PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1988,
edited by ICOT. © ICOT, 1988

ADAM: An Extension of Situation Semantics for Practical Use

Chisato Numaoka and Mario Tokoro

Department of Electrical Engineering
Keio University
3-14-1 Hiyoshi, Yokohama 223, Japan

ABSTRACT

This paper describes a model of discourse understanding called ADAM (Actual Discourse Assistant Model), a practical extension of Situation Semantics. ADAM gives a more intuitive framework for understanding discourse than situation semantics by using the concept of object orientation. In ADAM, objects are regarded as the conveyers of information. Information is extracted from objects themselves and the relations between objects. Furthermore, the behavior of a knowledge system depends on the information. ADAM has the following features:

- A world in ADAM is composed of a set of *situation objects*, which are characterized by the three primitive objects: temporal location, spatial location, and individuals.
- A discourse situation object is a possible world. Anaphoric references and focuses of discourse are interpreted in a discourse situation.

Because of the above features, ADAM gives us a factual framework for handling the difficulties of discourse understanding. Additionally, its framework simplifies construction of a user interface for knowledge systems using natural language.

1 INTRODUCTION

Natural languages give more friendly and flexible man-machine interfaces than existing command languages. However, natural languages simultaneously yield ambiguity in communication. In order to reduce ambiguity, discourse understanding is indispensable. Situation semantics [Barwise and Perry 1983] [Barwise 1985] [Barwise 1986] is a very attractive theory to explain some of the problems in discourse understanding by use of situation and set theories. One of the features of situation semantics is that described situations are interpreted within the relationship of real situations and abstract situation types. Owing to this feature, situation semantics can explain why a difference of situation causes a difference of interpretation.

Actually, many studies in discourse understanding are being carried out using this feature of situation semantics. DUALS [Yasukawa et al. 1985] and SCORE [Nagase 1986] are examples of discourse understanding systems which use situation semantics and utilize logic programming. The situated language research program in CSLI [CSLI 1987] is an integrated study based on situation semantics. In this program, many fundamental and applied studies for situation semantics are being carried out.

This present paper takes a pragmatic approach to representing the framework for discourse understanding. That is to say, we propose a method which can be practically used for the framework of situation semantics to be applied to a natural language user interface for a knowledge system.

As a model for building knowledge systems, we proposed DKOM (Distributed Knowledge Object Modeling) [Tokoro and Ishikawa 1984] in FGCS'84. In this model, every object contains a local knowledge base and is called a knowledge object. This knowledge object has powerful mechanisms to construct a knowledge system in a well-formed representation.

ADAM (Actual Discourse Assistant Model), which we propose in this paper, is an extended model of situation semantics which can yield a natural language interface system based on the idea of DKOM. Extensions to situation semantics are made in representing objects in discourse by using knowledge objects, and extracting information from the relation between objects and from within objects themselves. In order to understand a discourse, we construct a model of a speaker's mind within an individual discourse situation.

In this paper, section 2 explains an overview of the discourse understanding model ADAM. In Section 3, we present the representation of situations and events in terms of objects. Section 4 gives an example of discourse understanding using ADAM. In Section 5, we discuss features of ADAM from the viewpoint of the discourse understanding model. Finally, related works are described.

2 THE MODEL

2.1 World Répresentation using Object

In DKOM, two additional features to object oriented programming were described: the concurrent execution of objects and the local knowledge base for each object. This object in DKOM is called the knowledge object. These two additional features are very important for knowledge systems. A knowledge system has to deal with two types of object. One is the object as a component of the knowledge system, and the other is the knowledge itself. In many knowledge systems, the latter is taken as simply a symbol or frame. However, as for the knowledge whose structure can change, a knowledge object is more useful than a symbol or frame. This is because a knowledge object has a local knowledge base. Thus we propose a discourse understanding model based on the notion of knowledge objects.

Since a knowledge in ADAM is represented as a set of objects, the definition of an object is shown first. An object in ADAM is defined as follows:

(Definition 1-1) An object is composed of the following resources: individual, temporal-location, spatial-location, and situation [1].

(Definition 1-2) For some object i, if

$$i \in dom(f)$$

$$\vee$$

$$i \notin dom(f) \wedge i \in dom(f') \wedge f \longrightarrow f'$$

then

object i exists,

where

both f and f' are discourse situations.

Note: dom(f) denotes the domain of discourse situation f. And $f \longrightarrow f'$ denotes that f can refer to f'.

(Definition 1-3) In some discourse situations, if an object i exists, then it can be referred to. The converse is also true.

(Definition 1-4) In some discourse situations, if two different objects a and b exist,

$$a \neq b$$

is always true. On the contrary, if

$$a = b$$

is true, object a and b are one and the same.

(Definition 1-5) Any object has its internal states. Thus, all objects which are able to refer to object a has the capability to know the internal states of object a.

