, or neither has.*

In some cases, both the Same Type Rule and the Same Position Rule may apply (e.g., example in Section 9).

9. A worked example

We will work through the input sentence (3) to demonstrate how constraint solving is used in our system.

(3) "*delete the red car at the right of the blue car*"

The MRL expression for this sentence has a structure whose nesting is analogous to the syntax of 3; that is, the representation of "*the blue car*" is a substructure inside a large representation of "*at the right of the blue car*", which in turn is a subpart within the representation of the whole phrase "*the red car at the right of the blue car*". Finally, the representation of the phrase forms the argument to the operation denoted by the verb "*delete*".

In the following account of the CSP variables generated from this, variables with names `DE**` are what we called "described entity variables", ranging over entities, and variables called `Sde**` are to be bound to sources of described entity variables. Variables with names of the form `IR*` and `Sir*` represent an intended referent and its source, respectively.

The portion of MRL corresponding to "*the blue car*" gives rise to two DE variables, `DE11` (from `blue`) and `DE12` (from `car`), and two corresponding source variables `Sde11`, `Sde12`. Following the procedure sketched in Section 7.2, there are these constraints: *blue*(*DE*11), *car*(*DE*12), *have_source*(*DE*11,*Sde*11), *have_source*(*DE*12,*Sde*12), and *related*(*DE*11,*DE*12). Also for this phrase, there is a variable for the intended referent, `IR1`, and its source, `Sir1`, with constraints *have_source*(*IR*1,*Sir*1), *related*(*DE*11,*IR*1), *related*(*DE*12,*IR*1) (see Section 7.2 for the rationale for these).

The phrase "*the red car*" gives rise to a similar set of variables `DE21`, `DE22`, `Sde21`, `Sde22`, `IR2` and `Sir2`, with an analogous set of constraints.

The prepositional phrase "*at the right of*" is regarded as using the *intended* referent of the embedded phrase "*the blue car*" to add descriptive information to the NP whose core is "*the red car*"; that is, it contributes to defining the *described entities* for that larger phrase. This arrangement is not obvious, but it seems to capture the idea that "*at the right of*" has to take a spatial location as its argument (at the referential level), but the whole adjunct "*at the right of the blue car*" is part of the semantic content (i.e., descriptive material) for the larger NP. This perspective leads to a constraint *right_of*(*DE*23,*IR*1), where *DE*23 is a further variable generated for the `right_of` predicate, standing for whatever entity is to the right of *IR*1 (in keeping with the procedure of Section 7.2). Similarly, the verb "*delete*" depicts a "removable" attribute of the intended referent (represented by variable *IR*2 here) of the whole phrase "*the red car at the right of the blue car.*" This raises the constraint *removable*(*IR*2).

The set of constraints produced from the whole NP is shown in Table 2.

We have adopted Mackworth's AC-3 consistency algorithm [34], which achieves node and arc consistency within the network of constraints. If we assume the relevant knowledge bases contain the information summarised in Fig. 2, then Table 3 shows the results after applying this algorithm. In this example, when network consistency is achieved, not every variable has the right number of candidates (i.e., one) in its candidate set (see the column "after AC-3" in Table 3). Therefore, it is appropriate to try to apply preferences. In the phrases "*the blue car*" and "*the red car*", the modifiers "*blue*" and "*red*" meet the conditions of the Same Type Rule and of the Same Position Rule, both of which indicate a preference for these items to have the same source. The system therefore selects {`screen`} as the value for `Sde11`. After applying the preference, the network needs to reapply the consistency algorithms, which results in the solution shown in the last column of Table 3.

All of the above ideas have been tested in an implementation which, although primarily an exploratory testbed, embeds the constraint mechanism in a relatively realistic multi-modal environment [22]. In this, the mechanism achieves its task of identifying potential referents for NPs.

10. Conclusions

We have provided a formalisation of a form of metonymic reference that is very basic and limited, but does occur naturally in a certain type of applications. The formal account centres on a set of definitions which systematically relate the linguistic semantic content of a noun phrase to a set of potential symbolic referents, and which can be framed succinctly in terms of constraints imposing equivalences between abstract referent candidates. Providing a route to a set of potential referents allows existing computational mechanisms (e.g., inference) to be applied. We have also proposed some filtering heuristics which are specific to phrases that may refer in this indirect way.

An obvious extension is to see whether these ideas can be used in other, more general, forms of metonymy, and if so, how. Even within our limited application, there is scope for further improvement:

(i) There may be other generalisations about source-ambiguous forms (or even more general metonymic phrases) which could be added to the set of constraints.
(ii) A wider range of linguistic constructs could be covered. For example, anaphora, including reflexive pronouns, could be more thoroughly explored.
(iii) The heuristics used in the constraint-solving are mainly based on our intuition and experiments on a very small number of test dialogues. It would be beneficial to have a multimodal dialogue corpus upon which heuristics could be tested and on which further generalisations could be based. Statistical methods or machine learning approaches might be useful for gathering relevant data.

