

The Concept of Object and its Relation to Human Thinking: Some Misunderstandings Concerning the Connection between Object-Oriented and Human Thinking

Sten CARLSSON, Benneth CHRISTIANSSON

Division for Information Technology, Karlstad University

S-651 88 Karlstad, SWEDEN

e-mail: Sten.Carlsson@kau.se

Benneth.Christiansson@kau.se

Received: January 1999

Abstract. All throughout the computer science community object-orientation is accepted as being built upon the same basic concepts that human beings use to apprehend reality. This misconception, as we think, is what we focus on in this paper. To show this we analyse two well-recognised object-oriented systems development methods. We try to pinpoint in what way these methods do not correspond to the way human beings apprehend reality in terms of objects. We show that the methods do not use the concepts of object or class in a manner that corresponds to the way human beings use them to apprehend reality. Furthermore the method-creators do not separate the notion of an object and its representation in a model. We also suggest a better adaptation of the searching-for-objects model based on how human beings apprehend reality. When analysing, one should focus on the purpose of the actions and the two different modes with which an object can be treated: present-at-hand and readiness-to-hand. This will increase the conformity between object-orientation and the way human beings apprehend reality.

Key words: object-orientation, systems development, human-thinking.

1. Introduction

In this paper we focus on object-oriented methods concerning the development of computer based information systems (CIS). We have not let the aperture in our objective lens of analysis be wide open towards all the crucial issues in developing systems used to inform human beings by help of computers. Our aperture is delimited to one important aim of a method namely the support of design methods to what concerns human thinking in that development. According to the Greek meaning of the concept of method, to choose a way, we have chosen to make a brief analysis of some assumptions about the notion of object and human thinking in two object-oriented methods: Object-Oriented Software Engineering (Jacobson *et al.*, 1992) and Object-Oriented Analysis (Coad and Yourdon, 1991).

It is often insisted that when using object-orientation one is using man's natural way of comprehending the real world in terms of objects. "*People regard their environment*

in terms of objects. Therefore it is simple to think in the same way when it comes to designing a model" (Jacobson *et al.*, 1992, p. 42). "*OOA-Object-Oriented Analysis – is based upon concepts that we first learned in kindergarten: objects and attributes, wholes and parts, classes and members*"(Coad and Yourdon, 1991, p. 1).

By analysing some assertions of this commonly presented idea of perfection concerning object-oriented methods contra other approaches we want to reveal some incomplete (defective) assumptions of experiencing objects according to these assertions. This quotation from Schutz (1953) might be taken as a support for the assertions above. "*For example, the outer world is not experienced as an arrangement of individual unique objects, dispersed in space and time, but as "mountains", "trees", "animals", "fellow-men"*"(ibid, p. 5). But, by a closer and a deeper study of Schutz' explanations of how we are concerned with the objects, we soon may come to a conclusion that some of the ideas presented might be questioned concerning the methods we discuss. That does not mean that we reject the whole idea of object-orientation in development of CIS or that a method should match human thinking. But we do not accept that such a profitable and important factor of a method is so weakly grounded in terms of theoretical and philosophical arguments about human thinking and objects.

The aim of this paper therefore is:

- to point out some of the most startling differences between how the concept of object is related to human thinking in these methods compared with some theoretical and philosophical texts about human thinking in everyday;
- to make some suggestions about how we might obtain a better adaptation of searching for objects in systems development according to human thinking about objects in everyday life.

Our path to present this brief discussion and our suggestions is as follows:

Chapter 2. The concept of object in the view of everyday life.

Chapter 3. Searching for objects in the sense of thinking.

Chapter 4. Representation of objects in models.

Chapter 5. Some conclusions and theoretical suggestions of a better adaptation.

