Worldmaking with objects

Jorna, R.J.; Wezel, W. van

WORLDMAKING WITH OBJECTS:

A CASE IN SEMIOTIC ENGINEERING

1. Introduction.

In the new world of computer media, computer games and interactive video it is a common experience for many users that some games, videos and software applications are more attractive than others. Even for those who are not familiar with computers and the new media, the same kind of experience is known for movies or books. Apart from the fact that some games, movies or books are more exciting than others because they are, for example, about sex, violence or adventure, the phenomenon remains that in some computer games or computer applications there seems some magic element that does something with the player or user. An example of a successful game is Doom (figure 1).

figure 1: Snapshot from Doom, an adventure program for kids

The aim of this game is to destroy the enemy. On the one hand the game corresponds to everyday experiences of the player. All kinds of objects and phenomena fro m reality appear in the game, such as a 3-D world, walls, characters, good and bad , weapons and light and dark. It is as if the player is in a real world. On the other hand there also are phenomena that only exist in a dream world, such as monsters, laser weapons and unlimited killing. The combination of real and fantastic propertie s stimulates the player and makes him feel engaged. The question is: is this the case; what is going on, here?

Another fact is that since the invention of the computer, its applicatio n domain has grown immensely. Starting from number crunching and computation in the early fifties the computer is now used for design, for all kinds of communication and for the support of intelligent tasks and shortly the computer also will be used for three dimensional experiences. From a communication or interaction point of view the most important part of the computer is the user-interface. In fact, it is the only part of the computer users normally get familiar with. It is the communication part of the system and it expresses the inter nal structure of the handling of the task in the computer and the behaviour that is asked for from the user. However it is still the case that much software has a semantic gap between what the user wants and what the computer offers. Many reasons may be given for this, but an important one is that the computer developed from an e xtension piece of the foregoing task handling into an instrument that can be conformed to the needs of a user. The first wave of information systems consisted in programming the admin istrative office and the financial administration. This was mainly done on main frames. Although the rise of th e personal computer changed the scale of automation, it did not change the startin g point of automation. This point consisted in imitating the existing manual procedures, into the software. A word processor was designe d as a typing machine with a couple of extensions such as manipulating and reordering existing texts. Software tha t supported book-keeping or accounting was a continuation of the older task ad ded with new features, such as classifying, ranking or comp aring. The reason for this was that in earlier times - only twenty years ago - the technical shortcomings of compute r systems required users to adapt themselves to the computer. This imperfection i s rapidly disappearing. Most software companies now promise to integrate the user perspective, although recently an advertisement of a certain computer firm sarcasticly

guaranteed that they would even make people compatible. Our statement is that in the future software will be adapted to the user, or it will not be. The problem , however is that nobody really kno ws what it means "to be adapted to the user". Key words that feature in discuss ions about new software developments include interactive, attention oriented, game-oriented or intelligent support. One direction thi s development is taking, starts from the window perspective. This can be illustrated by a comparison of the old v ersion of SPSS (a statistical package for the behavioral sciences) and the new version under windows (figures 2A and 2B).
SPSS-windows works as a memory aid for the user, it facilitates input and output manipulation, it enables loosely workin g and makes interaction of program and user concrete.

figure 2A: DOS interface of SPSS

figure 2B: Windows interface of SPSS

The Dos version, on the contrary, requires precision because of t he command structure. The communication protocol is more explicit and more rigid. Although the Windows version really is a st ep forward, we do not really know whether this is the direction we want to go in user-interaction. Part of the problem we are discussing here, is that question as well as answer are vague. Apart from the before mentioned phenomena the issue is whether we really know what is meant with adequate user-orientation. We will postpone the answer.

The examples of the thrill or excitement in some new computer games and the dead end street in which many software developers find themselves - implying an acknowledged failure of many existing programs - asks for a conceptual an d theoretical analysis of the "why" and the "how further" of this situation. As already

said, what is really going on here? Why is it that less effort has to be put in playing some games, that less mistakes are made and that the attention continues to be captured in some applications? Is it possible to analyze and catch the determining factors? Is the phenomenon known from other areas? Do cognitive psychologists, for example, have answers?

Describing and classifying the phenomenon, however, is not enough. One also needs a conceptual framework to get hold on the features of the phenomenon and more important to manipulate these features in such a way that (the illusion of) control is present. We propose a semiotic framework in which questions about communication, meaning, user-orientation and sign usage have a natural place. The semiotic perspective is especially important in relation to the interpretation of signs as they are relevant in dealing with exciting computer games and so called user-centered software. We come back to semiotic issues and semiosis when we discuss:

a) object-oriented analysis and design,

b) objects and worldmaking and

c) the way we communicate with and within (virtual) worlds.

The leading theme in this article is an answer to the question why some computer games and computer applications seduce many users to behave as if they were in real life. Part of the answer may be found in the metaphor of worlds in which actors behave in a natural way. This "world metaphor" must be contrasted with what has been called the "conversation metaphor" that is usually taken for granted in getting things done with computers. Another part of the answer may be that the excitement of some games and applications create an environment in which what Laurel called "first-person experiences" show to full advantage. In contrast with third-person experiences, first-person experiences invite to direct, that is to say non-computational, interaction with software environments. Again another part of the answer may be related to the cognitive architecture of regular users. It is a well known fact that people do not like to work with notations that require precision and rigidity. Users try to minimize their mental work load. Most times, conventional software requires the opposite, that is to say demand precision and lead to heavy mental work load.

The answers of the kind of metaphor, first person experience and cognitive architecture will be elaborated in section 2. Furthermore, it is always interesting to look for similar developments in neighbouring domains. In designing and program-

ming domains a major issue is the perspective on the programming ideology . Nowadays, it is old fashioned to work only within a procedural paradigm. Modern insights prescribe an object-oriented view. This is viewed to be more in agreement with the way we see the real world. Object-orientation as an explanation for th e easiness of window applications and certain computer games will be the topic o f section 3. Apart from the discussion about object-orientation, we also will go into the details of the shortcoming s of this orientation. We will relate this to the view on worldmaking advocated by Nelson Goodman.

In section 4 we will evaluate what may be gained from the object-oriented view on developing computer games and applications. What does the wor ld metaphor and the view that we are dealing with objects bring us, so that we may understand and manipulate its advantages for developing new software (and games)?

In the last section we will come to conclusion s related to semiotic questions. What is the relevance of semiotics in the new worlds of virtual objects and virtual space? What will be the nature of communication in the metaphor of worlds an d objects? What is semiotic about it? What are the consequences of this perspective for ontological engineering and, what we called, semiotic engineering? What is the sign-oriented character of being in worlds? We do not have final answers, but w e think that an integration of research from user centered design, information system development, cognitive science and semiotics is required.

2.1 1st personness and 3rd personness

In taking an interface as mimesis, Laurel (1986) uses notions from drama and theater to show what makes computer interfaces different from each other. In quoting Aristotle she says that the *end cause* of a thing is the function that it is intended to serve: that is what it is supposed to do (p. 68). A play may be used to inform people, to teach them lessons or to make them contemplate. It fails to do so, if the audience is not engaged in the play itself. In the same way, a computer program may be used to search for data, to design a new car or to have a user edit an article. If, however, its interface does not engage me, does not in a certain sense bewitch me, a successful interaction will not appear. The key notion, according to Laurel, is mimesis. "It is a certain kind of representation. It is a made thing, not an accidental, or arbitrary one: using a pebble to represent a person is not mimetic, using a doll is." (p. 70). From a semiotic point of view, at least in the Peircean sense, this implies that, for example, icons approach the ideal of mimesis more than symbols. Apple and the Microsoft Corporation intuitively followed this road in developing the windows and icon environment on PC's.

A mimesis is a closed system, says Laurel, and for that reason it is theoretically completely knowable. Therefore the actions in a play follow logically. A new situation in the play may appear, but it is within the scope of the possibilities. A nice, gentle and charming person does not suddenly change into a scoundrel, unless hidden clues are present earlier in the character of the person. The same holds for interfaces. If they are mimetic, users are not supposed to wonder why certain messages appear on the screen or why certain commands are not executed. Looking at the many interfaces that are all around us, we may conclude that this mimetic situation is never reached. The question is not only why this is the case, but also how the more ideal situation can be realized.

Beside the mimetic aspect with its consistency, closedness and its complete knowability, the user interface has one very important extra dimension: it is interactive. An interface, says Laurel, "is literally co-created by its human user every time it is used. The interface represents a whole interaction." (p. 73) User and computer should build up an interplay making up a new reality in which the user should be

engaged. The important remark is "should", because mostly non e of this is true. Take the following example from Microsoft Windows (figure 3).

figure 3: An ambiguous message on a Windows interface
(courtesy Han Numan)

A user is executing a program. The reply in a window is: "Aborting the program . Continue?" and a "Yes" button and "No" button (Numan, 1995). Such a message is ambiguous, offensive and frustrating. It closes every form of engagement. It makes completely obvious that in such a situation a priority switch appears from doing the task, for example word processing, to handling the computer system and the s oftware. As Laurel says: "End users are not interested in *making* a representation (like a programmer); they want to *move* around inside one. The context in which they wish to operate is the mimetic context." (p. 74). In line wit h Coleridge it can be stated that a user should have "the willing suspension of disbelief (p. 76). The suspension o f disbelief, however, is very fragile and very often the trust in a new (virtual) world is very rapidly violated (Numan, 1995).

The ideal situation in the relationship between user and computer is called first-personness (Laurel, p. 76). A user should consider himself to behave in a way as if he continuously says: "I am going to ..." o r "I can make ...". In comparison with first-personness, second-pers onness expresses itself in statements like "You have to ..." or "You should ...", whereas examples of third-personness are "She did so and

so" or "She was doing ...". First-personness means that a user is immersed in a world, that he or she is in it. Just as is the case in the movie Tron. In this movie a user is watching a computer game in which the good and the bad fight each other and suddenly finds him self inside the computer. He is then in the game and may run a risk of being shot at. Most present-day computer programs address users in the second-person or third-person mode. As some computer games show, a better way of interaction is the first-person mode.

The important question is how can first-personness be operationalized in design principles. One aspect is very clear, that is the mimetic character of the representational scene a user finds him self in. Flight simulators or car simulators are very good examples of these mimetic scenes. As Laurel says "first-personness is enhanced by an interface that enables inputs and outputs that are more nearly like real-world referents, in all relevant sensory modalities. [...] In product-driven applications, new technologies are allowing researchers to replace indirect or symbolic representations and manipulations with direct, concrete ones; e.g., physically pointing or speaking as opposed to typing, spatial and graphical representation of data as opposed to textual representation." (p. 77). There are other aspects that fall under the heading of interaction. These aspects are interactive frequency, interactive range and interactive significance.

Interactive frequency indicates how often a user is allowed to give input to the system. One extreme is the situation in which the user is only allowed to press one key or to have one response button on the screen. The other extreme is the situation in which the user is allowed to give input to the system any time, for example by pressing the right mouse button or the escape key. As a response the program will halt and ask the user what to do.

In the interactive range the one extreme is the situation where a user can choose between two options, for example, yes or no. On the other hand a very large range may appear, for example, in the still not realized natural language interfaces of computer systems.

Interactive significance is very hard to define. It has to do with the consequences of the actions of the user on the whole. In some games or application s choices have little consequences for the user, e.g., the difference in some word processors between save and continue and save and quit. Although the result seems

very important, the only consequence in the case of a mistake is that a user has to restart a program. The result has little significance. This is different in the situation where the user has to choose between quitting without saving and quitting wit h saving. Although the result has the same small interactive range as in the other two options choice, the interactive significance is enormous.

From the foregoing it may be clear that first-personness is the situation in which the significance is high, the range is as large as possible and the frequenc y nearly as high as in real life. It would be interesting to measure existing softwar e according to these standards.

2.2 World metaphor and conversation metaphor

The here exaggerated difference between first-personness and third-personness has major implications for interface design. The m essage from Laurel's analysis is that first-personness is the situation for behaviour in the real world, whereas the third-personness is mainly what is going on in settings where the communication is with language-like structures. Norman (1986) expressed this difference in tw o metaphors, the first called the "world metaphor", the other called the "conversation metaphor".

Characteristics of the world metaphor are that it consists of objects, tha t manipulation of objects is possible, that one can act as if one were in the world and that one experiences a feeling of direct engagement. Its most emphatic aspect is its directness. According to Hutchins, Hollan & Norman (1 986) the objects in the world metaphor can match the intentions of th e user, but they also can be on a lower level. The last situation leads to misunderstanding and unintended actions in the user - interface. One may state that in such a case the level of granularity of user and system does not fit.

An example of this phenomenon in a completly different domain is th e scheduling of nurses. Scheduling is defined as attuning staff and shifts. Two levels can be discerned. At the higher level it is important to look whether there are sufficient nurses to fulfill the required shifts. At a lower level it is important to assig n individual nurses to specific shifts. Capacity control, i n fact the higher level, requires

different information compared to the scheduling task itself, the lower level. This distinction is very important for the kind of support that is necessary, and, therefore, for the user interface (figure 4).

Properties of the conversation metaphor are that it is indirect, that is to say a user has to make a mental model for communication, that it requires (complicated) expressions to make things clear and that it normally demands a rigid syntax. In the conversation metaphor one also may distinguish several layers of granularity, that may be called low and high level languages. There is little agreement between experts which metaphor is better in which situation, although the general feeling is that a high level language in the conversation metaphor and direct manipulation in the world metaphor give a better interface communication. Between the two, however, direct manipulation defeats high level languages. The reason this sometimes does not work out has to do with the wrong approach or level of granularity of the objects in relation to the intentions of the user.

figure 4: Example of different levels of granularity in designing scheduling support; a: high level, b: low level

2.3 Experiential and reflective cognition

The distinction in world metaphor and conversation metaphor has strong semiotic and cognitive consequences. In the first place there is a contradistinction in expressions and objects. Although both have shape, the differences are huge. On the one hand there are syntactic and semantic structures, on the other hand there is a spatial structure. A second aspect is the difference in reference. In the normal situation language expressions refer to something, whereas an object is its own referent. In the third place there are differences in manipulation and construction. Expressions can be formed out of elements, whereas objects are elements in themselves. The building blocks are rather clear in the language metaphor, but not in the world metaphor. In the fourth place the world metaphor and the conversation

metaphor make an appeal to different cognitive abilities of the user. This last difference is more related to cognitive than to semiotic features.

The consequences of the world and conversation metaphor for the cognitive processing of the user can best be explained by referring to Norman's distinction (1993) in kinds of cognition. Generally, it is well known that human memory is organized so that the gist of the matter is more easily stored and retrieved than the details. Furthermore, humans are able to trace tasks attentively provided that there are changes in the environment. Humans are very sensitive for disturbances, for deviations in normal procedures and therefore can remember new things better than familiar ones. Humans are also very good in pattern recognition. We are able to recognize faces, structures and textures, instantly. More empirical findings about the human cognitive system can be mentioned here, but that is not the point. Important is that the differential effects can be attributed to differences in our cognitive system (Norman, 1993).

The two kinds of cognition that are relevant here, are called: "experiential cognition" and "reflective cognition" (Norman, 1993). Experiential cognition relates to experiences and to perceptions one undergoes. The perceptual experience, felt in all kind of modalities, leads to efficient performance. Reflective cognition, however, is focused on deliberation, reasoning and thought. It is extremely relevant for decision making and problem solving.

The numerous events, happenings or occurrences in normal life appeal to the distinct kinds of cognition. So far so good. Humans have different kinds of cognition, the world has many forms of appearances and the combination of the two result in various kinds of more or less adequate behaviour. As everyone knows this behaviour is not without complications, but a new kind of complication shows up when between the world and the user, interfaces are constructed. This gives problems, not only, as Norman (1988) and Rasmussen (1986) demonstrated, in the design and displays of various machines, but the last ten or twenty years also in computer interfaces.

A complete mismatch appears if instruments are designed that are supposed to support a particular kind of task, but as a matter of fact are related to another kind of task. According to Norman this is the case if one uses experientially oriented interfaces for reflective tasks and reflection oriented interfaces for experiential tasks. Stated in semiotic terms one can say that in these situations the structure and sign

in the design of the interface hamper adequate semiosis (understanding). Norman gives several examples of these mismatches for computer and non-computer situations. Firstly, experiential instruments that do not fit experiential cognition, such as car radio's that wrongly require a lot of reflection. Secondly, reflective instruments that do not fit reflective cognition. An example is software for designing and drawing that does not support exploration, problem solving and comparison. A "bad" example is Slide Write, a "good" example Corelflow. Thirdly, an instrument may be important for experiential behaviour, but leads to reflection. A "dangerous" example is a button for forced stop that cannot immediately be discerned from ten surrounding identical buttons. One first has to think before one can act. Disasters may be the result.

Experiential and reflective cognition are related to the before mentioned distinctions in first- and third-personness and world and conversation metaphor in a special way. First-personness means that users are within a world, that interaction is direct, in the sense of objects that can be manipulated. This very much suits the cognitive mode of experiential cognition. On the other hand third-personness takes distance between user and computer system for granted, which means that thinking and deliberation are the cognitive mode users have to be in. This means that reflective cognition is at work and that the conversation metaphor naturally fits these interface situations. Semiotic issues differ completely in both. On the one hand language, syntax and semantics are important, on the other hand objects, object manipulation and the context of objects, the world, and the way we deal with these contexts. Until now semiotics had much to say about language, syntax and reference, but there is not much theorizing to be found about objects, object making and context or world-making. The only interesting development exists in an area that is far remote from semiotics, namely object-oriented analysis and object-oriented design (Jacobson, 1992). We will first discuss this new theoretical entry that seems so much in line with the world metaphor and first-personness and then we will evaluate its relevance for the analysis and design of new interfaces.

3.1 Worlds and object-orientation

The object-oriented approach starts from the assumption that modelling software in a computer system should correspond to reality. Those properties of the world that are relevant for computer support are described as objects that are connected together. As in every modelling system the primitives that are used are limited, but the limitations are a result of the assumption about the way the world is.

The object-oriented approach has its origin in the computer environment itself. A computer program consists of a static part that describes the state and a dynamic part that changes the state. The static part is called the data structure, whereas the dynamic part is called the program structure or procedure. In conventional software, data and procedures are separated. This principle has consequences for the modelling techniques that are used. Most techniques emphasize data or procedure, but not both. Emphasizing the one, necessarily leads to neglecting the other. Entity relationship modelling is data structure oriented and neglects the dynamic aspects, whereas ISAC models the processing or procedural structure and does not pay much attention to the static aspects of the modelled world. The object-oriented approach wants to get rid of this forced division and tries to model the static and dynamic structure at the same time. An object is a structure of data and processes. Processes and data that belong together are called to be encapsulated.

As the name already indicates the central elements in object-oriented programming are the objects. A computer program consists of several independently acting objects that each manipulate their data with their own pieces of program. The data structure of an object consists of several attributes and the processing structure consists of methods. Values on the attributes determine the state the object is in. Methods change these states by executing operations on attributes. For example, the method "open the throttle" of an object "vehicle" changes the attribute "speed" from "0" into "50".

In an object-oriented programming language a set of identical objects or tokens is described by a class or type. Classes are distinguished from each other, because they describe different attributes and methods. A difference between the classes "car" and "plane" is that the last one has wings. Objects or tokens of a class are distinguished from each other because they have different values for common attributes. Two tokens of the class "car" are distinguishable because they have different number plates.

Objects may be connected with each other. Mostly the aggregation-relation and the generalization-relation is modelled (figure 5).
Aggregation means that an object consists of several parts that are objects themselves, such as a car that consists of wheels, a motor, etc. In the generalization-relation properties that different classes have in common are described in a class one level higher. Because the class at a lower level inherits the properties from a class a level higher, the relation is called inheritance. Inheritance makes it easier to describe a new object because only that parts have to modelled that are not described in another class.

figure 5: Two different perspectives on the same object: 'car': generalisation and aggregation

As already stated, the object-oriented approach has a technical background. The first object-oriented programming language was Simula, and afterwards Smalltalk and C++ were introduced. It is sad to say but the emphasis in the development of object-oriented languages is on technical cleverness and not on the semantic modelling power. As a result of this there was more interest in all kinds of technical tricks that facilitate programming, whereas the modelling of objects in reality became more and more problematic. This shortage was realized and resulted in modelling techniques that could be used in the analysis phase of developing software. This adapted approach not only leads to a stepwise refinement of system specifications, but is also important in the communication to the user. Still, these models describe the computer system at several levels of abstraction instead of modelling reality as such.

We introduced the object-oriented approach as a possible answer for the development of more realistic interfaces and computer programs. The advantages of this approach are clear. Because of the levels of abstraction, it is a better structured programming tool and it offers better communication possibilities with users. Levels may be hidden or encapsulated, because they are not important for the overall view of the application. It also gives the possibility to reuse components that are built for specific applications. Furthermore, there is a direct relation between object-oriented analysis, object-oriented design and object-oriented programming. And last but not least the object-oriented approach claims to be more in line with the way people see the world, namely in terms of objects, properties and classes.

However, it is also clear that there are several disadvantages, for example, that the programming perspective is dominant. Furthermore, it is not clear what the context of the objects is. Objects are located in a world, but the world is not stated explicitly. We may say that an object-view not necessarily brings about a world-view. In comparison with classical ways of designing and programming software and user-interfaces, the object-oriented approach is more than one step further towards first-personness and the world metaphor in the sense of direct manipulation, but the final goal has not yet been reached. Therefore, we suggest to have a look at a

supplementary more philosophical perspective, called ways of worldmaking. This perspective, proposed by Nelson Goodman, may give the extr a dimension the object-oriented approach is missing. We then might have ways to conceptualize and t o implement interfaces that are in the first-personness mode.

3.2 Objects and worldmaking

The shortcomings of the object-oriented approach are clear, although the overal l perspective is welcomed. The question is what exte nsion can be found in Goodman's ways of worldmaking. Goodman has two important starting points. Firstly, he claims that we live in and make artifacts that may be called worlds. Worlds may be small or large or extended or contracted, but they always are designed, constructed an d man-made. Secondly, there are no a pri ori reasons that one world has priority above another world. Therefore, truth is relative and depending on the way the world i s made.

Ways of worldmaking is Goodman's answer to the questi on how to deal with different versions or descriptions of phenomena and events. All these descriptions may constitute "the" world. In practice we have to do with versions that constitute a world and, according to Goodman, right versions constitute right worlds. Hi s approach, he says, "is rather through an analytic study of types and functions o f symbols and symbol systems. In neither case should a unique result be anticipated; universes of worlds as well as worlds themselves may be built in many ways" . (Goodman, 1978, p. 5). Worlds consist of matter, energy, waves and phenomena . Worlds can be describe d in words. Here an asymmetry appears, because we cannot have worlds without words, but we can have words without worlds. The important question is: how can worlds be made, tested and known?
Goodman discerns five ways of worldmaking: a) composition and decomposition, b) weighting, c) ordering, d) deletion and supplementation and e) deformation.

All worldmaking starts with composition and decompositio n. This means the putting together of different entities, be it objects, classes, members or subclasses. Goodman says that this process is normally done with names , categorizations, labels, pictures and so on. When a hundred years ago soccer players acting in the world of

football took the ball in their hand and ran towards the goal-line and pressed the ball on the ground a new world was created, called rugby. This is a very down-to-earth example, but many more sophisticated instances can be found. Many worlds are created in a similar way.

Another part of worldmaking consists in weighting. This has to do with the difference in emphasis on entities and properties that constitute a world. Two worlds may have the same entities, for example, different kinds of cars, but in relation to the purpose or function of these worlds, the distinctions in the first world go as far as kind of model and year of manufacturing, whereas in the second world the important aspect is whether transportation is by ship, car or plane.

Ordering is a way of bringing specific structure into a world. Two identical worlds consisting of the same entities, may differ in their ordering relation. In broadcasting a portrait of the queen, the portrait may be an oil painting or a digitally remastered image. The ordering of the elements in the first situation is not really relevant. In the second situation, however, the ordering of the pixels, is decisive for transmitting. Even the different ways of ordering within the digital environment are important. This ordering can be vector-graphic, bitmap and so on. According to Goodman these orderings are not found in the world, they are built into the world.

The example of soccer changing into rugby also makes clear that composition and decomposition naturally imply deletion and supplementation. Playing by foot was supplemented by playing by hand. Deletion and supplementation also become clear in the kind of instruments we use. A digital thermometer, measuring in hundredth of degrees, has advantage over an old analog one measuring the human body temperature more roughly. Another interesting example is the phi phenomenon. Looking at two flickering points at a certain distance, gives the impression of (apparent) motion in the eye of the beholder (Goodman, 1978, p. 15). Something is supplemented that is not really there.

Another less important process of worldmaking is deformation. This is involved in making corrections or distortions within existing worlds. Correcting a digitally transmitted image is an example of positive deformation, adding a moustache to the face of the Mona Lisa is an illustration of negative deformation.

Ways of worldmaking also make clear that it is rather indifferent to determine which world is the real world. All worlds are constructed, are artifacts, and,

as such, there is no final frame of reference. However, this does not imply that anything goes. The problem is, says Goodman, that we may have trouble with truth. Truth still holds as a concept to indicate whether versions and worlds correspond or are consistent, but truth does not matter any more as the ultimate referee for all worlds. Truth already was a problem when a world contained no statements at all, truth now is also problematic, because there is no ultimate reference. According to Goodman, truth relates to the reality of a world and reality, he says, is largely a matter of habit. This perspective has strong implications for the way we deal with designing virtual worlds, object-oriented worlds in relation to the so called real world.

4. Conclusions: User-orientation, engineering and the semiotic consequences

Although having its predecessors in Wittgenstein and Winch, Goodman's project of the ways of worldmaking was rather new in the seventies. Nowadays it is common usage to make worlds, of course in a more technical sense than Goodman analyzed. If we strip the details, modelling software and information systems is nothing but worldmaking. We discussed this already in the object-oriented approach. What yet cannot be found in this approach is the perspective that one is making worlds, that one is composing, ordering, supplementing and deleting. In information science, computer science and cognitive science much may be gained from a sign-oriented (semiotic) point of view as well as from Goodman's perception on worldmaking. Object-orientation and worldmaking belong together and both may be used to model and understand first personness and the world metaphor in user interfaces.

Finally we have to answer the questions we asked about the "what" and "why" of the advantages of some games and software applications compared to others.

What is the relevance of semiotics in the new worlds of virtual objects and virtual space? If the conclusion from the above is that perception, first-personness and the world metaphor are conceptually linked together, at first sight the answer is very disappointing. Semiotics has little to offer, here. Truth, reference and communication structures are problematic features of directness, first-personness and experiential cognition. However, if we connect the objects in the worlds we construct

with Goodman's ways of worldmaking, semiotics is of relevance, but not immedi ately applicable.

The same ambiguous answer also holds for the qu estion what the nature will be of communicat ion in the metaphor of worlds and objects. Perception, directness and first-personness ask for a different kind of communication than we are used to in the language-like interp lay. The semiotic apparatus for this is missing. Semiotics as the study of sign systems and sign use is inadequate. Conce pts that might give grip on the feeling of directness and engagement relate to attuning and res onance, but how this relates to semiotics is unclear.

The question of the sign-oriented character of being in worlds is therefore very easy to answer. Worlds may consist of all kinds of build ing blocks, that is to say objects or signs. We may reac t to signs in worlds; not in a communicational kind of way, but like in actions. Furthermore, there is not one world, but there are many . Acting in a world is not a ma tter of communication, but a matter of habit. Here, the relevance of Goodman's ways of worldmaking is evident.

The discussion of the world metaphor, the ways of worldmaking and th e construc tion of worlds in the case of information system development, interfac e design and software applications also answe rs the question what kind of engineering is suitable for user-centered interfaces and games. One might stick to the old idea that artifacts re semble "the" world. Ontological engineering is coined as a label for this approach. However, it is better to say that worlds are created and that feeling well within a world is a matter of used sign systems and therefore a matter of habit. In this case we talk about *semiotic* engineering. Not because descriptions ar e depictions, but because descriptions create depictions.

The contrast is eloquently stated by Lenat & Guha (1990) who call th e application-oriented construction of systems of represent ation primitives: ontological engineering. "Choosing a set of representation primitives (predicates, objects , functions) has been called *ontological engineering* - that is, defining the categories and relationships of the d omain." (1990, p. 23). The process is caught in a language called CYCL in which rules are specified for handling units (objects), slots, (predicates, attributes), v alues, entries, kinds of slots, variables, propositional objects and so on. In onto logical engineering nature is carved on its edges in such a way tha t objects, relations, agents, events, processes and so on are represented in a general

language. The basic idea is that a world in all its complexity can be represented by a finite set of language-like entities.

The most problematic aspect of ontological engineering is the assumption of a basic ontology. In the construction of information systems, that is to say databases, decision support systems and knowledge systems and, especially, user interfaces, the starting point of an ontology is questionable. What we find in organizations are constructs, artifacts and more or less complicated man-made sign structures. None of this can be considered to be an ontology in a classical way. In this sense building any kind of information system or user interface, does not presuppose an ontology, but creates a structure; a semiotic structure.

Now, objects are signs, but not necessarily symbols. Larger fragments of words are worlds. And worlds can be created. In fact, they are created, for example in playing computer games or in being engaged in adequate user interfaces. The theoretical underpinnings for worldmaking with object are clearly visible. It is time that more practical applications become available.

Literature

Goodman, N. (1978). *Ways of WorldMaking*. Hassocks: Harvester Press.

Hutchins, E.L., Hollan, J.D. & Norman, D.A. (1986). Direct Manipulatio n Interfaces, in: Norman, D.E. & Draper, S.W (Eds.). *User Centered System Design*. Hillsdale: Lawrence Erlbaum.

Jacobson, I. (1992). *Object-Oriented Software Engineering*. Reading: Addiso n Wesley.

Laurel, B.K. (1986). Interface as Mimes is, in: Norman, D.E. & Draper, S.W (Eds.). *User Centered System Design*. Hillsdale: Lawrence Erlbaum.

Lenat, D.B. & Guha, R.V. (1990). *Building Large Knowledge-Based Systems*. Reading: Addison Wesley.

Norman, D.A. (1986). Cognitive Engineering, in: Norman, D.E. & Draper, S. W (Eds.). *User Centered System Design*. Hillsdale: Lawrence Erlbaum.

Norman, D.A. (1988). *The Psychology of Everyda y Things*. New York: Basic Books

Norman, D.A. (1993). *Things That Makes Us Smart*. Reading: Addison-Wesley

Numan, H.J. (1995). *Trust in User-Interfaces*. Dissertation to appear in 1996. Groningen: University of Groningen.

Rasmussen, J. (1986). *Information Processing and Human-Machine Interac tion*. New York: North Holland.

René Jorna & Wout van Wezel
Faculty of Management & Organization
University of Groningen
P.O. Box 800
9700 AV Groningen
The Netherlands