Table 2
All the constraints raised from (3)

Con1	*domain*(*Sde*12)	Con2	*domain*(*Sde*22)
Con3	*screen*(*Sde*23)	Con4	*same_source*(*Sde*23, *Sir*1)
Con5	*blue*(*DE*11)	Con6	*car*(*DE*12)
Con7	*red*(*DE*21)	Con8	*car*(*DE*22)
Con9	*right_of*(*DE*23,*IR*1)	Con10	*removable*(*IR*2)
Con11	*have_source*(*DE*11,*Sde*11)	Con12	*have_source*(*DE*12, *Sde*12)
Con13	*have_source*(*DE*21,*Sde*21)	Con14	*have_source*(*DE*22, *Sde*22)
Con15	*have_source*(*DE*23,*Sde*23)	Con16	*have_source*(*IR*1, *Sir*1)
Con17	*have_source*(*IR*2,*Sir*2)	Con18	*related*(*DE*11, *DE*12)
Con19	*related*(*DE*11,*IR*1)	Con20	*related*(*DE*12, *IR*1)
Con21	*related*(*DE*21,*DE*22)	Con22	*related*(*DE*21, *DE*23)
Con23	*related*(*DE*21,*IR*2)	Con24	*related*(*DE*22, *DE*23)
Con25	*related*(*DE*22,*IR*2)	Con26	*related*(*DE*23, *IR*2)

Table 3
The candidate sets of variables in Table 2 during constraint satisfaction

Variable	Initial candidates	Candidates after NC	After AC-3	After preference
Sde11	{screen, domain}	{screen, domain}	{screen, domain}	{screen}
Sde12	{screen, domain}	{domain}	{domain}	{domain}
Sde21	{screen, domain}	{screen}	{screen}	{screen}
Sde22	{screen, domain}	{domain}	{domain}	{domain}
Sde23	{screen, domain}	{screen, domain}	{screen, domain}	{screen}
Sir1	{screen, domain}	{screen, domain}	{screen, domain}	{screen}
Sir2	{screen, domain}	{screen}	{screen}	{screen}
DE11	{*}	{carl, icon3}	{carl, icon3}	{icon3}
DE12	{*}	{carl,car2, car3}	{carl, car3}	{car3}
DE21	{*}	{icon1, icon2}	{icon1, icon2}	{icon1}
DE22	{*}	{carl,car2, car3}	{carl, car2}	{carl}
DE23	{*}	{*}	{icon1, car2}	{icon1}
IR1	{*}	{*}	{carl, icon3}	{icon3}
IR2	{*}	{icon1,icon2,icon3}	{icon1, icon2}	{icon1}

{*} = {carl, car3, icon3, icon1, icon2, car2}; NC = node consistency.

(iv) Mapping relations are used during the resolution of reference, 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 plausibly be used in an interaction.

(v) The problem addressed in this paper assumes that there are only two sources, the domain and the screen. The question arises – could there be more than one source? We have not encountered any such situation, so it is hard to be sure what the correct generalisation would be in such a case.

Acknowledgements

The first author (Daqing He) was supported by a Colin & Ethel Gordon Studentship from the University of Edinburgh. The work reported here was carried out at the University of Edinburgh.