2. The Concept of Object in the View of Everyday Life

By starting to discuss the concept of object in the perspective of everyday life we emphasise the very fact that this is the most natural way to do it concerning object-oriented methods. The reason however is not just the object-oriented notions that states that people regard their environment in terms of objects. Instead, as all kind of scientific analysis are founded upon the thought objects of common sense in everyday life (Schutz, 1953), consequently the object-oriented development of CIS as such is founded upon that common sense. Therefore the everyday life is a priori of any human construction of methods or concepts in these methods. The content of the method is constituted by the stock of previous knowledge at hand of the constructor when interpreting the everyday life according to some personal worldview of that life (ibid).

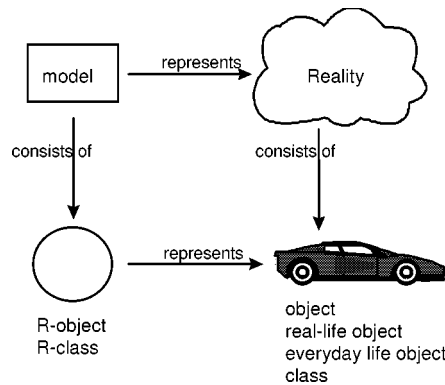


Fig. 1. The used concepts and their interconnections.

The question therefore is how the concept of object is defined in the studied methods and how this definition stands according to the common view of objects in everyday life.

To avoid any confusion concerning our meaning of the conception of object we use the notion “object” concerning real-life objects and the notion “representation of objects” (*R*-objects) when using the concept in models and methods. The different concepts and their interconnections are described in the Fig. 1.

2.1. The Concept of Object and Objectory

One aspect of the definition of *R*-object in Objectory we want to challenge is that Jacobson *et al.* (1992) have not made a clear cut definition according to the common use of the concept of object in everyday life. They assert: “*The first and most important concept that we describe is, of course, the concept of object. What is an object? The word “object” is misused and used in nearly all contexts. What we mean by an object is an entity able to save a state (information) and which offers a number of operations (behaviour) to either examine or affect this state. An object is characterized by a number of operations and a state which remembers the effect of these operations*”(ibid, p. 44). By not explaining what Objectory *R*-object really means in comparison with what an object really is about in everyday life they are misusing the concept themselves. Their definition is a confusion of an everyday life object and a representation of such an object (*R*-object). But the problem with this confusion is that their definition neither corresponds to the definition of real-life objects nor to the definition of a representation. Why does their definition not correspond to the definition of real-life objects? Concerning, for instance, the relations between the objects they assert: “*Objects usually correspond to real-life entity objects, such as an invoice, a car or a mobile telephone*”. (Ibid, p. 44). As the authors’ aim is to make a model of reality, the content of such a model always corresponds to the reality¹ we want to represent in any model, and so do

¹According to Schutz and Luckman (1974) reality means the reality which the human being confronts in natural attitude in everyday life.

the *R*-objects. If *R*-objects just usually correspond to real-life objects, in what world do these objects exist which do not? We think this confusion is due to the fact that they are stuck to the idea that the object-oriented way of representing objects in models is a new foundation of the conception of object. But, it is not. As the description of the operations of a software system is a model of something (Cook and Daniels, 1994), objects represented in this model are not objects themselves. The representation of an object instead is an objectification of a real-life object. Accordingly their definition of the concept above is just a description of how to represent objects in models. That is why their definition of the concept of an object does not correspond to that of real-life objects. But why does their definition of an object not correspond to that of a representation?

What still is confusing is that this definition is not applicable to all types of representation. The part of the definition of the quotation (“What we mean . . . or affect this state . . . these operations.”) is applicable to just an everyday life object such as an artefact with functional capacities of representation according to these functions of saving and so on.

By making a model of everyday life objects and by representing these objects on a piece of paper according to these forms of notation means that we also represent everyday life objects in a model. In this case we represent the objects in an object-oriented manner on a piece of paper. But this contradicts their definition of the concept because we cannot attach behaviour in a representation of something on a piece of paper. We can just make a description of that behaviour.

2.2. *The Concept of Object and OOA*

Coad and Yourdon (1991) define an object as: “*An abstraction of something in a problem domain, reflecting the capabilities of a system to keep information about it, interact with it, or both; an encapsulation of Attribute values and their exclusive Services*” (p. 53).

