DESIGN AND IMPLEMENTATION OF AN INTERACTIVE OBJECT MODELLING SYSTEM

Regine Becher
Institute of Computer Science and Engineering, University of Karlsruhe (TH)
Germany

Peter Steinhaus
Institute of Computer Science and Engineering, University of Karlsruhe (TH)
Germany

Raoul Zöllner
Institute of Computer Science and Engineering, University of Karlsruhe (TH)
Germany

Rüdiger Dillmann
Institute of Computer Science and Engineering, University of Karlsruhe (TH)
Germany

Speaker: Regine Becher, Institute of Computer Science and Engineering, Universität Karlsruhe (TH), CSE Dillmann, Gebäude 07.21, Postfach 69 80, D-76128 Karlsruhe, Phone: ++49 (0)721 608 5944, becher@ira.uka.de

Topic: 1. Research and Development
Keywords: Object modelling, service robot, human-robot interaction

Introduction

In recent years, interest in research areas like service robotics has been concentrating more and more on real-life applications in human-centered environments, like e.g. assisting humans in households. In such unstructured and highly dynamic environments, it is no longer possible to model in advance every possible piece of world knowledge that e.g. a service robot system might need. Such world knowledge includes amongst others knowledge about objects, actions and tasks, and about human users and their preferences. The following paper concentrates mainly on the acquisition of knowledge about objects and their features. For object models in particular, it is often suggested that a service robot system operating in such an environment should be able to build up new object models and enhance old ones autonomously (e.g. Pollefeys/Van Gool 2002, Miller et.al. 2003).

Although we think that autonomous learning of object knowledge is very important for such a system and proves to be a great ease for the human user, on the other hand we do not think that it can learn every possible type of object knowledge completely autonomously for several reasons. First, user-specific information like a favourite cup of a certain user is very hard to infer from sensor data. Secondly, human help can ease the task of learning new objects or new aspects of objects considerably. Being confronted with a previously unknown object, it will be much harder, if not impossible, for a robot system to determine the object's name, properties, functionalities and the class of objects it belongs to without any help of a human. Sometimes, it simply is easier to learn something from somebody else than on your own. Thirdly, it is very important for such applications that the robot and the human user can interact with each other. Therefore, it is helpful if the robot represents objects and their features in roughly the same categories as the human to ensure a sensible communication about things. Setting up object models interactively ensures that the structure of the object models and the knowledge that is included are as compatible to human object knowledge as possible. For these reasons, we developed and implemented a system which allows for an interactive, semi-autonomous learning of object models.