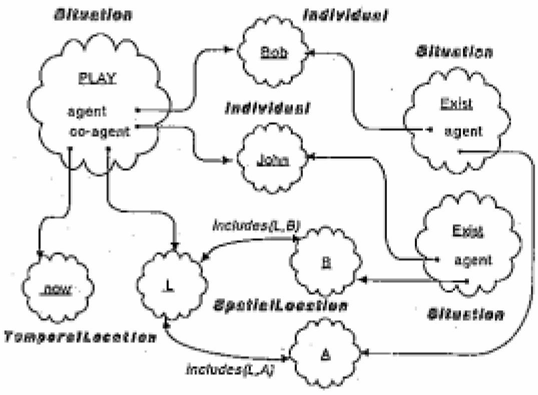

Figure 1: The Representation of a Situation in ADAM

2.2 Situation

In ADAM, the situation object is defined as follows:

(Definition 2-1) A situation object is a special frame object, which is composed of a relation, primitives, and a truth value.

(Definition 2-2) Relation is described as an n-arity predicate, such as

$$relation(X_1, X_2, ..., X_n).$$

(Definition 2-3) A situation object is a kind of domain.

The situation object is described in the following form for simplicity:

$$situation(relation, X_1, X_2, ..., X_n; TV).$$

where TV means truth value.

2.3 Primitive Objects

In ADAM, an utterance situation is composed of *primitive objects.* There are three kinds of primitive objects: temporal-location, spatial-location, and individual. By using these primitive objects, the situations for the sentence *"Bob is playing with John"* are represented as shown in Figure 1.

Here, owing to the division of the spatio-temporal location in situation semantics into the temporal-location and the spatial-location, each location comes to be related with actual computational resources.

2.3.1 Temporal-Location (TL)

In this model, two types of representation of time are given. One is the absolute temporal-location, and the other is the relative temporal-location.

[1] In general, this term has two meanings. One is for states of affairs, and the other is for courses of events. Our use of the term is limited to the first meaning. That is, a situation represents a static relation between objects.

- Absolute temporal-location: a time is *absolute* if it is specified in a sentence. Current (computer-)system time is also an absolute temporal-location. Therefore, the time when the utterances are made is of this type.
- Relative temporal-location: a time is *relative* if it is not specified in a sentence.

The temporal-location, which we have defined here, is different from that of interval temporal logic [Moszkowski 1983]. In ADAM, time is not regarded as divided fields but as a point on a continuous axis.

2.3.2 Spatial-Location (SL)

There are also two types of location in spatial-location. One is absolute spatial-location, the other is relative spatial-location.

- Absolute spatial-location: a point in a 3-dimensional space is *absolute* if it is specified in a sentence.
- Relative spatial-location: a point in a 3-dimensional space is *relative* if it is not specified in a sentence. In this case, if some place has already been mentioned before, that place is used for this situation. Otherwise, an adequate spatial-location will be assigned to the situation.

2.3.3 Individual (Ind)

An individual is either a physical or a conceptual entity. Each individual in a discourse situation is a particular existence in the real world. In addition, in this model, an individual has its related knowledge. Thus, if we can access an object, then we can obtain the object's specific knowledge. An individual object is also a unit of inference [Numaoka 1988].

2.4 Discourse Situation

In situation semantics, a discourse situation is explained as a special situation which includes three specific roles: the speaker, the addressee, and the described situations. We agree with situation semantics at this point. Situation semantics, however, does not give more details about the discourse situation. Our defined discourse situation object is regarded as a personal knowledge base. That is, by discourse with some person, the knowledge about the person is created in the discourse situation object for the person's exclusive use. This kind of function is useful in the situation that some person wants to inform another person something by using a knowledge system as the medium. In ADAM, a discourse situation is defined as follows:

(Definition 3-1) A discourse situation is an event. That is, the situation is a transitional one. The transitional situation causes a series of sentences in the discourse situation to be described in the frame of the situation.

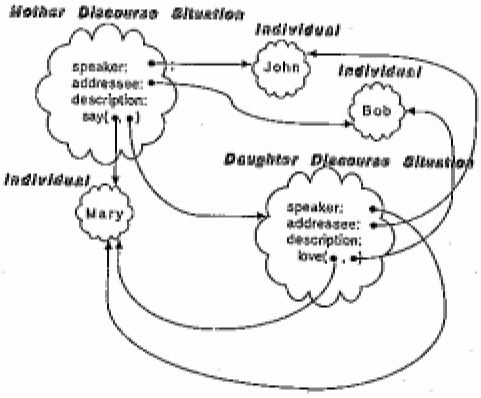

Figure 2: The Hierarchical Relation in a Discourse Situation

(Definition 3-2) If a discourse situation exists in another discourse situation as a described situation, the former is called *the daughter discourse situation* and the latter is called *the mother discourse situation*. Hence, a mother discourse situation can refer to the daughter discourse situations while a daughter discourse situation cannot refer to the mother situations.