When we are talking about an abstraction of something it is very important to remember that an abstraction can be either a real-life object or an objectification of an object. In fact some abstractions are real-life objects themselves, such as an organisation. This implies that it is possible to abstract an abstraction when using OOA. It is very important to stress this difference.

When dealing with the object model (the *R*-objects) it is therefore crucial that we realise firstly, the difference between the object and the *R*-object, and secondly, the difference between an object as an abstraction and the *R*-object as an abstraction. In Coad and Yourdon’s method these differences are not explicit or obvious, maybe not even regarded. These might be two big contributors to the confusion that exists when speaking of objects in object-oriented systems development.

Another reason for confusion in the method is that they define one symbol that represents an object and the Attributes and Services it holds and the same symbol also represents a Class which is: “*A description of one or more Objects with a uniform set of Attributes and Services, including a description of how to create new Objects in the Class*” (ibid, p. 53). But we cannot create a new *R*-object by a model or a *R*-class. Using the concept of a class to describe one or more objects seems to be quite understandable. We

use the class to collect the descriptions of similar attributes and services. But how does the idea of creating new *R*-objects out of an *R*-class relate to the notion that an *R*-object is an abstraction of something in a problem domain? This behaviour is not an abstraction of a real-life object it is something utterly artificial. Instead we assert that we do not create real-life objects from classes. Classes are created in our mind based on existing objects and their characteristics (Martin and Odell, 1995). *“It is important above all to stress that the orders of reality do not become constituted through the ontological structure of their Objects, but rather through the meaning of our experience”* (Schutz and Luckman, 1974, p. 23). That means we cannot constitute *R*-objects through any ontological structure of an *R*-class. To illustrate this problem, imagine a building as an object. A building consists of rooms. This particular building happens to be a school. We cannot just by viewing the building and its rooms, conclude that it is a school. A school exists in a building so by that notion we have conformity. The meaning of the real-life object school is however what takes place in the building, the actions and social constructs inside the building and its rooms. We could use an *R*-class of a building to create an *R*-object of a school but then we would not be able to see the difference between any *R*-object of a building and an *R*-object of a school. This we cannot extract through the ontological structure of the *R*-class. We have to focus the assigned meaning of the building to be able to identify what type of real-life object it is, for instance a school.

3. Searching for the Objects in the Sense of Thinking

In this chapter we will discuss how human thinking influences the search for objects in CIS development. The focus is how the intentions of this searching in the discussed methods are adjusted to human thinking of objects in everyday life. Firstly, we will present some basic theoretical assumptions about this thinking. Secondly, in Part 2 and 3 we explain our critical standpoints concerning the methods discussed.

3.1. Some Basic Outlines about Thinking of Objects

How does the human being think of objects? Firstly we will emphasise that this thinking is not ad hoc in any sense. We just do not think of the objects as such. This thinking is always tied to some action or as Winograd and Flores (1986) put it: *“In sum, Heidegger insists that it is meaningless to talk about the existence of objects and their properties in the absence of concerned activity, . . .”* (p. 37).

Being tied to the action is also important for the interpretation of the meaning of the object because, according to Schutz (1953): *“... we cannot understand a cultural object without referring it to the human activity from which it originates”* (p. 7). When acting, our thinking is just concerned with those objects that are related to the action. All other objects that we also know about fall into the background in our memory (ibid). That means what particular objects we think of and what characteristics of the objects we identify depends on the aim of the action. *“It is this purpose at hand which defines those elements among all the others contained in such a situation which are relevant for*

this purpose”(ibid, p. 6). Consequently it is the aim of the action which constitutes what object we think of but also what typical characteristics of the object we take into account in our thinking. What characteristics of the object we select in our thinking according to a current purpose of an action might therefore be quite different if we perform another action with the same object but with a different purpose (ibid).

