

On the object and process of software product development

Pentti Kolari
pentti.kolari@nokia.com
Nokia Telecommunications
Network Management Systems
P.O. Box 759
FIN-33101 Tampere, Finland
+358-3-2577742

Abstract

There are still problems in developing useful and usable software products. One potential reason for this is that the nature of software product development is not yet fully understood. In this paper, a conceptual framework for understanding software product development is provided. The relevant contexts of a product are identified and product development activity is studied as part of this whole. Models for the object, process and structure of product development activity are presented. It is suggested that the object of product development activity is not only the product itself but the product in its context of use. This should be reflected in the process of product development, too.

Keywords: Software product development, object, process, conceptual framework, context

BRT Keywords: FA, FD

Introduction

The goal of software product development is to develop useful and usable products. Various types of models have been proposed and used for understanding and guiding the process of product development. However, there are still problems in developing useful and usable products. Why is it so? One potential reason is that the nature of software product development is not yet fully understood.

In this paper, I will try to deepen this understanding. I will put forward a conceptual framework for understanding software product development activity. The usefulness of such a framework is based on the assumption, and belief, that practical mastery of work activity often implies its conceptual mastery, too. This is the case particularly in work activities where the object of activity is intangible and elusive. In this respect, software product development is a case in point. The type of software products I have primarily in mind are computer-based information systems intended to be used by human users as tools in supporting their work activities.

When studying the organizational contexts of development and use of computer-based artifacts, Grudin (1996) has distinguished between three organizational contexts of development: competitive contract development, internal or in-house development and commercial or off-the-shelf product development. The context of development of interest

in this paper is the last one of these.

The study of software product development is important because today more and more organizations are using software products instead of in-house systems. There are at least two reasons for this trend. First, it has turned out that the development of in-house systems is extremely expensive. Second, organizations are more and more focusing on their core businesses and many support activities are outsourced. Consequently, the willingness even to change the work activities when implementing information systems has increased. Information systems are used to enable the change of work activities, not only to automate existing information processing activities.

In developing the conceptual framework, I draw heavily from the principles and conceptual tools provided by activity theory and developmental work research (Engeström, 1987; Engeström, 1990; Engeström, 1991; Kuutti, 1990; Nardi, 1996) and the philosophy behind them (Ilyenkov, 1977; Septulin, 1977). In addition to activity theory and developmental work research, the conceptual framework has been inspired at least by Grudin's (1996) notion of organizational contexts of development and use, the notion of task-artifact cycle of Carroll et al. (1991), International Organization for Standardization's (ISO, 1997a; ISO, 1997b) notion of context of use as well as Interpretive Systemology (Fuenmayor, 1985; Lopez-Garay, 1987; Fuenmayor & Lopez-Garay, 1991).

In this paper, I will primarily introduce my conceptual framework. Comparing and contrasting it with other frameworks as well as case studies illustrating the use of the framework, though important and worth doing, are out of the scope of the paper.

The paper is organized as follows. First, I will identify some guiding principles that I try to follow in developing the conceptual framework. Second, I will introduce the central concepts of context and human activity system. Next, I will identify the relevant contexts of a product, the role of the product in each context as well as the relations between the contexts. The primary focus is in the context of development. Finally, I will draw some conclusions.

Guiding principles

Before starting to develop the conceptual framework, a few words about the underlying philosophical positions and guiding principles.

There are no separate things, there are only things in relation to other things, i.e., in context. Separation and connection are the two sides of any relation. In some relations, separation dominates, whereas in some other relations, connection dominates. Holism instead of atomism.

What a thing eventually is, is determined by the role it plays in the whole it is part of. In different wholes, a thing may play different roles. The thing may play various roles in the same whole, too, i.e., the role played by the thing is not always the same but may change. The thing and the context determine each other.

Consequently, a thing should not be studied or designed separately as itself but in relation to other things, i.e., in context. This takes place by identifying and making explicit the relevant contexts of a thing (wholes the thing is part of), the role of the thing in each context as well as relations between the contexts. Expansionism instead of reductionism (Fuenmayor, 1985:29).