These definitions allow us to express the parallel-world in discourse as shown in [Nakashima 1986].

In ADAM, the hierarchical relation of discourse is shown in the following example:

> *John said to Bob, "Mary says that she loves you."*

In this case, the discourse situation between Mary and John is included in the discourse situation between John and Bob as shown in Figure 2.

2.5 Event

Event in ADAM is classified as follows:

- Event from the point of view of the speaker
- Event from the point of view of the addressee

As for the latter, more specific classification is assumed:

- An event which is necessarily made in the course of a process to understand a situation
- An event which is a set of the speaker's utterance situations

As for the former, we will show an example in Section 4. As for the latter, we give the following definition:

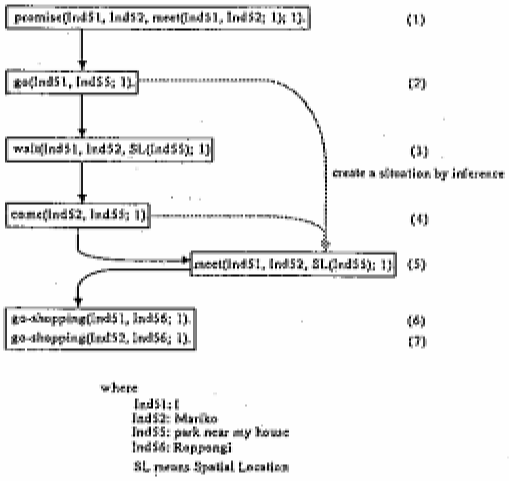
promise(Ind51, Ind52, meet(Ind51, Ind52; 1); 1).
(1)
go(Ind51, Ind55; 1).
(2)
wait(Ind51, Ind52, SL(Ind55); 1)
(3)
create a situation by inference
come(Ind52, Ind55; 1)
(4)
meet(Ind51, Ind52, SL(Ind55); 1)
(5)
go-shopping(Ind51, Ind56; 1).
(6)
go-shopping(Ind52, Ind56; 1).
(7)
where
Ind51: I
Ind52: Mariko
Ind55: park near my house
Ind56: Roppongi
SL means Spatial Location

Figure 3: Event as a Sequence of Situations

(Definition 4-1) If a set of individual object(s) is restricted according to any rules, the chain of situation objects, which is related with this set of individual object(s) and have a causal relation with each other, is *a course of events.*

Assume the following example:

> *I had promised to meet my friend Mariko yesterday. So, I went to the park near my house and waited for her. Fifteen minutes later, she came there. And then, we went shopping in Roppongi.*

In this case, three points of view are available for the course of events.

(Case 1) Assume that I and Mariko are included in the restricted set of individual objects.
See Figure 3.

(Case 2) Assume that I am included in the set. In this case, the event corresponds to the sequence that lacks situation (4) and (7) in Case 1.

(Case 3) Assume that Mariko is included in the set. In this case, the event corresponds to the sequence (4) → (5) → (7) in Case 1.

Here, both Case 2 and Case 3 are obviously subsets of Case 1. That is, for Case 2 and Case 3, two types of event can be defined.

Creating a course of events is effective for the decision of the topic of a discourse because it can restrict the scope of the knowledge system's interest.

2.6 Assumptive Situation

In discourse, a sentence may give an attitude report. This type of sentence does not describe a real situation. For example,

> *I believe that he will pass the examination.*

In this sentence, the fact that the speaker believes that someone will pass the examination is described. However, the situation in which the speaker believes may not be realized. It is only an assumption. Accordingly, this type of situation should be expressed as a situation object because a situation object is made as the result of a situation being realized. In the utterance,

> *Though I wanted to go to the States, I did not go there after all.*

The description *"I wanted to go to the States"* includes the description *"I go to the States"*. This embedded description is expressed as follows:

situation(go, I, the States; 1). (1)

Here, though it is assumed that this situation occurs at some temporal-location TL in the past, TL is not fixed yet. The remainder of the above utterance includes the description *"I did not go"*. This is expressed as follows:

situation(go, I, the States; 0). (2)

Here, though it is also assumed that the situation occurs at some temporal-location TL' in the past, TL' is not fixed yet, either. If we assume that the situation objects (1) and (2) are referring to the same real situation, a contradiction occurs. Accordingly, situation objects (1) and (2) are not the same (i.e., multiple situation objects, which are referring to the same real situation, cannot exist). However, if the situation object (1) and (2) are different objects, the real situations which correspond to these objects are also different. This indicates that the two different situations occur simultaneously. These two situation objects have the same individuals, the same temporal-location, and the same spatial-location. The difference between these objects is only their truth-value. This is impossible in the real world, if we do not assume multiple worlds. Thus, this is again a contradiction.