Let us explain this by an example. Assume that you think of a car, just a car as an object. Then you perhaps think of its colour, its wheels and so on. If you are able to limit your thoughts just to the car as an object nothing else will come into your mind than its parts. But as soon as you think of some action with the car you will also think of other objects. The objects you think of depend on the action. If you have got a puncture and you are going to fix it, objects relevant to this action will come into your mind. Planning a trip with the car you perhaps instead think of a map or if you have got enough petrol for that trip. Therefore our searching for the objects should be built upon what characteristics we assign these objects, constituted through the purpose of the action and through our present understanding of the objects. This constitution however, should be seen as momentary. When acting, the human thinking treats the purpose, the action, what objects and what characteristics we think of as an intrinsic whole. Consequently we cannot search for the objects and the selected particular characteristics of the object in development of CIS without at the same time searching for the actions and their purpose.

3.2. *Searching for Objects by Objectory*

Searching for objects according to Objectory is based on use cases, which is governed by the requirement model. This model consists of three parts: “*the use case model, the problem object model and user interface descriptions*” (p. 157). As the use cases are identified through the users of the system and the use case model means a specification of the functionality of the system we mean that this is some sort of an action the system will fulfil according to the user. In that sense the actions are in focus when searching for objects by Objectory. It is for instance recommended to ask (ibid, p. 161): “What are the main tasks^(sic) of the actor?” The problem is however that it is just the action as such which should be described in Objectory. But it is the purpose at hand of the action that defines those objects that are relevant for the aim of the action and consequently the objects and the characteristics of the objects (attributes and behaviour) involved in this action. The purpose of the action is however not mentioned as an important factor in searching for the objects and the attributes by Objectory. We mean that this is a problem in the method according to human thinking of objects. Neither do they put focus on this in the user interface description or in the discussion about the problem object model. Concerning the problem object model it is asserted that this model will help developing an Objectory-noun list. “*In this way we will have a glossary that can be used to formulate the functionality of the use cases*” (ibid, p. 167). But nothing is mentioned concerning the purpose of these nouns according to the analysis of that functionality. The relations between the objects and the attributes when searching for the objects are also weakly notified in their description of the requirement model. “... *a common base for developing*

the requirements model, we believe that the object name and possibly also the logical attributes ... are an appropriate level to stay at" (ibid, p. 168). First of all we mean that there are no logical attributes related to the objects. Which attributes we relate to the objects depend on the aim of the action and how the human being thinks of the aim, the action and the objects as an intrinsic whole. Secondly instead of saying "possibly" the attributes "always" should be taken into account, according to Schutz (1953), when analysing the relationship between the actions and the objects. Because it is just in that moment when we are talking about the action and the object concerning this action that the specific characteristics (attribute and behaviour) of the object concerning the action and its purpose will come into our mind.

3.3. Searching for Objects by OOA

In order to use OOA as a method for analysing a real-life situation you are supposed to perform five major tasks:

- Finding Class and Objects
- Identifying Structures
- Identifying Subjects
- Defining Attributes
- Defining Services

In the method Coad and Yourdon (1991) stress that these are five activities and not sequential steps. They also stress that you should consider these activities as overlapping layers and furthermore that attributes and behaviour of the *R*-objects must be considered as an intrinsic whole. But there are no clear-cut recommendations of how to handle this overlapping or this intrinsic whole concerning actions and purposes. According to (ibid, 1991) they claim that their method: "...organizes analysis and specification using the methods of organization which pervade people's thinking" (ibid., p. 3). They do not explain how to use OOA according to how the objects are constituted by human thinking in relation to actions and purposes.

Another problem is, when focusing on the first of the five major activities in OOA; "Finding Class and Objects" we assert that OOA is not built upon a well thought out theoretical ground of how people apprehend the objects in real-life. This can be motivated as follows. Coad and Yourdon's recommendations of "where to look" for the objects, that is to say how to search for the objects, just consist of some common rules of thumb such as observe, listen, check, and read. These recommendations do not give any hints about how to observe, listen etc., according to how people really think of the objects. Therefore we assert that a more well thought out theoretical connection between how to search for the objects and the human way of thinking might result in a more stable way of searching for these objects than the more or less experience-based recommendations by Coad and Yourdon (1991), see Chapter 5.2.