References

[1] J. Allgayer, K. Harbusch, A. Kobsa, C. Reddig, N. Reithinger, D. Schmauks, XTRA: a natural language access system to expert systems, International Journal of Man–Machine Studies 31 (1989) 161–195.
[2] E. André, T. Rist, The design of illustrated documents as a planning task, in: M. Maybury (Ed.), Intelligent Multimedia Interfaces, AAAI/MIT Press, Menlo Park, CA/Cambridge, MA, 1993, pp. 94–116.
[3] E. André, T. Rist, Referring to world objects with text and pictures, in: Proceedings of the 15th International Conference on Computational Linguistics(COLING'94), Kyoto Japan, 1994, pp. 530–534.
[4] I. Androutsopoulos, G. Ritchie, Database interfaces, in: R. Dale, H. Moisl, H. Somers (Eds.), Handbook of Natural Language Processing, Marcel Dekker, New York, 2000, pp. 209–240. chapter 9.
[5] I. Androutsopoulos, G. Ritchie, P. Thanisch, Natural language interfaces to databases – an introduction, Journal of Natural Language Engineering 1 (1) (1995) 29–81.
[6] H. Ben Amara, B. Peroche, H. Chappel, M. Wilson, Graphical interaction in a multi-modal interface, in: Proceedings of Esprit Conferences, Kluwer Academic Publisher, Dordrecht, Netherlands, 1991, pp. 303–321.
[7] J. Binot, L. Debille, D. Sedlock, B. Vandecapelle, H. Chappel, M.D. Wilson, Multimodal integration in MMI^2: anaphora resolution and mode selection, in: H. Luczak, A. Cakir, G. Cakir (Eds.), Work With Display Units-WWDU'92 (Abstract Book of 3rd. International Scientific Conference), Berlin, Germany, 1992.
[8] S.E. Brennan, M.W. Friedman, C.J. Pollard, A centering approach to pronouns, in: Proceedings of 25th Annual Meeting of the Association of Computational Linguistics (ACL-87), Stanford, CA, 1987, pp. 155–162.
[9] J. Chai, Semantics-based representation for multimodal interpretation in conversational systems, in: Proceedings of International Conference on Computational Linguistics (COLING), 2002.
[10] J.Y. Chai, P. Hong, M.X. Zhou, A probabilistic approach to reference resolution in multimodal user interfaces, in: Proceedings of Intelligent User Interfaces 04, 2004, pp. 70–77.
[11] P. Cohen, M. Dalrymple, D. Moran, F. Pereira, J. Sullivan, R. Gargan Jr., J. Schlossberg, S. Tyler, Synergistic use of direct manipulation and natural language, in: Proceedings of the 1989 Conference on Human Factors in Computing Systems (CHI'89), 1989, pp. 227–233.
[12] A. Copestake, K. Sparck Jones, Natural language interfaces to databases, The Knowledge Engineering Review 5 (4) (1990) 225–249.
[13] R. Dale, N. Haddock, Generating referring expressions involving relations, in: Proceedings of European Chapter of Association of Computational Linguistics (ECAL-91), 1991, pp. 161–166.
[14] G. Fauconnier, Mental Spaces: Aspects of Meaning Construction in Natural Language, Cambridge University Press, Cambridge, UK, 1994. First published by MIT Press, 1985.
[15] A.A. Fernandes, A Meaning Representation Language for Natural-Language Interfaces, Master's thesis, Department of Artificial Intelligence, University of Edinburgh, 1990.
[16] A.A. Fernandes, G. Ritchie, D. Moffat, A Formal Reconstruction of Procedural Semantics, Technical Report DAI-RP-678, Department of Artificial Intelligence, University of Edinburgh, 1994.
[17] G. Frege, Sense and reference, in: D. Davidson, G. Harman (Eds.), The Logic of Grammar, Dickenson, Encino, CA, 1892, pp. 116–128. Reprinted in 1975.
[18] A. Gatt, Structuring knowledge for reference generation: a clustering algorithm, in: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Trento, Italy, 2006, pp. 321–328.
[19] B.J. Grosz, A.K. Joshi, S. Weinstein, Centering: a framework for modelling the local coherence of discourse, Computational Linguistics 21 (2) (1995) 203–225.
[20] N.J. Haddock, Incremental Semantics and Interactive Syntactic Processing, Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland, 1988.
[21] M. Halliday, An Introduction to Functional Grammar, second ed., Edward Arnold, London, UK, 1994.
[22] D. He, References to Graphical Objects in Interactive Multimodal Queries, Ph.D. thesis, Division of Informatics, University of Edinburgh, Edinburgh, Scotland, 2001.
[23] J.R. Hobbs, Resolving pronoun references, Lingua 44 (1978) 311–338.
[24] H. Horacek, An algorithm for generating referential descriptions with flexible interfaces, in: Proceeding of the 35th Annual Meeting of the Association for Computational Linguistics/8th Conference of the European Chapter of the Association for Computational Linguistics, 1997, pp. 206–213.
[25] M. Johnston, Unification-based multimodal parsing, in: Proceedings of the Conference of COLING-ACL'98, Montreal, Canada, 1998, pp. 624–630.
[26] M. Johnston, S. Bangalore, Finite-state multimodal integration and understanding, Natural Language Engineering 11 (2) (2005) 159–187.
[27] M. Johnston, P. Cohen, D. McGee, S. Oviatt, J. Pittman, I. Smith, Unification-based multimodal integration, in: Proceedings of 35th ACL conference ACL-97, 1997, pp. 281–288.
[28] H. Kim, J. Seo, Resolution of referring expressions in a Korean multimodal dialogue system, ACM Transactions on Asian Language Information Processing 2 (4) (2003) 324–337.
[29] P. Koch, Metonymy between pragmatics, reference and diachrony, metaphorik.de, Available from: <http://www.metaphorik.de/07/>, 2004.
[30] A. Kronfeld, Reference and Computation: An Essay in Applied Philosophy of Language, Studies in Natural Language Processing, Cambridge University Press, Cambridge, UK, 1990.
[31] R. Langacker, Pronominalization and the chain of command, in: D. Reibel, S. Schane (Eds.), Modern Studies in English, Prentice-Hall, Englewood Cliffs, NJ, 1969.
[32] S. Lappin, H.J. Leass, An algorithm for pronominal anaphora resolution, Computational Linguistics 20 (4) (1994) 535–561.
[33] W.J.M. Levelt, Speaking: From Intention to Articulation, A Bradford Book, MIT Press, Cambridge, MA, 1989.
[34] A.K. Mackworth, Consistency in networks of relations, Artificial Intelligence 8 (1977) 99–118.
[35] K. McKeown, S. Feiner, J. Robin, D. Seligmann, M. Tanenblatt, Generating cross-references for multimedia explanation, in: Proceedings of 10th National

Conference of the American Association for Artificial Intelligence (AAAI'92), San Jose, CA, 1992, pp. 9—16.
[36] C.S. Mellish, Computer Interpretation of Natural Language Descriptions, Ellis Horwood Series in Artificial Intelligence, Ellis Horwood, 1985.
[37] J. Neal, Z. Dobes, K. Bettinger, J. Byoun, Multimodal references in human–computer dialogue, in: Proceedings of 6th National Conference of the American Association for Artificial Intelligence (AAAI'88), St. Paul, MN, USA, 1988, pp. 819–823.
[38] J. Neal, S. Shapiro, Intelligent multi-media interface technology, in: J. Sullivan, S. Tyler (Eds.), Intelligent User Interfaces, ACM Press, New York, NY, 1991, pp. 11–44.
[39] G. Nunberg, The non-uniqueness of semantic solutions:polysemy, Linguistics and Philosophy 3 (2) (1979).
[40] S. Oviatt, A. DeAngeli, K. Kuhn, Integration and synchronization of input modes during multimodal human–computer interaction, in: Referring Phenomena in a Multimedia Context and Their Computational Treatment, A Meeting of the ACL Special Interest Group on Multimedia Language Processing, Madrid, Spain, 1997, pp. 1–13.
[41] C. Perrault, B. Grosz, Natural language interfaces, in: H. Shrobe (Ed.), Exploring Artificial Intelligence, Morgan Kaufmann Publishers Inc., San Mateo, California, 1988, pp. 133–172.
[42] L. Pineda, G. Garza, A model for multimodal reference resolution, Computational Linguistics 26 (2) (2000) 139–193.
[43] H. Read, English Prose Style, Bell and Sons, London, 1934.
[44] G. Ritchie, Computer Modelling of English Grammar, Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland, CST-1-77, Department of Computer Science, 1977.
[45] G. Ritchie, Semantics in parsing, in: M. King (Ed.), Parsing Natural Language, Academic Press, North Holland, 1983, pp. 199–217.
[46] C. Sidner, Focusing in the comprehension of definite anaphora, in: B. Grosz, K.S. Jones, B.L. Webber (Eds.), Reading in Natural Language Processing, Morgan Kaufmann, Los Altos, CA., 1987, pp. 363–394.
[47] C.L. Sidner, Focusing for interpretation of pronouns, American Journal of Computational Linguistics 7 (4) (1981) 217–231.
[48] K. Van Deemter, Towards a logic of ambiguous expressions, in: K. Van Deemter, S. Peters (Eds.), Semantic Ambiguity and Underspecification, CSLI Publications, 1995, pp. 203–237.
[49] R. Vieira, Definite Description Processing in Unrestricted Text, Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland, 1997.
[50] M. Walker, Centering, anaphora resolution, and discourse structure, in: M. Walker, A. Joshi, E. Prince (Eds.), Centering Theory in Discourse, Clarendon Press, Oxford, England, 1997.
[51] D. Waltz, Understanding line drawings of scenes, in: P.H. Winston (Ed.), The Psychology of Computer Vision, McGraw-Hill, New York, 1975.
[52] T. Winograd, Understanding Natural Language, Academic Press, New York, 1972.
[53] W. Woods, Semantics for a Question-Answering System, Ph.D. thesis, Harvard University, 1967.
[54] W. Woods, Procedural semantics for a question-answering machine, in: Proceedings for the Fall Joint Computer Conference, AFIPS, New York, NY, 1968, pp. 457–471.
[55] W. Woods, Semantics and quantification in natural language question answering, in: M. Rubinoff, M. Yovits (Eds.), Advances in Computers, Academic Press, New York, NY, 1978, pp. 2–64.