In order to avoid the contradiction, we introduced the idea of the assumptive situation in ADAM. This type of situation object represents a situation whose occurrence has not been realized yet. The assumptive situation is described as follows:

(Definition 5-1) An assumptive situation object describes a situation other than that which has already occurred or is occurring now (i.e., an interrogative sentence).

(Definition 5-2) A situation object has priority over an assumptive situation object. That is, if an assumptive situation object *A* and a situation object *S* exist, which is unifiable with the assumptive situation except for the truth-value, then we can assume that they describe the same real situation.

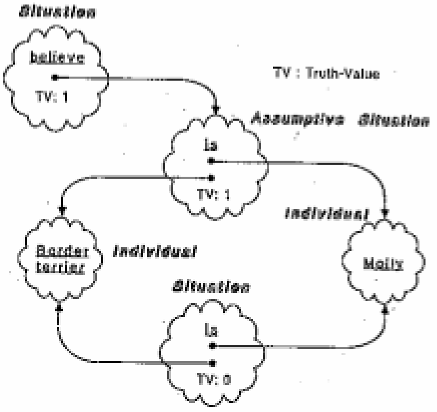

Figure 4: Assumptive Situation and Situation: In this case, the situation object, "Fred believed that ...," is not a successful situation, because its embedded assumptive situation, "Molly is a Border terrier," was not realized.

(Definition 5-3) In addition to Definition 5-2, if the truth-values are the same, then the situation object in which the assumptive situation object is included is called *a successful situation.*

(Definition 5-4) An assumptive situation object is a pseudo object. Definition 1-4 is inapplicable for this type of object.

Assume the following utterances:

Fred believed that Molly was a Border terrier.
But the dog is not a Border terrier.

Here, the first utterance includes an assumptive situation object, Border-terrier(Molly; 1), and the second one includes a situation object, Border-terrier(Molly; 0). This is shown in Figure 4.

3 OBJECT ORIENTED REPRESENTATION OF DISCOURSE

In this section, we show the representation of a discourse situation object and the event management. We can represent an object in ADAM using a Knowledge Object (KO) in DKOM. The features of KO are as follows:

- Procedural description is given in its behavior part. The behavior of an object in a knowledge system should be defined by a well-defined control sequence. For example, in the description of an event manager, this part is a great help.
- Declarative description is described in its knowledge-based part. This part has an efficient framework for the inference using meta knowledge for the object. For example, constraints between objects should be described in this part. All the knowledge about the object is registered in this part as a predicate.
- Communication control and the predicate demon is described in its monitor part. This mechanism is useful for describing constraints.
- Each KO runs in parallel. By utilizing this function, a knowledge system can make inference concurrently as described in [Numaoka 1988].

Thus, we use Orient84/K [Ishikawa 1987], which is a programming language based on DKOM.

3.1 Discourse Situation Object

A discourse situation object is represented as an instance of class ADAMDiscourseSituation. The most basic feature of this object is that it creates a situation object. For this purpose, a method *makeSituation:* is defined in this class. One of the important roles of this ADAM object is object identification. Assume the following sentence:

Do you remember the girl whom we saw in Central Park yesterday?

It is necessary to understand what the phrase *"the girl whom we saw yesterday"* indicates. After syntactic analysis, the following structure is created:

```
[topic=[situation, object],
 situation=[relation=see,
            agent=we,
            object=[attribute=specific,
                    semantics=girl],
            temporal-location=yesterday,
            spatial-location=Central park],
 tense=past]
```

Here, item *topic* indicates that item *object*, which is included in item *situation*, is the focus of the utterance. An ADAMDiscourseSituation object has a method *anaphoric-Operation:* to resolve an anaphoric reference. In this case, by executing this method, the ADAMSituation object, *"we see a girl,"* that occurred in the past, is taken out from the current ADAMDiscourseSituation object. If this ADAMSituation object is *Ind034*, the given sentence can be simply represented as follows:

Do you remember Ind034?

The management of focuses in a discourse is also an important issue to be treated in an ADAMDiscourseSituation object. This object has four local memories in order to manage focuses in a discourse. These are for individual, temporal location, spatial location, and situation as can be seen in Figure 5. The management mechanism of focuses in discourse is as follows:

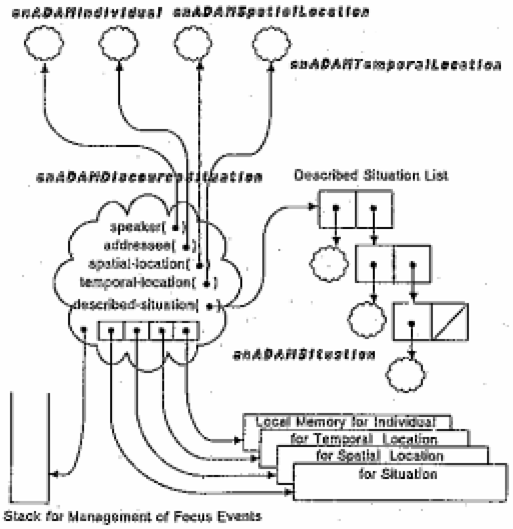

Figure 5: Configuration of a Discourse Situation Object

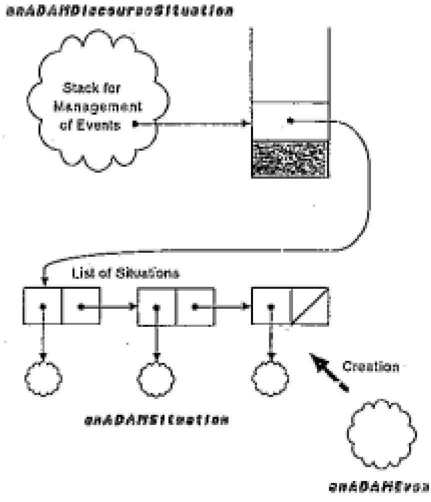

Figure 6: Mechanism for Event Management

- **(REGISTRATION)** After syntactic analysis, if item *topic* exists, it is registered in a corresponding local memory. If it does not exist, item *subject* is registered in the individual local memory.

- **(SEARCH)** After syntactic analysis, if any indeterminate terms exist, each of them is the last registered object in a corresponding local memory. As for the temporal location and the spatial location, the description of these location objects being included in the last registered location object is registered in the class ADAMTemporalLocation object.

3.2 Functions of the Event Management

The functions for managing events are realized by class ADAMEventManager. The role of ADAMEventManager is the identification of object, creation of events, and so on. These operations are based on the definitions of events in section 2.

There are two important roles for the ADAMEventManager. One is to produce the chain of ADAMSituation objects which is restricted by individuals in an utterance. The other is to evaluate an ADAMSituation objects and to make an event on a computer.

As for the former, assume the following sentence:

> *Do you remember what I talked about with you yesterday?*

In this case, it is necessary to identify the event *"what the speaker talked about with the addressee yesterday"*. The ADAMEventManager, first, creates a list of situations, which is sorted for temporal location in ascending order (see Figure 6). Second, the ADAMEventManager pushes the list on the stack for events. By these operations, the event, *"what the speaker talked about with the addressee yesterday,"* is produced. For these operations, ADAMEventManager supports method *makeSituationSequenceFor:*.

In order to realize the latter role, ADAMEventManager has method *evaluateSituation*:

```
evaluateSituation: aSituation
    | aRelation anEvent |
    aRelation ← aSituation relation.
        "take a relation out from the situation."
    foeachUnify(event(aRelation, ?anEvent))
        do: ["If an event which is related with
                            the relation is defined,
                then evaluate the event."
            (ADAMEventManager
                perform: anEvent
                with: aSituation argument)
                ifTrue: [↑true]].
                    "If the operation is success,
                    then return from this method."
    ↑false
```

Additionally, predicate *event* is described as follows:

```
event(#add, #addNumber:).
    "If relation is #add,
    then call the method addNumber:."
    ...
```

4 DISCOURSE UNDERSTANDING USING ADAM

In the previous sections, we have shown an overview of our model, ADAM. In this section, we explain with two examples how ADAM is utilized in discourse understanding. The first example will show how discourse situations are used for comprehending anaphoric reference. In the second, we will explain how a course of events occurs along with the temporal location.

4.1 Topic of Discourse

User: *When I was talking with Mariko yesterday,*
she said something. *(1)*
System: *What did she say?* *(2)*
User: *She said that she was going to go to the*
States. *(3)*
And she said that she would go to study
there, *(4)*
but I think that she will go sightseeing. *(5)*

In this case, we assume that a discourse situation object between the user and the system has already been created. For sentence (1), the following operations are executed:

- By the utterance *" When I was talking with Mariko yesterday, "* two discourse situation objects between the user and Mariko are created. One is for the situation in which the speaker is the user, and in the other, Mariko.
- By the utterance *"She said something, "* the current discourse situation moves to the daughter discourse situation object whose speaker is Mariko. Sentence (2) represents the movement of interest which is caused by the movement of the discourse situation object. The utterance *"something "* is a cataphora. As this cataphora corresponds to the daughter discourse situation object, the situation *"She said something "* is represented as follows:

  situation(say, Ind056, DSit012, TL03; 1).(a)
  where
  DSit012: *daughter discourse situation object*
  TL03: *yesterday*
  Ind056: *Mariko*
  DSit012: daughter discourse situation *object*

  In addition, this situation is placed in the mother discourse situation object as a described situation.