A third problem is that the recommendations of "what to look" for, the objects, just also contains an enumeration of typical system aspects such as structures, other systems,

devices etc, by which the objects should be derived. Neither does this enumeration of aspects build upon a well thought out theory of how the human being really understands the objects in everyday life. Instead it is through the help of a subject and its purpose with an action that we are able to explore the need of an object, and furthermore how these objects are experienced in the context of systems development of CIS. What to look for is namely just one side of the coin of understanding objects. The notion “what” is, according to pedagogical theory, just the reference to the objects (Kroksmark, 1987). How these objects are understood depends on how the human being discerns them. So the meaning of “what” depends on “how” the objects are discerned (*ibid*). This understanding might also vary as our knowledge is a construction in our mind (Schutz, 1953) and in the case of “how” depends on our pre-understanding of the objects.

4. Representation of Objects in Models

In this chapter we are going to discuss some important issues about representation of objects and human thinking.

In everyday life, according to Heidegger (1981), the human being treats the objects in two different modes; present-at-hand and readiness-to-hand (Winograd and Flores, 1986). The first mode means, if you hold an object in your hand, you can tell what the object looks like. For instance, assume that this object is a ball pen. When you look at the ball pen you can realise its colour, the shape and so on. But when you use it you never think of what the ball pen looks like. When using the ball pen, you are only thinking of the action of writing as such. In that case the ball pen is used in a readiness-to-hand mode. The character of the ball pen (present-at-hand) falls into the background of your mind, you do not think about it at all (*ibid*). The character of the ball pen will only come into your mind again in a present-at-hand mode if there is a break down in your writing. For instance, the ink has run out. At that moment the ball pen will be in a present-at-hand mode in your mind again (*ibid*). You look at it in this mode and think what is wrong with it.

There are some very important differences between present-at-hand mode and readiness-to-hand mode concerning understanding of an object in relation to systems development of CIS and representation.

Firstly, when you look at an object (present-at-hand mode) and never have seen such an object before, you are not able, by help of what it looks like, to fully understand how to use it. Furthermore you do not even understand why the object has got its characteristics according to how it should be used. This you just can fully understand when using the object in a readiness-to-hand mode. Accordingly, this means that you cannot understand how to design an object without understanding how the object will be used in a readiness-to-hand mode.

Concerning the representation of objects we therefore have to think of which of these modes we use when representing objects in our models. Assume that the designer knows how an object is functioning in a readiness-to-hand mode and then the designer just represents the object in present-at-hand mode, that is to say what it looks like. If the user has

no experience of using this object in a readiness-to-hand mode it must be very difficult for him or her to understand how it is used by help of a model that just presents the object in a present-at-hand mode. Assume furthermore that neither the designer nor the user have any experience of how the object they are designing is used in a readiness-to-hand mode they will never be fully aware of this by representing the objects in models in a present-at-hand mode (Winograd and Flores, 1986). By this we do not mean that it is not important to represent objects present-at-hand. What we want to stress is the limitation of what we are able to interpret from a model according to these different modes, see Chapter 5.3.

4.1. Representation of Objects in Objectory

When representing objects by models in Objectory you use for instance a problem domain model. This model is used to define the task of the system according to a use case and as Jacobson *et al.* (1992) assert: “Such a problem domain model will be a strong support also when specifying the use cases The major benefit of such a model, though, is that it is a very good tool with which to communicate about the system” (p. 162). The representation of objects in this model is made by small circles that are bound together with arrows according to the Fig. 2.

This is a typical example of representing something in a present-at-hand mode. The Fig. 2 just shows “present-at-hand” how the represented objects are related and nothing about how the recycling machine is functioning in a readiness-to-hand mode.