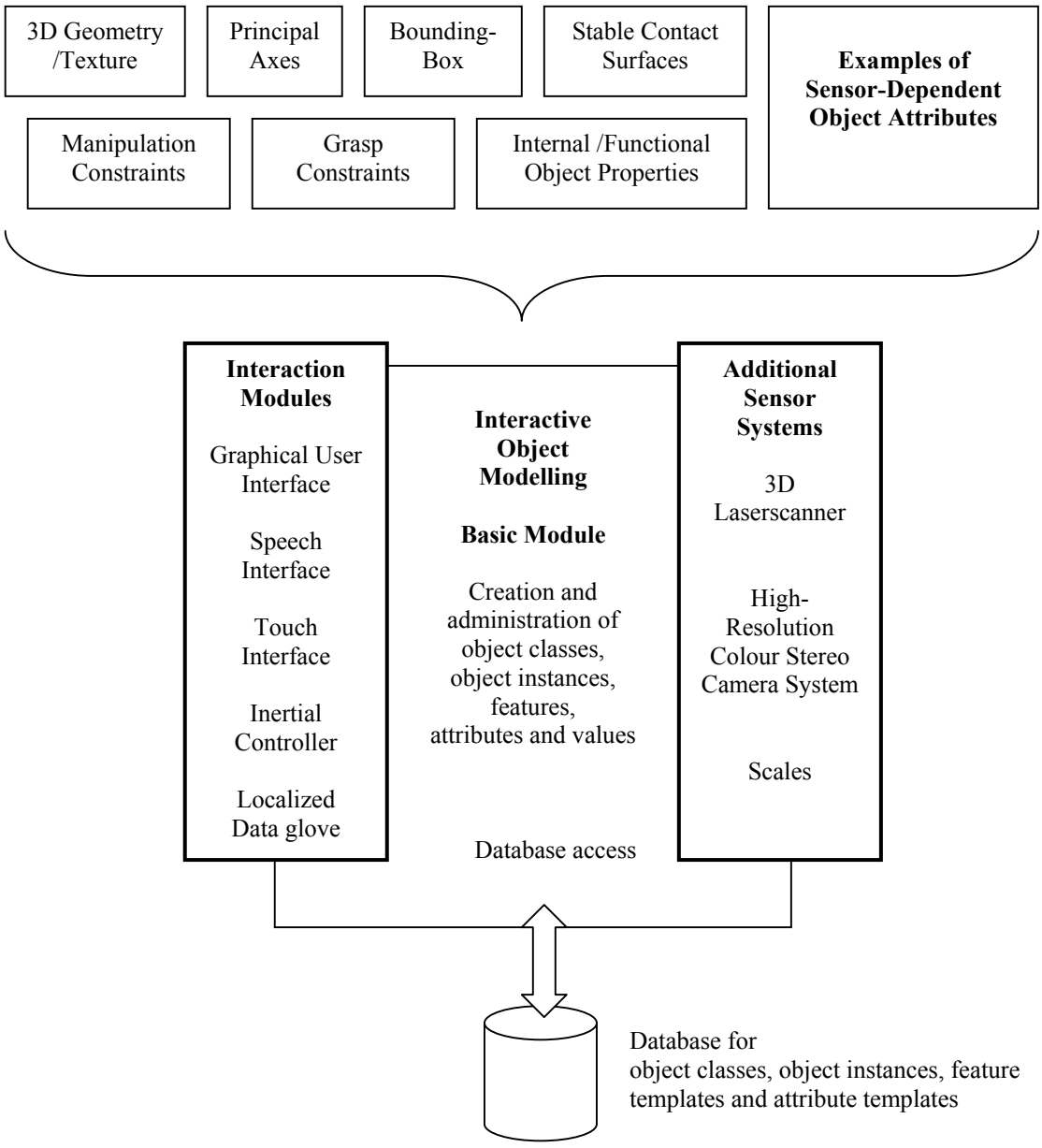
The remainder of this paper is structured as follows: First, we give an overview of the structure and components of our object modelling system. Then, the object representation developed in the course of this work is presented in detail, together with information about the basic software module for object modelling. The use of sensor systems for object learning is discussed, and finally the interaction with the user that is necessary to gather meaningful object models. The paper is concluded by a short summary and an outlook onto current and future work.

System Overview

The aim of this work is to design and implement an interactive object modelling system which can e.g. be used for a service robot. This involves both designing a useful object representation and designing and setting up the actual interactive modelling system. Both takes place with regard to the specific needs of a service robot interacting with a non-expert user, e.g. in a household, but could also be used in other settings or for other machine systems building up an object representation. The interactive object modelling system consists of five components, as shown in fig. 1: a basic modelling component which contains an implementation of the object representation we propose and grants access to the second component, a database of world knowledge which also contains information about objects. To build up the object

knowledge, modules for both interaction with the user and sensor systems to collect the relevant data are needed. Finally, modules are included to determine the relevant object attributes depending on sensory information. Some examples for such sensor-dependent object attributes are shown at the top of figure 1.

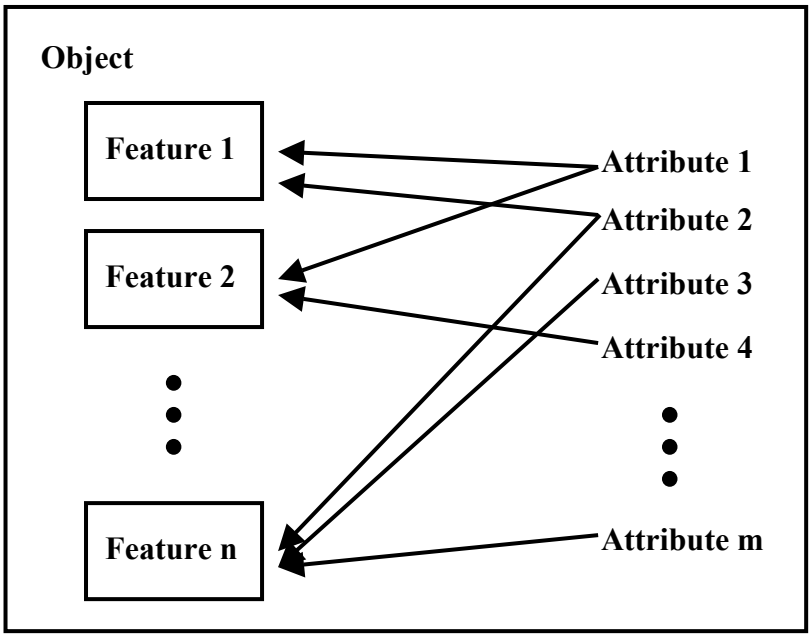
Figure 1: Structure of the interactive object modelling system



Object Representation – Design and Implementation

According to the object representation developed in the course of this work, an object model consists of features, attributes and attribute values. An object can have different features, which in turn each imply several attributes with corresponding attribute values. The basic structure of the object model is shown in fig. 2.