For sentences (3), the following operations are executed:

- A situation *"she is going to go to the States "* is registered in the current discourse situation object as a described situation. In this case, *"is going to "* indicates an instant in the future. This is described as follows:

  whollyPrecedes(now, TL053).

  Furthermore, as this situation has a future temporal location, it is created as an assumptive discourse situation object.

  assumptive-situation(go, Ind056,
  Ind035, TL053; 1). *(b)*
  where
  Ind035: the States

- The utterance *"she said that she would go to study there "* is represented as follows:

  situation(say, Ind056,
  assumptive-situation(go, Ind056, Ind035,
  goal(assumptive-situation(study,
  Ind056, ?tl1; 1)), ?tl0; 1),
  ?tl; 1).
  where a symbol prefixed by '?' indicates indeterminate term.

Here, discourse understanding starts by searching for a situation object which is unifiable with the situation, *"Mariko says something at sometime. "* In this case, the object (a) is an answer. Thus, ?tl = TL03, and,

assumptive-situation(go, Ind056, Ind035,
goal(assumptive-situation(study, Ind056,
?tl1; 1)), ?tl0; 1)
=DSit012 *(*)*

The unification between a(n) (assumptive) situation object and a discourse situation object succeeds if the (assumptive) situation object is included in the discourse situation object. Then, the existence of following two assumptive situations is guaranteed:

assumptive-situation(go, Ind056, Ind035,
?tl0; 1). *(c)*
assumptive-situation(study, Ind056,
?tl1; 1). *(d)*

As the result, the situation object (b) unifies with (c). Thus, ?tl0 = TL053. Here, the assumptive situation object (d) is created because it does not exist in the discourse situation object. And the unification (*) is success.

- By the utterance, *"I think that she will go sightseeing, "* the focus move back to the mother current discourse situation (i.e., the mother discourse situation) again. As the situation, *"she goes sightseeing, "* is composed with the following two assumptive situation:

  assumptive-situation(go, Ind056, ?place,
  ?tl0; 1). *(c')*
  assumptive-situation(sightseeing, Ind056,
  ?tl1; 1). *(d')*

These assumptive situations must be found in the referable domain. Here, (c) and (c') are unifiable, but (d) and (d') are not unifiable. Thus, a new assumptive situation is created (see Figure 7).

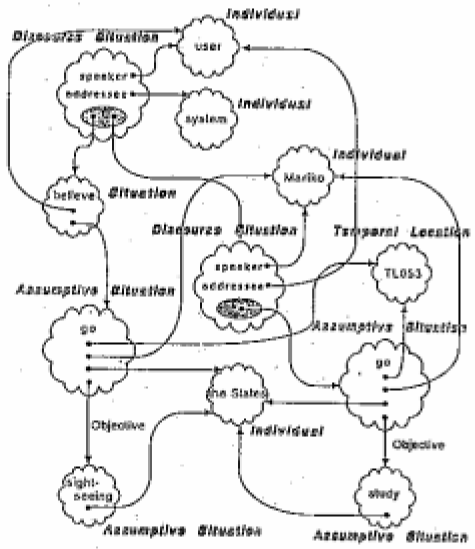

Figure 7: Using Discourse Situation in Recognizing Topic

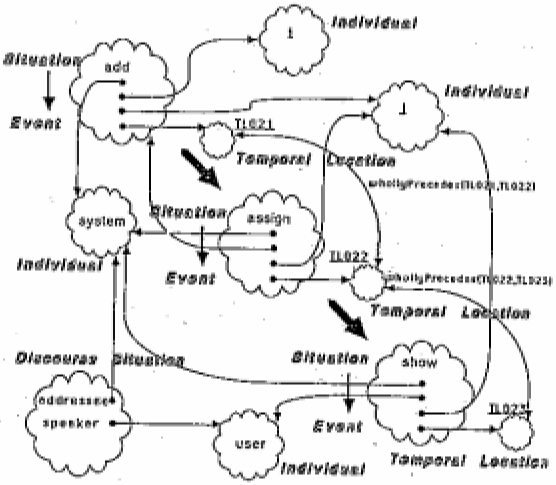

Figure 8: Course of Events

4.2 Event in Arithmetic Calculation

Assume that the following text is given:

There are two variables, i and j. *(1)*
The initial value of variable i is 1,
and that of j is 2. *(2)*
Assign the result of i plus j to j. *(3)*
Show me the value of j. *(4)*

These utterances are understood in a discourse situation object between the user and the system. First, by utterance (1), the following situation is created:

situation(exist, GR005, TL021; 1).
where
GL005: a group object which
includes two individual objects
corresponding to i and j.

Here, i and j are stored in the local memory as a focus. Second, by utterance (2), the following two situation objects are produced:

situation(is, initial-value(Ind034),1,
TL022; 1). (a)
situation(is, initial-value(Ind035), 2,
TL022; 1). (b)
where
Ind035: the variable i
Ind036: the variable j

Because the situation objects (a) and (b) have a relation *is*, the second argument, 1 and 2, is assigned to the slot *initial-value* of Ind034 and Ind035, respectively. Utterance (3) creates the following situation object:

Sit110 = situation(add, Ind035, Ind036,
TL023; 1). *(c)*
situation(assign, result(Sit110),
TL024; 1). *(d)*

Additionally, the following relations for temporal location are made:

whollyPrecedes(TL022, TL023).
whollyPrecedes(TL023, TL024).

Here, as utterance (3) is a command, the situation objects (c) and (d) are evaluated as an event (see Figure 8). Finally, the following situation object for utterance (4) is created:

situation(show, Ind001, Ind012,
value(Ind035); 1).

And this situation object is evaluated because it is also a command. As a result, the value *"3"* is given.

5 DISCUSSION

In this section, we will discuss certain features of ADAM.

Discourse Understanding Model for Practical Use Our primary objective in building ADAM is to propose a model of discourse understanding which can directly be used for application. This is because situation semantics, which gives an appropriate model for discourse understanding, is an explanatory theory and thus it cannot directly be applied for practical use. In order to realize this objective, we introduced an object oriented approach to situation semantics.

First, ADAM is such an object oriented model. The advantages of ADAM are as follows:

- Owing to representing an object in discourse by using object oriented computation, we can realize an object in discourse as a frame, whose slots are changeable. For example, an individual object can form a specific image within that object.

- We can take information which relates with an object out from itself directly.

- Inheritance enables us to represent a thesaurus.

- Procedures can be described within an object in order to utilize many kinds of computational resources.

Second, in ADAM, the spatio-temporal location is divided into the spatial location and the temporal location. This is helpful for a computer system to understand the speaker's requirement for computer resources. Because computer resources is composed of the temporal resources (e.g. CPU time) and the spatial resources (e.g. memory).

Third, ADAM has an extended interpretation for events as we have already described in section 2. That is, ADAM can deal with the course of events. Therefore, we can take the chain of situations which is restricted by individuals, and we can produce an event by calling an existent application program on a computer.

Management of Issues for Discourse Understanding

Anaphoric Reference Any object as knowledge (e.g., individual, location, situation) in ADAM is represented as an instance object, and it is managed in the corresponding class object. In addition, an object referred to in a discourse is included in a specific discourse situation object. In particular, a recently referred object is in the local memory of the discourse situation object. Cataphora, in many cases, is resolved by referring to these local memories. If it cannot be found in the local memories, then the proper discourse situation is referred to. If it cannot still be found, then the proper class object is referred to. By using this three step reference mechanism, anaphoric reference is treated efficiently.

Acquisition of Implicit Information In the current framework of ADAM, no method for acquisition of implicit information is given. However, as explicit information taken from an utterance is represented as an object in a computer system, it is possible to treat an acquisition of implicit information by using the relation between the objects in a knowledge system.

Recognition of the Focus of Discourse A discourse situation object has a local memory which contains recently referred objects. In ADAM, those objects are the objects which are the focus of utterances. Furthermore, the discourse manager in ADAM manages discourse situation objects. These two mechanisms enable us to comprehend the focus of discourse.

6 RELATED WORK

Logic programming certainly plays an important role for natural language processing, especially for parsing and inference. In addition, theories for natural language understanding are formalized based on logic. Situation semantics is one such a theory. Therefore, several discourse understanding systems based on situation semantics have been implemented in the logic programming style. DUALS [Yasukawa et al. 1985] is a famous discourse understanding system which has been developed in ICOT. This system has the following features:

- It utilizes situation semantics in the framework of semantics and pragmatics.

- It efficiently solves partial problems from the viewpoint of logic programming.

- It uses the Kameyama's model [Kameyama 1984] for anaphoric reference in Japanese, which is one of the most important issues in discourse understanding.

SCORE [Nagase 1986] provides a knowledge representation form. This representation form is based on situation semantics, and a world is represented as a set of situations. The features of this form are as follows:

- It manages the acquisition of implicit information by using classified constraints.

- It represents nominal concepts hierarchically.

The system using SCORE resembles DUALS in their representation forms. These systems cannot utilize information for spatio-temporal location. In contrast, ADAM supports an efficient framework for operating a spatio-temporal location.

ODDS [Ohsawa 1985] is a system which utilizes the object oriented programming style. This system has the following features:

- It forms a semantic structure using a case frame.

- It grasps the focus of discourse using a situation object.

The ODDS system resembles ADAM in their object oriented programming styles and situation objects. In the ODDS system, the situation itself forms a domain for understanding the focus of discourse. In contrast, ADAM utilizes a discourse situation object. In discourse, as the speaker's connection is important, we believe our model is superior to the model of ODDS.

7 CONCLUSION

In this paper, we have described an extended model of situation semantics for discourse understanding, called ADAM (Actual Discourse Assistant Model), and have shown some examples of discourse understanding using it.

Our efforts were based on the belief that the object oriented programming paradigm supports a highly effective framework for utilizing the applications using natural language interface system. Thus, ADAM gives us a model of situation semantics for practical use using DKOM, which is an object oriented programming methodology.

In ADAM, an object is a unit of knowledge and an information medium. An object which is represented in the computer is connected with the speaker's knowledge about the object. By utilizing this user specific knowledge, a knowledge system can reduce some of ambiguities of discourse. Thus, if speaker want to operate some computational resources which are managed by a discourse understanding system, the system understands the intention of the speaker and can operate the resources. Objects in ADAM are basically composed of the situations and the primitive objects which compose situations. Primitive objects are composed of the individual, temporal location, and spatial location. These are basic elements of the real world. A real situation is related to these real elements. Thus, we can represent situations in a world by using these primitives.

A discourse situation object is used for expressing the user specific world and for confirming the topic in a discourse. As this object is a kind of domain, we can utilize this object in order to restrict the objects which can be referred in a situation. Furthermore, the extended interpretation of events enables the natural language interface to be applied to the existing application program.

The prototype discourse understanding system based on ADAM has been implemented in Orient84/K. In this paper, the full specification of this system have not been given, but the specification of the discourse situation object and event management mechanism have been described, since these play the most important roles in ADAM. The functions of knowledge object of Orient84/K, a local knowledge base and concurrency, has been useful for the implementation of ADAM.

Finally, we have discussed the significance of ADAM for the study of discourse understanding and the advantages of object oriented representation for situation semantics. We have already developed a prototype discourse understanding system and implemented a story understanding system and a question-answering system. Based on these experiences, we have started to develop the discourse understanding system ASUKA [Numaoka 1987] on which we are going to develop a natural language interface system.

ACKNOWLEDGEMENT

The authors are indebted to Takeo Maruichi and Tadashi Nagano for their valuable comments. The authors are also grateful to the members of ICOT WG2; their DUALS programs inspired this work.

REFERENCES

[Barwise and Perry 1983] Barwise, J., J. Perry, *Situation and Attitudes*, MIT Press, 1983.

[Barwise 1985] Barwise, J., *Notes on Situation Theory and Situation Semantics for CSLI Summer School*, CSLI Summer School, at Stanford, July 1985.

[Barwise 1986] Barwise, J., *On the Circumstantial Relation between Meaning and Content*, CSLI, 1986.

[CSLI 1987] CSLI, *Fourth Year Report of the Situated Language Research Program*, Report No.CSLI-87-111, 1987.

[Ishikawa 1987] Ishikawa, Y., M. Tokoro, *Orient84/K: An Object-Oriented Concurrent Programming Language for Knowledge Representation*, in Object-Oriented Concurrent Programming, A.Yonezawa and M.Tokoro(eds.), MIT Press, 1987.

[Kameyama 1984] Kameyama, M., *Centering in Japanese*, Ph.D.Thesis,Dept.of Linguistics, Stanford University, 1984.

[Moszkowski 1983] Moszkowski, B., Z. Manna, *Reasoning in Interval Temporal Logic*, Stanford University, Department of Computer Science, Report No.STAN-CS-83-969, 1983.

[Nagase 1986] Nagase, T., H. Mizoguchi, *An Attempt to Design Discourse Understanding System Based on Situation Theory*, Proc. of the Third Annual Conference of JSSST, pp.37-40(in Japanese), 1986.

[Nakashima 1986] Nakashima, H., *Uranus Reference Manual*, Bul.Electrotech.Lab., Vol.50 No.8, 1986.

[Numaoka 1987] Numaoka, C., M. Tokoro, *ASUKA: Features and Specifications of a Discourse Understanding System*, Proc. of the Fourth Annual Conference of JSSST, pp.55-58(in Japanese), 1987.

[Numaoka 1988] Numaoka, C., M. Tokoro, *A Concurrent Approach to Abduction in ADAM*, Proc. of the Fifth Annual Conference of JSSST, pp.129-132(in Japanese), 1988.

[Ohsawa 1985] Ohsawa, I., A. Yonezawa, *An Object Oriented Approach for Constructing a Dialog Understanding System*, Computer Software, Vol.2, No.1, pp.11-28(in Japanese), 1985.

[Tokoro and Ishikawa 1984] Tokoro, M. and Ishikawa, Y., *An Object Oriented Approach to Knowledge Systems*, Proc. of the Int'l Conf. on FGCS, Nov. 1984.

[Yasukawa et al. 1985] Yasukawa, H., Hirakawa, H., Mukai, K., Miyoshi, H. and Tanaka, Y., *The Outline of Discourse Understanding System DUALS*, Tech.Memo, TM-0118, 1985.