What we may say is that this model will perhaps support communication about which objects the model contains and their present-at-hand mode. But, a person who does not know in advance how a recycling machine works may have problems to understand how this machine works in a readiness-to-hand mode using a model that only represents objects in a present-at-hand mode. Heidegger (1981) asserts that it is impossible to fully understand the usefulness of an object just represented in a present-at-hand mode. Therefore a problem domain model does not contribute to the understanding and information about the usefulness concerning objects in their readiness-to-hand mode. There has to be another type of model to fully understand this.

They certainly assert that prototyping by help of a computer might be useful during the development of CIS. This is of course one way of showing how something is used in

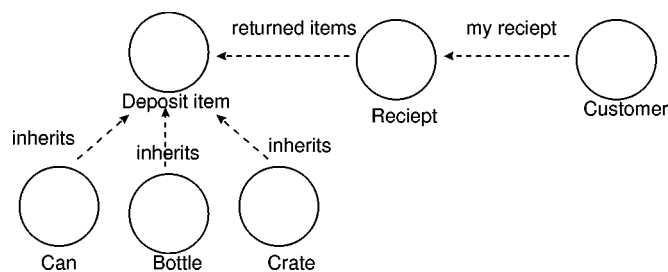


Fig. 2. A problem domain model of the recycling machine.

a readiness-to-hand mode. But they never talk about this according to the fact that their models just show the work of systems development in a present-at-hand mode.

4.2. *The Representation of Objects in OOA*

In the method there exists a certain notation which is based upon the object and its class. It is represented with a rounded rectangle divided into three parts. The upper part contains the name of the class. The middle part contains the description of attributes of the *R*-objects created from the class and the lower part contains the representation of the services performed by the *R*-objects created from the class. The outer rounded rectangle represents the actual object.

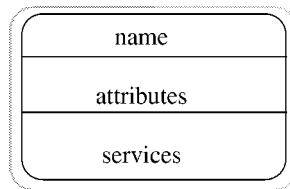


Fig. 3. Symbol that represents a *R*-class and an *R*-object in the OOA method.

The symbol in the Fig. 3 is used to represent the object and its relations to other *R*-objects in a strictly present-at-hand mode. This is the mode used throughout the analysis and is the only one emphasised in the notation. There does not exist any representation in OOA that visualises how *R*-objects are used in a readiness-to-hand mode.

The closest to a representation of an object corresponding to a readiness-to-hand mode there is in OOA, is in the “Defining Services” activity. In that activity you are supposed to examine the messages and connections belonging to the *R*-objects by executing different situations (scenarios). But this does not mean that we identify or represent the objects in their readiness-to-hand mode. To be able to understand this mode from the activity of “Defining Services” you have to understand how a computer executes a program. But, you get no help to test your ideas of the services and their execution by *R*-objects which you just have represented in a present-at-hand mode.

5. Some Conclusions and Theoretical Suggestions for a Better Adaptation

In this chapter we will make some brief suggestions about how we should speak about and use the concept of object in analysis and design of CIS in order to get a better adaptation of that concept according to human thinking.

5.1. *Definition of the Concept of Object*

When speaking about the definition of the concept of object in analysis and design and using it in texts about modelling we assert that it is of great importance to make a clear-

cut distinction between the concept of object and the concept of objectification. The real-life object is the object that the human being relates to itself when doing something in everyday life. The real-life object is always either an external thing or an abstraction, that is to say an object that we just can get a grip on by thought, such as a policy, marketing, organisation, system and so on. An objectification on the contrary is a representation of an object in a model on a piece of paper or in a computer.

Coad and Yourdon (*ibid*) for instance use the concept of abstraction synonymously with the represented object in a model. This is of course quite right if you mean to make a category of an object in a model concerning something in everyday life. But this abstraction is not a real-life object. It is an objectification of something. As the concept of abstraction also might be used in the sense of a reference to a real-life object we have to be aware what we mean when we are talking about an abstraction. Do we mean an objectification or a real-life object? By not making any distinction between abstraction as a representation and abstraction as a real-life object, their definition of the concept of object will be confusing. The problem can be illustrated with an example from our education situation. During a course in object-oriented analysis, the students learn how to use the object-oriented principles and a certain method, for instance OOA (Coad and Yourdon, 1991). There is often a big problem for the students to understand and handle the difference between a real-life object and its representation when using the method. When describing a real-life object via an *R*-object in a model the students confuse the *R*-object and the real-life object it represents. This occurs because the difference between a representation of a real-life object and the real-life object itself is not emphasised at all in the method.

Jacobsson's *et al.* (*ibid*) definition corresponds neither to the definition of a real-life object nor to the definition of a representation. Their definition of that concept just corresponds to representation of an object in a computer. This definition however cannot be used when talking about representation on a piece of paper as this definition implies mechanical devices for saving and so on.

5.2. Searching for Objects

When we are searching for the objects during analysis or design we have to be aware of how the human being is thinking of the objects when using them. This thinking is constituted by the action and its aim in that sense that it is the meaning of the action which determines what objects and what characteristics we identify in the objects. This thinking is also momentary, that is to say the action, its aim, the objects and its characteristics are treated in our mind simultaneously as a wholeness. Therefore we assert that searching for objects and their characteristics might result in a more consistent model if we are taking the actions and their aim into account. Consequently we should not separate the searching for objects and their attributes by methods from what current action the actor has in focus and the aim of that action.

A more stable way of searching for the objects and the attributes than just observe, listen, check and read as Coad and Yourdon (*ibid*) suggest, might be as follows.

To find a more stable way of searching for objects we have to know how knowledge is treated in the mind according to the meaning of something in relation to different subjects. Schutz and Luckman (1974) assert that the meaning of something is gathered in our memory in different stratas (finite province of meaning). Everything that a person knows about, for instance, invoicing is gathered in a specific province of meaning. Other routines, which are not compatible with invoicing, are situated in other stratas, and these stratas fall into the background in the memory when talking about invoicing.

One of the problems with searching for objects is selecting a strategy, one which enables the actors to know that they are talking about the same thing. The actors have to know that they focus on the same meaning strata (Theman, 1983), otherwise it is easy to misunderstand one another (*ibid*). To have a chance to focus on the same strata, Theman (*ibid*) recommends pointing out something with a well-known name, for instance a routine name. Then we should stick to that routine (action) in the discussions so everything of importance for the development according to that meaning strata is extracted (the aim of the routine, what objects, attributes and behaviour it contains). If we ask for objects, attributes and processes and take the idea of meaning strata into account we think that such a strategy will be more successful and stable, see Chapter 5.4.

In Objectory the action is focused on by the use cases but the aim of the actions is not taken into account in the models neither in analysis nor in design. Furthermore there is no clear-cut advice about how the characteristics (attributes) of the objects are related to objects, actions and purpose.

In OOA the searching for objects is still more separated from that of the action and that of the aim of that action in comparison with Objectory. According to OOA Coad and Yourdon (1991) recommend to look for objects, relations, attributes, and services in five major separated tasks. The recommendations seem more to be some rules of thumb than a firmer instruction-set of how to search for the objects. Is it possible to give such a set of instruction tips? There is of course not any formal method for that searching. But by analysing this as wholeness concerning the actor, action, aim, the objects and the attributes we think will be more successful than doing it stepwise.

Heidegger (1981) asserts that we have a language for our understanding of an object's readiness-to-hand mode. This is however depending on that we already know how to use the object. You can get an idea of where the usefulness of the objects might be analysed and designed by asking where, how, by whom and for what the objects are readiness-to-hand.

For instance, assume that you are going to design the content and the view of a form together with a user. We also assume that this will be done by prototyping using a computer. When you discuss this form pictured on the screen you have to realise that the readiness-at-hand of this form cannot be fully analysed just in relation to this user if anyone else also is going to use the form. Concerning every field in that form you have to analyse to whom will this field be readiness-to-hand for what, where and how. It is by asking this you are able to get a full understanding of where you can get information of the readiness-to-hand of the form. By testing it there you will see if your suggestions of the content of the form will fit readiness-to-hand according to the different users.