My aim in developing the conceptual framework is that it should be a systemic whole or a holistic system. Dialectics instead of eclectics.

In presenting the conceptual framework, I will use various types of models to make my point.

I have tried to follow these guiding principles in developing the conceptual framework.

Concepts of context and human activity system

If the concept of context is the springboard for understanding things, one may ask how to get grasp of the context? Engeström (1990:77) discusses the concept of context and takes the position that, for activity theory, contexts are human activity systems. Furthermore, he presents a model of the structure of a human activity system, too (Figure 1).

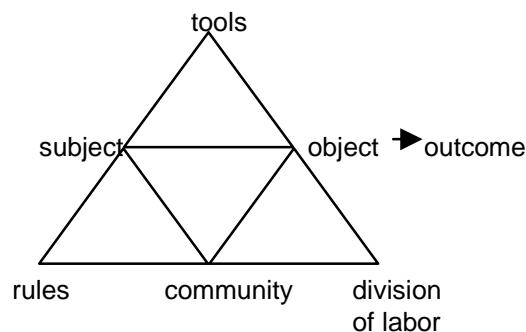


Figure 1: A model of the structure of a human activity system (Engeström, 1990:79).

Engeström (1990:79) describes the model of the basic structure of a human activity system as follows:

"In the model, the subject refers to the individual or sub-group whose agency is chosen as the point of view in the analysis. The object refers to the 'raw material' or 'problem space' at which the activity is directed and which is molded or transformed into outcomes with the help of physical and symbolic, external and internal tools (mediating instruments and signs). The community comprises multiple individuals and/or sub-groups who share the same general object. The division of labor refers to both the horizontal division of tasks between the members of the community and to the vertical division of power and status. Finally the rules refer to the explicit and implicit regulations, norms and conventions that constrain actions and interactions within the activity system."

Contexts of a product: Use value and exchange value side related

What are the relevant contexts of a software product? What could be the basis for identifying the contexts of the product, the role of the product in each context as well as relations between the contexts?

First of all, in our socio-economic formation, i.e., capitalism, all things, activities and relations have two sides: use value side and exchange value side (Engeström, 1987:84). Software products are no exception. On one hand, they can be used to achieve something useful. On the other hand, they have a certain value in terms of money, i.e.,

price. This is the first distinction I will use in structuring the identification of the relevant contexts of software products.

What are the use value side related contexts of a software product? In identifying the use value side related contexts, I have used Engeström's model of the structure of a human activity system (Figure 1).

Any thing may play two roles as part of human activity. It is an object or it is a tool (Engeström & Escalante, 1996:361). This means that a software product may play the role of an object or a tool in human activity. The product is primarily an object in the context of development whereas it is (or, at least it should be) primarily a tool in the context of use. In the context of implementation, the product is transformed from an object to a tool. As the result we end up to the model depicted in Figure 2.

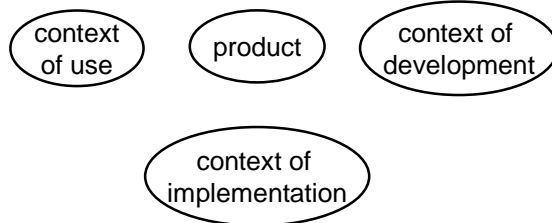


Figure 2: Use value side contexts of a product.

What are the exchange value side related contexts of a software product? They are context of buying and context of selling (Figure 3).



Figure 3: Exchange value side contexts of a product.

All the contexts of a product are in relation to each other and form a systemic whole shaping each other. Exchange value side concerns influence the use value side contexts and vice versa.

From now on, I will focus only on the use value side related contexts of a software product. However, it is useful to understand the existence of exchange value side related contexts, too.

Relations between use value side related contexts

How are context of development, context of implementation and context of use of a product related?

If we think of the temporal relation between them, the product is first developed, then it is implemented and finally it is used (Figure 4). To be able to use a product it has to be implemented and to be able to implement a product it has to be developed. This cycle is usually repeated many times as new development ideas are identified.

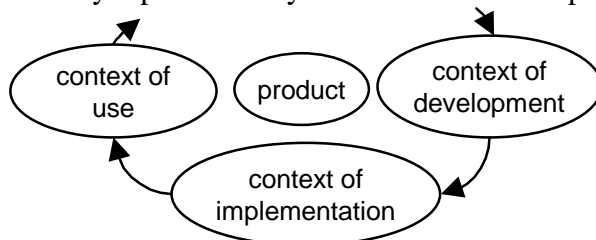


Figure 4: Temporal relation between context of development, context of

implementation and context of use.

We may think the relation between the contexts also in terms of which one of them is the primary, determinative one to which all other are subordinate. The obvious answer is that context of use is the primary one to which context of implementation and context of development are subordinate. Context of development is subordinate to the context of implementation, too. Context of use and context of implementation set requirements and constraints for the context of development.

Context of development

The primary focus of this paper is the context of development of a product, i.e., product development activity. What are the object, process and structure of product development activity?

Object of product development

Often, only the product itself is seen as the object of product development. My proposal is that not only the product itself but the product in its context of use is the object of product development (Figure 5).

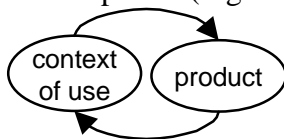


Figure 5: A model of the object of product development.

The basic idea is that the product and its context of use determine each other, i.e., they imply and follow from each other. Why the product is as it is can only be understood in the light of its context of use that has been in mind when developing it. This means that it is not sensible to talk about them separately. And it is not even possible. When you say something about one, you at the same time make assumptions about the other, whether you want or not. What you can do is make them both explicit and study and design them together.

Process of product development

If we accept that the product in its context of use, and not only the product itself, is the object of product development, what implications does it have for the process of product development? My proposal is the following. The process should start by studying the present activity and present tools. This is followed by the development of a model of the new activity and new tools that corresponds with the model. Finally, the model of the new activity and new tools are implemented to test whether the model of the new activity and new tools really correspond with each other (Figure 6).

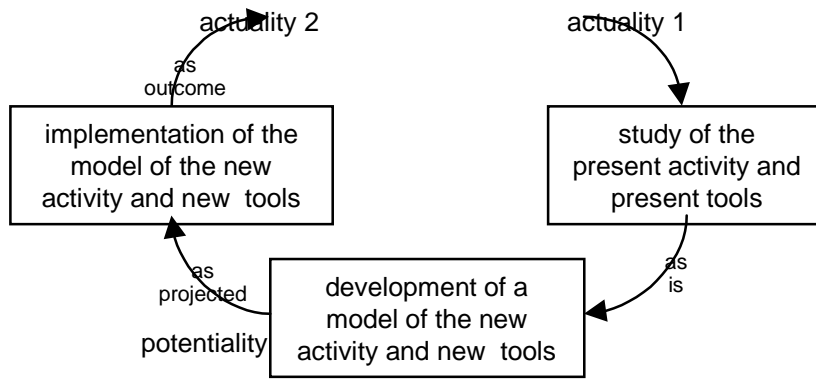


Figure 6: A model of the process of product development.

If only the product itself is seen to be the object of product development, the first and last phases are usually neglected. In addition, in the second phase, only the product itself is usually under design. Context of use is not under conscious design. If the product is anchored on something, it is anchored on user requirements, not on the context of use.

Structure of product development

Based on the discussion above, the model of the structure of a human activity system (Figure 1) can be expanded as shown in Figure 7.

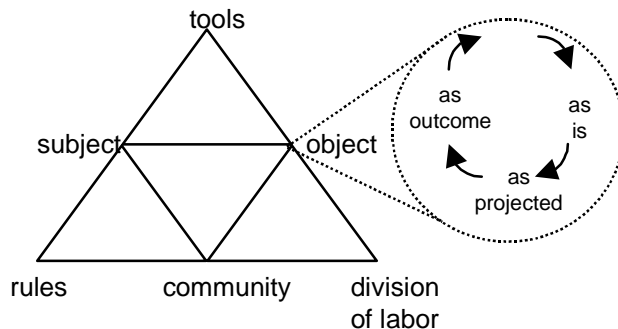


Figure 7: A model of the structure of the product development activity system.

The main difference compared to the original model is that the main transformations of the object are shown. The object is transformed in three steps, in line with the model of the process of product development (Figure 6). After the first phase, we have a description of the present activity and present tools (object as-is). After the second phase, we have a model of the new activity and new tools (object as-projected). After the third phase, the fit between the model of the new activity and new tools has been tested and we have a description of the resulting new activity and new tools (object as-outcome).

Object of product development: Product expanded

In software products, it is possible to make a distinction between the external and internal sides of the product (Figure 8).

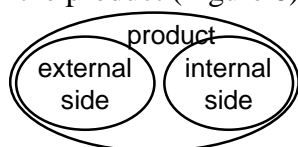


Figure 8: Product: External side and internal side.

External side includes all that the user has access to. Internal side includes the software constructs that make the product to behave as it does.

Many product development approaches do not make a clear distinction between the external and internal sides of the product. It is true that these are in practice closely interrelated but it is useful to make a distinction between them. Otherwise, there is a danger that, and that happens often, the external side of the product somehow emerges as a by-product of the design of the internal side of the product.

When Figures 5 and 8 are combined, we end up with the following model of the object of product development (Figure 9).

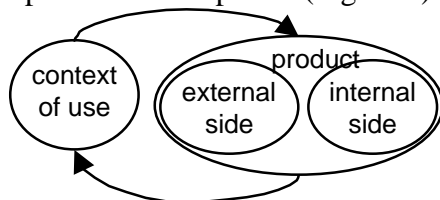


Figure 9: Context of use of the product vs. external and internal sides of the product.

Two transformations of product development

Based on the model of the object of product development presented in Figure 9, two transformations of product development may be identified (Figure 10).

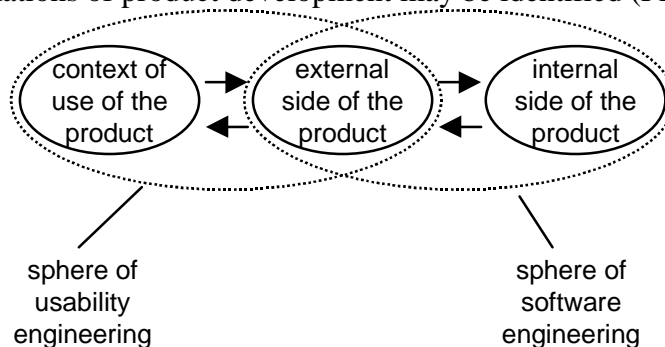


Figure 10: The two transformations of product development.

The first one is the transformation between the context of use of the product and the external side of the product. One has to ensure that these correspond with each other. This is the sphere of usability engineering.

The second transformation is between the external side of the product and the internal side of the product. After the external side of the product has been specified, software meeting the specification is developed. This is the sphere of software engineering.

In practice, the spheres of usability engineering and software engineering are closely interrelated. They are two sides or streams of one and the same process rather than two separate processes (Figure 11). They go on all the time parallel to each other. However, which one of them dominates varies depending on the phase of product development.

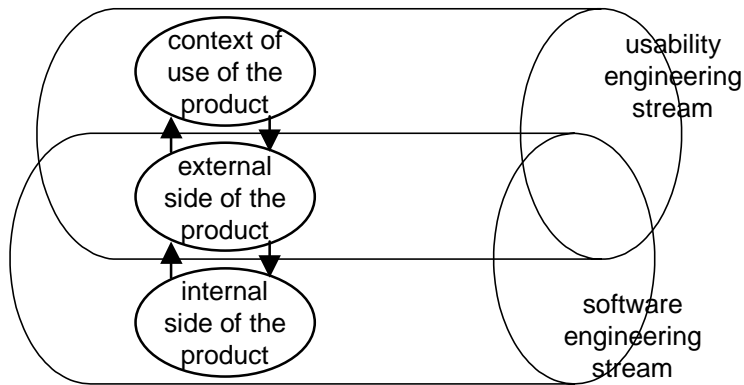


Figure 11: The two streams of product development.

Usability: Correspondence between the product and its context of use

The presented model of the object of product development (Figures 5 and 9) can be used in defining usability, too. Usability is correspondence between the product and its context of use. This definition of usability has certain advantages over many other definitions.

First of all, it shows and admits that usability is a relative thing. A product that is usable in a particular context of use may not be usable in some other. There is no absolutely usable product.

Second, it shows the different channels to influence product usability. Product usability can be influenced by changing the product (through product development) or by changing its context of use (through training or changing the ways of working).

Third, it shows different objects of study and design and the methods needed. From the usability viewpoint, the objects are: context of use of the product, external side of the product, and the relation between them. Consequently, there is a need for methods for studying and designing the context of use of the product, for designing the external side of the product as well as for testing the fit between the product and its context of use.

Fourth, it is a white-box (proactive, developmental, formative), not black-box (reactive, summative), approach. It helps in developing a usable product instead of only helping to assess the usability of a product after it has been already developed.

How to get grasp of the product in its context of use?

To get grasp of the product in its context of use, conceptual tools for understanding human activity and the role of artifacts as part of it are needed. The model of the structure of a human activity system (Figure 1) can be used for that purpose. As Figure 12 shows, it is in fact an expansion of the model of the product in its context of use (Figure 5).

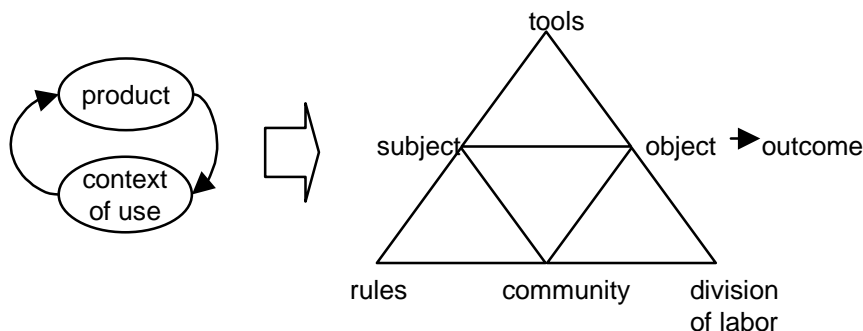


Figure 12: Model of the structure of a human activity system as an expansion of the model of the product in its context of use.

Relations between use value side contexts reconsidered

In Figure 6, a model of the process of product development was provided. If we compare it with the model of the relation between context of development, context of implementation and context of use depicted in Figure 4, we can see that they resemble each other. If we rotate all the components of the model in Figure 4 one step clockwise (Figure 13), we, in fact, end up in the same model as in Figure 6. The same model can be used to describe the process of product implementation, too.

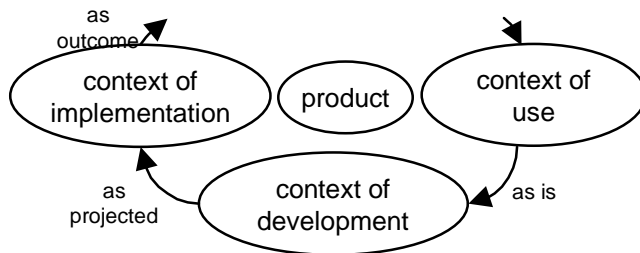


Figure 13: Process of product development and product implementation.

The same phases should be lived through during product development and product implementation. However, the reason for doing this is different.

The goal of product development is to develop a useful and usable product. All activities serve primarily this purpose. In connection with this, process innovations may be done as a side-effect of product innovations.

The goal of product implementation, in turn, is to develop the activity the product is supposed to support. All activities serve primarily this purpose. In connection with this, product innovations may be done as a side-effect of process innovations.

To summarize, the object and process of product development and product implementation are the same. The object is the product in its context of use. The process starts by studying the present activity and present tools. This is followed by the development of a model of the new activity and new tools that corresponds with the model. Finally, the model of the new activity and new tools are implemented. The difference is that during product development the product is primarily under transformation whereas during product implementation the context of use of the product, i.e., the activity or process, is primarily under transformation. If product development is production, product implementation is reproduction. Figure 14 summarizes the discussion above by expanding the context of development and context of implementation components of Figure 4.

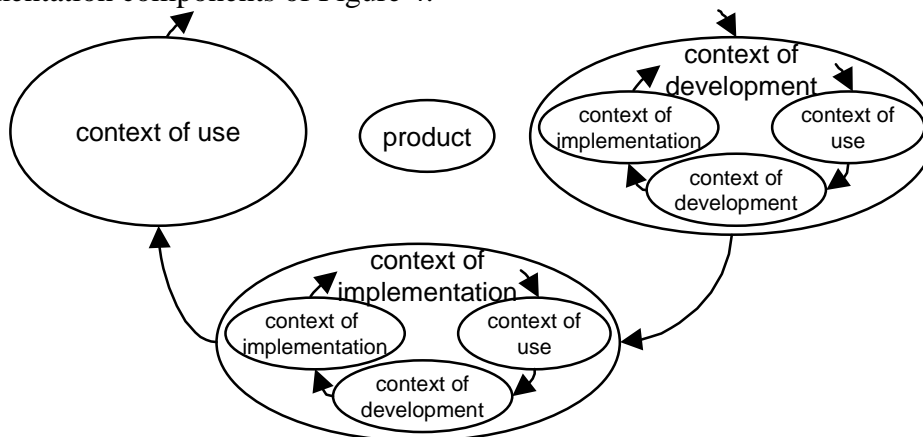


Figure 14: Temporal relation between context of development, context of implementation, and context of use.

implementation and context of use: Context of development and context of implementation expanded.

Conclusions

In this paper, I put forward a conceptual framework for understanding software product development.

My approach was the following. I first put product development in a broader context by identifying all the relevant contexts of a product and the relations between them. After that, I discussed product development in this broader context, in relation to other contexts of a product, as part of a whole.

The main conclusion was that the object of product development activity is not only the product itself but the product in its context of use. Implications of the broadening object for the process of product development were drawn. A definition of usability was given and the relation between usability engineering and software engineering was discussed. Finally, the relation between product development and product implementation activities was elaborated.

An alternative approach would have been to start to study directly product development, to treat it as a whole which is then divided into parts.

My approach is a manifestation of expansionism whereas the alternative approach would have been a manifestation of reductionism. The assumption behind reductionism is that separation dominates in relations between things, whereas the assumption behind expansionism is that connection dominates.

Today, it is often said that the world is coming more and more complex. I believe a lot of the perceived complexity is self-caused because of the dominating reductionistic approach. If two things are first separated and later tried to relate to each other it may not succeed. The philosophical mind-body problem is an example of this. To be able solve the problem, connection should be assumed from the beginning instead of separation. The relation between the product and its context of use discussed in this paper is comparable to the mind-body problem. To get forward, we have first to take a few steps backward.

References

- Carroll, J. M., Kellogg, W. A. & Rosson, M. B. (1991). The task-artifact cycle. In: Carroll, J. M. (ed.) *Designing interaction. Psychology at the human-computer interface*. New York: Cambridge University Press. Pp. 74-102.
- Engeström, Y. (1987). *Learning by expanding: An activity-theoretical approach to developmental research*. Helsinki: Orienta-konsultit.
- Engeström, Y. (1990). *Learning, working and imagining: Twelve studies in activity theory*. Helsinki: Orienta-konsultit.
- Engeström, Y. (1991). Developmental work research: Reconstructing expertise through expansive learning. In: Nurminen, M. I., Järvinen, P. & Weir, G. (eds.) *Precedings of the conference on Human jobs and computer interfaces, Tampere, June 26-28, 1991*. University of Tampere, Finland. s. 124-143.
- Engeström, Y. & Escalante, V. (1996). Mundane tool or object of affection? The rise and fall of the Postal Buddy. In: Nardi, B. A. (ed.) *Context and consciousness. Activity theory and human-computer interaction*. Cambridge, MA: MIT Press. Pp. 325-373.

- Fuenmayor, R. (1985). The ontology and epistemology of a systems approach. Ph.D thesis. University of Lancaster. 430 pp.
- Fuenmayor, R. & Lopez-Garay, H. (1991). The scene for interpretive systemology. *Systems Practice*, 4, 5, pp. 401-418.
- