

Doubting the OBJECT World

CECILE K. M. CRUTZEN and JACK F. GERRISSEN

Faculty of Science and Technology, Open University of the Netherlands

Abstract: Object Oriented approaches (OO) may be placed within functionalism. It is argued that we should not use OO for analysing human worlds. OBJECTs are neither presentations of humans nor equal to humans as the literature often suggests.

But everything which is assumed without question, which is taken for granted in our intercourse with one another and nature is what, at the given time, is called knowledge. Thinking on the contrary starts... from doubt or uncertainty. It marks an inquiring, hunting, searching attitude, instead of one of mastery and possession [8].

1. THE ONTOLOGICAL STRUCTURE OF THE OBJECT WORLD

The OBJECT is the basic unit in a description of an OBJECT world, which functions as a script for an 'interaction play' of cooperating OBJECTs. A presentation of an OBJECT world is like performance in theatre. If the OBJECTs are models of real entities then the performance is only a modelled presentation of a part of real world interaction. OBJECTs are actors in a play, with predetermined dynamic and static properties described in classes of which they are instances. In an OBJECT class the dynamic and static properties are bundled together, interdependent and encapsulated. Every OBJECT is an input-output-system with invisible internal behaviour and information. During the performance only the OBJECT surface is visible. The environment of an OBJECT consists of all

other OBJECTs that have the potential to interact with it. The script further consists of the description of structural relations between the OBJECT classes. Each OBJECT class can be specialised or generalised (is-a-relation, inheritance). Another important relation, the aggregation, is an OBJECT (class) which is composed from several parts described by separate OBJECT classes (is-part-of-relation). For an aggregation it is necessary to define the cooperation between the aggregated parts. The OO script has a potential of many different performances, since it depends on the OBJECTs which are 'asked' to play and the events that will take place in the environment of the performing OBJECTs. But the potential of the performances is restricted to the planned behaviour of these OBJECTs, acting within a life cycle with a set of finite discrete states and transitions. OBJECTs can not improvise.

2. THE CONSERVATISM OF THE OBJECT ORIENTED REALISATION

OO are not revolutionary. Reuse, inheritance and encapsulation are old concepts in software engineering and have been merely formalised within one approach and named explicitly. The modularisation of systems and software is based on one single concept: the OBJECT: object-oriented techniques allow us to reuse far more than code. We can reuse requirements, analysis, design, test plans, user interfaces and architecture. In fact, virtually every component of the software engineering life cycle can be encapsulated as a reusable object [23, p.6].

OO has reinforced functionalism; seeing an information system as a system which can transform well-defined inputs into outputs, meeting prescribed requirements with measurable attributes, controllable functions and desirable features, fitting in well-ordered organisation and seeing the system as a neutral actor in meaning construction processes. The OO system is 'objective' by name. Subjectivity is seen as errors of people [15, p.46-69]. The production of large class libraries has led to a tendency of realisation to reuse something old which is available in place of creating something new which is really suited to and focused on new settings of interaction. OO conserves and closes the meanings of the past and does not open up the future as a projection of our possibilities which we, as beings in the world of every day, could anticipate on. One of the most dominant ideas is to produce software without ambiguity and unmastered complexity. Software should be controllable and deterministic. Software engineers are afraid of the complex, the chaotic, the undecided and the unpredictable.

3. A METAPHORICAL VIEW ON THE OBJECT ORIENTED APPROACH

The evolution of OO from the data oriented approach can be compared with the evolution of filming from photography. Both are meant to represent reality. Data model and photo are static representations of a moment in a dynamic world. The position of the camera, the lighting and the photographer's perspective are subjective. A data model contains and represents the view of the modelling subject and the abilities of the used modelling method used. The subject has disclosed some elements of reality and negated some of the others and this presented view is seen as the 'truth' of the data model. The dynamic aspects of reality are presented statically in the photo and database model. Only a series of 'photos,' or data models, can give a glimpse of the real world dynamics, represented the way it was seen by the subject. Shifting from photo- to movie-representations and from a data model to an OO model could shed light on the dynamics. You can record with OO some of the real world dynamics and represent the changes as series of discrete moments in time. There are changes that the OO camera can not disclose or which have happened between the separate takes done in different times. The OO camera is only focused on those transitions and states, which are planned in the observation script and observable by the OO camera. It cannot discover spontaneous changes. The 'camera-subject' can take only the dynamics, which match the expectations. OO strongly suggests that it represents the total dynamics of the reality objectively like a documentary. It also suggests that the meaning construction process of the audience during the performance of the OO play only needs the reception of unambiguous objective messages. Filming and OO are seen as a progression in representation techniques. But a lot of the presentation dynamics take place behind the white screen of the OO 'stage'. The rules and procedures, which will direct the transitions, are encapsulated under the surface. These are not identical with the situations, the discussions and the causes of the transitions that took place in reality. In the OO representation changes are caused by messages from one OBJECT to another OBJECT or by descriptive rules which force an OBJECT to change its state. The process of change in an OO ontology is modelled like a stimulus-response procedure. Additionally, neither the script of OO observation (recording) nor the script of the OO presentation are identical with the way the actions took place in reality. But as with the movies, OO opens up the possibility that the act of meaning construction of the audience and the act of observation of the camera-subject can be prescribed in scripts. Prescription that could increase the potential of representing subjectivity and "multiple voices" [21, p.22] of the subjects could give room to the audience to start their own constructional

processes. However, at the same time these scripts open up the potential of the dominance of views concerning how and why the world has changed and how this OO play should change the interaction of the 'audience' with the represented reality. But software engineers are not aware of that potential of subjectivity. OO is the ultimate tool for representing phenomena neutrally that discloses the relevant aspects of problems that can be solved later:

While object-based approaches can theoretically be used to implement either subjectivist or objectivist interpretations of data, it is possible to conceive of them as predisposed towards subjectivism... Surprisingly, the literature has taken the opposite direction: objects...are conceived to be the representation of real world objects [15, p.62-63].

But this supposed neutrality and objectivity are precisely the dominance and power that software engineering exercises:

Perhaps only to the extent that one person or group can dominate the whole can 'reality' appear to be governed by one set of rules, be constituted by one privileged set of social relations, or be told by one 'story' [10, p.28].

4. THE COLONISATION FROM REALISATION INTO THE ANALYSIS

Out of the ideas on controlling complexity and on reducing ambiguity and the relatively successful deterministic approach of software realisation software engineers got the illusion that they know how to master the complexity and ambiguity of the real world, too. Equipped with their powerful software tools for abstractions such as classification and division they started the colonisation of the analysis process of real world interaction. This is enforced by OO through its strong opinion that it is possible not only to handle the facts but also to handle and therefore control behaviour itself. This colonisation process is now dictated by the fear of the analysing subject in meeting complexity and ambiguity. Instead of confronting the complex it tries to avoid it altogether with the ambiguity of transformation and translation by selecting the most formalised documents, texts, tables, schemes, etc. which are close to the syntax of the OO language. Software engineers are feeling more comfortable in the 'truth' of these formal models:

A model simply provides us with a richer, higher level, and more semantically precise set of constructs than the underlying natural language. Using such a model reduces ambiguity, makes it easier to

check for incompleteness, and may at times improve understandability [7, p.214].

OO has also fallen on the fertile soil of Taylorism: the division of work and organisations in closed units with formal interaction patterns and the reduction of humans to functions. To see ‘truth’ as an ongoing process of meaning constructions, linguistically and socially, between actors in an analysis process and in an organisation as a whole, only complicates matters. OO is seen as the right language for naming humans, things, the objects of reality, and the OBJECTs with the same words. This naming is supposed to be the objective and unambiguous translation from reality into the analysis results and from the designed model into realisation. The suggestion that OO was focused towards the end user is misleading because OO was started out of the need to reduce the gap between the analysis process and the realisation: “object (oriented) programming allows a more direct representation of the real-world model in the code. The result is that the normal radical transformation from system requirements...to system specification...is greatly reduced” [18].

5. THE FEMINIST CRITIQUE OF EPISTEMOLOGICAL ASSUMPTIONS IN OO

As stated by Jane Flax, the assumption that every person gives the same meaning to the object if the name of the object is the same, posits or presumes “a realist or correspondence theory of language in which objects are not linguistically or socially constructed; they are merely made present to consciousness by naming or by the right use of language” [10, p.31]. In the correspondence view, representations are one-to-one projections of people and things in reality. The OBJECT corresponds with what is “out there” [5, p.183]. Every perceived object has a structure that is independent from the observing subject. OO is only the mirror that helps the subject to discern and represent that structure objectively. In the correspondence view the meaning construction process is reduced to a selection process of the relevant aspects of the object. The OBJECT has the same meaning as the original object if the structure of OBJECT and the structure of the object are the same. In the correspondence view there can be no misunderstanding between the subjects which represent and the subjects which will perceive those representations because OO is assumed to be very suitable to extract the ‘metastructure’ of everything that exists in reality. Grady Booch tries to prove this assumption of the existence of this uniform metastructure by several examples of very divergent existing systems: the PC, the plant and animal, the galaxy, the atoms and social institutions [2]. Within OO the metastructure is ‘a world

divided in OBJECTs with a stimulus-response cooperation and an encapsulated self.' OO modelling in this concept of truth is not revolutionary within computer science because it has in it all the basic assumptions of functionalism. It supposes that the real world is structured orderly and that we can represent the world objectively independent of the observer's appreciation. The OO model is 'true' if it "accurately depicts the underlying reality of the universe of discourse. Different opinions...must be a reflection of human error and...can be eliminated...Objects such as rivers, fog and crowds of people" [15, p.58] are not representable in the form of closed objects. But OO proponents are anxious to control these phenomena by saying that these concepts such as time, beauty, colour; emotions such as love and anger are attributes of objects. "For example, we might say that a man (an object) loves his wife (another object), or that a particular cat (yet another object) has gray fur" [2, p.77]. It is supposed that for most subjects which use OO there is no difference between the interpretation of parts of the world and the representation of these parts and the construction of artificial products. "Our surroundings, for instance, consist of objects, such as people, trees, cars, towns and houses which are in some way related to each other...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...We use 'person' and 'object' in this description to mean the same thing; we actually mean the object that represents the person...one should be careful to separate the reality from the model" [16, p.43-45].

The hypothesis that if the constructed software has the same structure as reality then this artefact has the potential to fit in the same way in the world of software engineers as in the world of users. They think that users also want and will understand each OBJECT in the meaning of mastery and possession and will emphasize its structure as natural.

Conclusion: The OO approach conforms to Enlightenment traditions

- that subjects apply a correspondence view of reality,
- that everybody sees the world as a world with one metastructure,
- that observing subjects are sustainable,
- that everything and everybody can be represented as OBJECTs; there is no fundamental difference between people or things,
- that change is a logical process of action and reaction, and
- that there exists a common language in which all people (who are affected by software) can understand each other.

OO is presented as a language that is very close to that ideal. From feminist theories [1, 4, 11, 14] one may have doubts about these assumptions because they are based on the same illusions of objectivity and neutrality of representation; the negating of power and dominance by translating it into 'natural and obvious' and on the existence of truth by transforming it into

progress, which has classified woman and femininity as unitary and subordinated in opposition to man and masculinity. In OO the significant dualism is between the formal and the informal, the predictable and the unexpected and last but not least the OBJECT and the human. We should scrutinize the methods and languages “by which meaning and categories are constituted” otherwise “one only imposes oversimplified models...that perpetuate conventional understanding rather than open up new interpretive possibilities” [19, p.135]. The presence of OO products enforces the disclosure of what OO has hidden. Transformation can start out of doubt: “As feminists we are led to battle with the abstractions in several ways: noting that they are historically specific, not timeless...not universal; biased, not neutral. We want to make what the abstractions have hidden, visible” [20, p.82].

6. THE VIEW ON CLASSIFICATION AS AN OO ABSTRACTION

To classify and to standardise is a human activity. Without this we find it difficult to live and work in a world of artifacts. Classification and standardisation are always political processes with a high risk of dominance. Ideally, these should be negotiated descriptions and rules, because there are no unitary schemes for doing the work of classification or defining a standard. Conformance to one standard should be a construction process in communities because there are many different classification schemes even within the same object domain [3]. In OO a class is not negotiated. It is a construction that functions as a factory to produce similar OBJECTS. Diversity can only be constructed by specialisation out of a predetermined similarity. To classify means to recognise the similarities between objects and interactions in the real world, to concentrate on the similarities, to ignore for the time being the differences [2]. Phenomena of the real world are reduced to OBJECTS with a limited amount of properties, states and with formal planned procedures. Differences are mostly neglected because they are not easy to handle and are only opposite to equal [19]. An object can be represented as a member of several classes only when it inherits all the properties and procedures of these classes completely. It cannot escape the hierarchical structure within OO. The class concept is at a loss in representing a social process because a group of humans can only be an aggregated class with a harmonious and planned coordination structure. A group can only exist if it has planned transitions and a fixed goal. Grouping itself can not be an ad hoc social process in which people feel connected or disconnected. It can never be a social process in which groups come into

being or disappear [9]. A mutual concern cannot be presented as a negotiation process; it can only be planned as a formal purpose. Because of the lack of presentation of possible diversity in and between objects, social groups and of the possible change in transition in objects and social groups, OO is not suitable for analysing these processes. The filters of classification and of aggregation cannot disclose such social processes. A change in an interaction process can only be represented as sets of states and sets of transitions between these states. They can only occur when the conditions in the environment are confirming to prescriptive rules. Within OO the 'change of change' is zero. There is no need of a meaning construction processes in OO because they cannot exist within its ontology. Situated action is essentially ad hoc and therefore OO cannot be the ultimate way for "artful integration work" [22, p.16-18]. Methods such as OO are used because they seem to prevent ambiguity and doubt. It is precisely ambiguity and doubt makes an organisation creative and alive. In OO ambiguity and doubt are only hidden, they are not absent. Therefore we have to 'leave' the concept of OO and use it only for the purpose it was meant: The realisation of software. Realised OO software consisting of predictable and planned interaction cannot form the bases of the representation of humans because otherwise we would make humans into an available resource (Bestand) which can be ordered repeatedly [13, p.14-24].

7. CRACKING AND ENJOYING OBJECTS

In the role of audience people can enjoy the OBJECTS' play in the OO world. The OBJECTS' play can be a useful tool integrated in our daily live. Even more, if we could change our role of only being a passive audience member for the role of being the director or scriptwriter of the play. Then we could create new OBJECTS or aggregate old ones into a new surprising play. We could solve the conflicts within the aggregated OBJECT so that they cooperate in a way suited for special situations and act in a way that gives priority to our purposes. But therefore the OBJECTS should stop acting behind their surface, even if this would render our self-created OBJECTS unpredictable or unreliable. We enjoy the most those plays which plot we do not know. Open OBJECTS can give us the opportunity to edit the OO movie and to replace parts with our own (inter-)actions. But not when we are actors within the OO play as Brenda Laurel suggests [17]. Because if we are, as users, just actors in an OO movie, then we are determined to play as OBJECTS, without any doubt, and follow the prescriptions of the given

script without thinking. Functioning as an OBJECT we cannot escape from the life cycle of states and transition rules. OBJECTs living in a movie cannot step out. As an alternative of performing only in the role of an audience, the director role is preferable. As humans we can enjoy the interactions in OO performances. But at the same time we would like to create new interaction worlds out of the present at hand OBJECTs, too. Education experiments including the critical discussion partner Gender studies already show that this can be very successful and enjoyable [6]. OO as it is used for the (re)presentation of the dynamics of interaction worlds gives the opportunity to discuss the nature of human behaviour. Knowing that the essence of human behaviour is not predictable and is situated in the interaction itself we can discover that OO will only disclose planned action. Therefore we should have doubts looking with the OO camera into the world of humans. OBJECTs can not be representations of humans or act in the same way as humans. And that is precisely the essence of their attraction.

REFERENCES

1. Alcoff, L., Potter, E. (1993). *Feminist Epistemologies*. New York: Routledge.
2. Booch, G. (1991). *Object Oriented Design with Applications*. Redwood City: Benjamin/Cummings.
3. Bowker, G. C., Star, S. Leigh (1996). How things (actor-net) work: Classification, magic and the ubiquity of standards. <http://alexia.lis.uiuc.edu/~bowker/actnet.html>.
4. Code, L. (1991). *What Can She Know? Feminist Theory and the Construction of Knowledge*. Ithaca: Cornell U. P.
5. Coyne, R. (1995). *Designing Information Technology in the Postmodern Age, From Method to Metaphor*. Massachusetts: The MIT Press.
6. Crutzen, C. K. M., Vosseberg, K. (1999). Die Interaktion zwischen objektorientiertem Denken und feministischer Kritik - eine dynamische Verbindung. In: Dreher, B. e.a. (Eds). *Software Engineering im Unterricht der Hochschulen*. Stuttgart: Teubner Verlag.
7. Davis, A. M. (1993). *Software Requirements: Objects, Functions, and States*. Englewood Cliffs: Prentice-Hall International.
8. Dewey, J. (1916). The Individual and the World, In (Eds.). *Democracy and Education*.
9. Evans, J. (1995). *Feminist Theory Today* London: SAGE.
10. Flax, J. (1990). *Thinking Fragments; Psychoanalyses, Feminism and Postmodernism in the Contemporary West*. Berkeley: University of California Press.
11. Harding, S. (1991). *Whose Science? Whose Knowledge?* Buckingham: Open U. P.
12. Heidegger, M. (1936) *Der Ursprung des Kunstwerkes* Stuttgart: Philipp Reclam (1960).
13. Heidegger, M. (1962). *Die Technik und die Kehr..* Stuttgart: Verlag Günther Neske.
14. Hekman, S. J. (1990). *Gender and Knowledge*. Cambridge: Polity Press.
15. Hirschheim, R., e.a. (1995). *Information Systems Development and Data Modeling*. Cambridge: Cambridge U. P.
16. Jacobson, I., e.a (1992). *Object-Oriented Software Engineering*. Reading: Addison-Wesley.
17. Laurel, B. (1993) *Computers as Theatre*. Reading: Addison-Wesley.
18. Ledbetter, L., Cox, B. (June,1985). Software-IC's. *BYTE* .

19. Scott, J. W. (1990). Deconstructing Equality-Versus-Difference. In: Hirsch, M. e.a. (Eds.). *Conflicts in Feminism*. p. 134-148. New York: Routledge.
20. Star, S. L. (1991). Invisible Work und Silenced Dialogues in Knowledge Representation. In Eriksson, I., *WWC*. p. 81-92. Amsterdam: North Holland.
21. Suchman, L. (1994). Working Relations of Technology Production and Use. In *Computer Supported Cooperative Work*, 2 (1-2), p. 21-39.
22. Suchman, L. (1994). Supporting Articulation Work: Aspects of a Feminist Practice of Technology Production. In Adam, A., *WWC, Breaking Old Boundaries - Creating New Forms*. p.7-21. Amsterdam: Elsevier Science.
23. Yourdon, E., Argila, C. (1996). *Case Studies in Object Oriented Analysis and Design*. Upper Saddle River: Yourdon Press.