5.3. Representing the Objects

According to Heidegger (1981), we treat objects in two different modes; present-at-hand and readiness-to-hand. When using them the mode is readiness-to-hand and when there is a breakdown in using the object we think of it in a present-at-hand mode. Our recommendation therefore is that both of these modes have to be represented in our models.

For instance if we are going to design a register of the parts of the objects, we have to analyse the object present-at-hand. It is by this analysis we can get a grip on the parts of objects in our system. But when designing the usefulness of an object we also have to describe the readiness-to-hand mode of the object in our models. It is only in this mode that we can fully understand the usefulness. We are totally aware of the difficulties involved when describing an object in a readiness-to-hand mode. One possible way to do this is by using prototyping. In the future we may see computer based tools that enable the description of an object in a readiness-to-hand mode as well as a description of its present-at-hand mode.

Both in Objectory and OOA all the different models just show the representation of objects in a present-at-hand mode.

References

- Coad, P., Yourdon, E. (1991). *Object-Oriented Analysis*. Prentice-Hall, Inc.
- Cook, S., Daniels, J. (1994). Object-Oriented Methods and the Great Object Myth. *Objects in Europe*.
- Heidegger, M. (1981). *Being and Time*. Lund: Doxa (in Swedish).
- Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G. (1992). *Object-Oriented Software Engineering. A Use Case Driven Approach*. Addison-Wesley Publishing Company.
- Krokmark, T. (1987). *Fenomenografisk Didaktik*. Gothenburg Studies in Educational Sciences. Acta Universitatis Gothenburgensis (in Swedish).
- Martin, J., Odell, J. J. (1995). *Object Oriented Methods. A Foundation*. Englewood Cliffs, NJ: Prentice-Hall, Inc.
- Schutz, A. (1953). Common-sense and scientific interpretation of human action. *Philosophy and Phenomenological Research*, **14**(1), 1–37.
- Schutz, A., Luckman, T. (1974). *The Structures of the Life-World*. (Trans. by R.M. Zaner, H.T. Engelhart, Jr). London: Heinemann.
- Theman, J. (1983). *Uppfattningar av politisk makt*. Gothenburg Studies in Educational Sciences. Acta Universitatis Gothenburgensis (in Swedish).
- Winograd, T., Flores, F. (1986). *Understanding Computers and Cognition. A New Foundation for Design*. Norwood, NJ: Ablex Publishing Corporation.

B. Christiansson, lecturer in Information Systems, Karlstad University. In 1994 he started his doctoral studies at University of Linköping, Computer Science Dept. specialising in object-oriented methods for systems development and component-based systems development.

S. Carlsson, Senior Lecturer in Information Systems, Karlstad University. He has been a teacher in Information Systems since 1982. Research specialities are learning and communication in the systems development process.

Objekto samprata ir jos sąryšiai su žmogaus mąstymu. Kai kurie objektinės paradigmos ir žmogaus mąstymo prieštaravimai

Sten CARLSSON, Benneth CHRISTIANSSON

Informatikų bendrijoje priimta teigti, kad objektinė paradigma grindžiama tomis pačiomis sąvokomis, kurias žmonės vartoja tikrovei konceptualizuoti. Straipsnyje parodoma, kad iš tiesų šitaip nėra. Analizuojami objektiniai metodai ir išryškinami jų prieštaravimai žmogui įprastam tikrovės konceptualizavimo būdai. Parodyta, kad objektinėje paradigmoje objekto ir klasės sąvokų traktuotės gerokai skiriasi nuo kasdieniniame gyvenime įprastų traktuočių. Be to, daugelis autorių nedaro skirtumų tarp objekto sąvokos ir jos vaizdavimo modelyje. Straipsnyje pasiūlyta, kaip reikėtų keisti objektinę paradigmą, norint priartinti ją prie žmogui įprasto mąstymo būdo.