Figure 2:Basic structure of the object model



Features carry information on a relatively high semantic level. Examples for such features are "is a container" or "is transportable". Objects that have such a feature belong to a category of objects which has a semantic meaning for humans (e.g. the category of all containers). At the same time, features allow to associate objects with actions. E.g., an object with the feature "is transportable" can be transported, a container can be filled or emptied etc. Thus, the concept of features in this sense has two main advantages: on the one hand, it entails a maximally easy communication and interaction between the human user and the system, as both are thinking about objects in the same categories. On the other hand, task instructions of the user can easily be mapped onto the corresponding object categories.

For a robot system, the information based on the features alone is not detailed enough to work with. The robot would lack information which is necessary for the processing of sensor information, like colour information for object recognition, or information for the actor systems, e.g. movement restrictions like the maximum acceleration when carrying a full glass. A second important part of the object representation we developed are thus *attributes* and their *attribute values*. Attributes describe all of this necessary information on a lower level than features, and each feature implies the existence of several attributes. E.g., an object being a container has a fill level, a maximum acceleration and tilt angle depending on the fill state etc, attributes which do not necessarily make sense for other objects not being a container. I.e. once the features of an object are known, its relevant attributes are also given. It is important to note, though, that an attribute does not exclusively belong to one feature. Rather, an attribute can be attached to several features. E.g. the maximum acceleration which is allowed for an object can depend on different object features (its fill state, transport restrictions etc). In such cases, an attribute is associated with as many features as necessary. When its value for one of the features changes, the updated attribute value is set globally.

For each attribute, the type of attribute value is chosen in advance from a predefined set of attribute value types, such as 3D vector, trajectory, HLS colour information etc. An attribute's value can depend on the values of other attributes. This is typically the case with attributes which depend on information from sensor perception, e.g. a 3D model of an object is necessary to calculate its stable surfaces for putting the object down. These dependencies between attributes are used when an attribute is inserted into a feature or an object. In this case, the other attributes it depends on are inserted automatically as well.

Objects are grouped into *classes*, and a concrete object is an *instance* of such a class. Classes inherit from other classes hierarchically, i.e. they also inherit information about relevant features, default attribute values etc. An example would be the class of white cups from a specific series of dishes, all of which are white, have a certain height, weight, diameter etc. This class inherits from the more general class of cups, and therefore every instance of this class of white cups is a container, since every cup is a container. Several concrete instances of this class might exist in the robot's world, some of which are located in the scene: two on the shelf, one in the dishwasher etc. An important point is that features, attributes

and attribute values are inherited among object classes, and that an object instance only has the features and attributes that belong to the corresponding class, although the instance's attribute values might differ from the default values of the class. Information like the position of an object which is only relevant for objects occurring in a scene is not treated as an attribute, but is only allocated to object instances which are part of a concrete scene.

The basic module of the interactive object modelling system contains an implementation of this object representation and of functions to create and enhance objects, features and attributes. Additionally, it offers access to a database system which holds the relevant world and object knowledge of the robot system. In this database system, information about objects which exist in the robot's world, their features and attributes is stored amongst other things and can be retrieved by various components of the robot system (Becher 2003).

In the basic object modelling component, attributes, features and object classes and instances are called *descriptors*. They can be set up interactively by the user (cf. section about user interaction below) and then be stored in the world knowledge database. When a new attribute is created by the user, it is at first represented by an attribute template. Such an attribute template consists mainly of the name of an attribute, the type of its attribute value and the default value set by the user, as explained above. A newly created feature is called feature template, correspondingly. When an attribute is inserted into a feature, actually an instance of the template is generated and inserted. This instantiated descriptor is then no longer connected to the original template, i.e. any changes to the template have no effect on the instantiated descriptor. The same holds true for features that are inserted into object classes: an instance of the feature template is inserted in this case. The distinction between template and instantiated descriptors allows the user to create as many useful template attributes and features as necessary, and change them later as required, without unexpected side-effects on existing object information. On the other hand, the use of templates simplifies the task of the user considerably for complex domains.

Object information which is to be stored in the world knowledge database is represented as attribute-value-pairs in an XML syntax. A generic example is shown in fig. 3. The database itself is a MySQL database, and Xerces is used as XML parser. The interface of the database is realised in Corba to grant easy platform- and programming language-independent access on the robot's world knowledge.

Figure 3: XML representation of an object

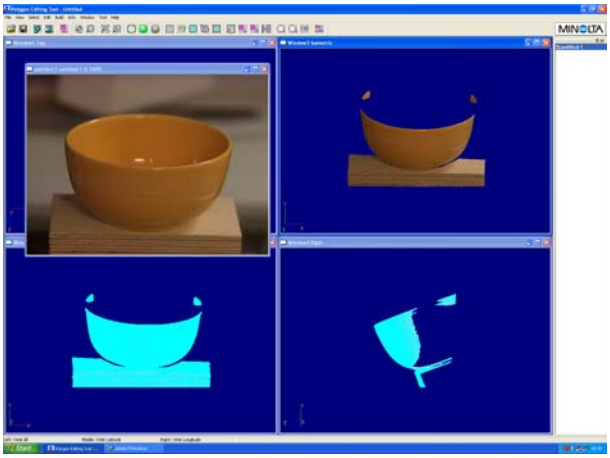
```
<objectclass>
  <name>cup</name>
  <features>
    <feature>
      <name>is_container</name>
      <attributes>...</attributes>
    </feature>
    ...
  </features>
  <attributes>
    <attribute>
      <name>fill level</name>
      <valuetype>percentage</valuetype>
      <value>0</value>
      ...
    </attribute>
    ...
  </attributes>
  ...
</objectclass>
```

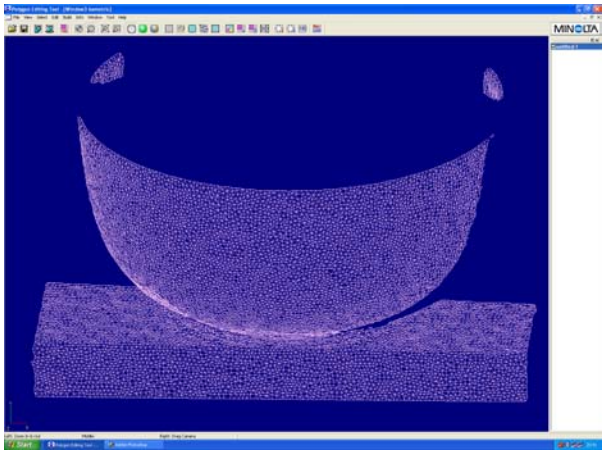
Sensor Systems for Object Learning and Sensor-Dependent Object Attributes

The upper part of figure 1 shows some examples of object information which is important for manipulation and other tasks: 3D geometry, manipulation constraints, etc. Some part of this information can be infered autonomously from sensor data, but for other parts, interaction with the human user is the easiest and most reliable way of acquiring the relevant information. To enhance the quality of the resulting object models, we set up an object modelling area equipped with specific sensor systems. However, this does not mean that object modelling cannot take place on a robot directly, but simply that modelling objects in the modelling area will give more precise results because of the controlled environment and the quality of the sensor systems. The sensor systems in this modelling area are a 3D laser scanner and a high-resolution colour stereo camera system. The brightness and direction of the lighting can be varied via a software module. The modelling area is shown in fig. 4. For the future, we plan to integrate additional sensor systems like scales to gather weight information more precisely. It has to be noted, though, that the system in itself is not dependent on exactly the chosen types of sensors, but would also work with any other kinds of sensors, provided that the desired information about objects can be acquired. Fig. 5 presents exemplary data collected with these sensor systems to show the quality of the results.

Figure 4: Object Modelling Area with Sensor Systems

Figure 5: Example data gathered with sensor systems in object modelling area: a) original object (upper left), texture (upper right) and 3D point cloud (lower left and right); b) result of triangulation





The extent to which user interaction is necessary in this step depends very much on the specific sensor-dependent object attributes. Some attributes can be determined automatically from sensor data and only be presented to the user for correction if problems should occur. This includes information like a colour histogram or a 3D point cloud from a single view. When information based on the complete surface of an object is necessary, and not only from one specific angle, we propose a method to use the users knowledge for fast and successful processing. To this means, objects are placed on a rotary plate such that they can be presented to the sensor systems in arbitrary angles, and the user decides about the views which are to be scanned next (the method is described in more detail below when we discuss our concept of including information given by the user). Other information like functional object properties cannot be determined without the help of the user (e.g. how to use a microwave to heat up some liquid) or is much more easily and robustly gained in interaction with the user than it could be autonomously.

Interaction with the User

For the object representation suggested in this work, interactive object modelling means to learn new objects as well as new attributes and new features of objects. In our implementation, interaction between user and system is realised as follows: Object models are built up in the object modelling area described above which guarantees a high quality of sensor data. The method itself is naturally not dependent on this modelling area, but can also be used on a robot system itself during its daily work. Interaction with the user takes place using several different modalities, namely a graphical user interface which can also be projected onto a touch interface, a joystick bar with inertial controller, and a speech processing component will be integrated in the near future. A data glove in combination with a localization system for the human hand will be integrated in the near future. The user can then use the data glove to provide information like possible grasp points, grasp forces, or maximum acceleration when an object is transported in a very intuitive and comfortable way. An exemplary use of the data glove can be seen in fig. 4.

The graphical user interface presents all relevant information to the user and at the same time offers him the opportunity to build up object models by specifying an object's features, attributes and attribute values. The graphical interface can be projected onto a touch-sensitive display, offering the opportunity to work independently of mouse and computer screen and thus reducing the psychological barriers for users which are not used to working with the computer. Additionally, the speech recognition component allows the user to enter a set of commands depending on the module of the graphical interface he is currently working on. The speech commands are processed by the system as the corresponding mouse clicks (or presses of a finger, respectively) are.

To gather as complete and as correct an object model as possible, sensory information is combined with information interactively provided by the user. E.g., a laser scanner allows to build up a 3D point cloud of an object's surface which can then be used to determine a surface model, bounding box etc. It is sometimes suggested that these computations should be done autonomously by the robot system from sensor data. We think, however, that this is not feasible for some attributes with passable effort. To gather a complete object model, the system must be able to get sensor information from all points of the object surface, where holes in the model have to be detected and filled up by new measurements. This is usually a very tedious and time-consuming process. In our system, the acquired data is presented to the user who can then decide whether more sensor information is necessary or not. At the same time, the user is able to detect problems or errors of the system which can then be corrected. The user can also point the system to parts of the object which have been occluded to the sensor systems so far. To this means, we set up a pointing device consisting of an Inertia Cube integrated into a standard joystick bar. The Inertia Cube collects the data of the human hand which holds the joystick bar. With this device, the user can rotate 3D models displayed on the screen. He can, for example, turn the model as far as necessary to point the system to an angle where more sensor information should be collected. The angle chosen can be confirmed with the buttons on the joystick bar. The rotary plate on which the object lies is then directed onto the chosen angle, and at the same time the information can be presented with the new angle in the graphical user interface.

In this way, we combine the strengths of both automatic and interactive object modelling: The system determines object information automatically as far as possible or sensible in terms of processing time and quality of the results. The user then controls the output of the system and adds as many completing information as he or she finds necessary and helpful. The result of the combined process is represented within the object model presented above. In particular, the creation of attribute templates and feature templates can hardly be done by the robot system. Once the user has set up the feature templates which are relevant for a certain application area, he can easily build up new objects by simply choosing the appropriate templates for each class of objects. The system can then fill in attribute values automatically or with the help of the user, respectively.

Conclusions

In this paper, we proposed a system for modelling objects in interaction with a human user. We suggested an object representation which is designed for the purposes of a human-centered dynamic environment, where models of new objects have to be added and a sensible communication with humans has to take place. We then discussed our implementation of an interactive object modelling system which is based on this representation as well as modules for interaction, sensor systems and sensor-dependent object attributes. The achievement of this work is twofold: first, the object representation itself, and secondly the insight that interaction with the user is indispensable for an object modelling system e.g. of a service robot. The functioning of our approach is shown by the implementation of the system in the interactive modelling area.

In the future, more sensor systems and different means of interaction with the user will be included into the modelling area, and the (semi-)automatic detection of object features from sensor data will be further expanded.

Acknowledgements

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center 588 “Humanoid Robots – Learning and Cooperating Multimodal Robots” and by the German Ministry of Education and Research BMBF within the German Service Robotics Initiative DESIRE.

References

R. Becher, P. Steinhaus, R. Dillmann: Interactive Object Modelling for a Humanoid Service Robot. In: Proceedings of the Third IEEE International Conference on Humanoid Robots, Karlsruhe, Germany, 2003.
A.T. Miller, S. Knoop, H.I. Christensen, P.K. Allen: Automatic Grasp Planning Using Shape Primitives. In: Proceedings of the International Conference on Robotics and Automation, p. 1824-1829, Taipei, 2003.
M. Pollefeys, M.J. Van Gool: From Images to 3D Models. In: Communications of the ACM, 40(7), p. 50-55, July 2002.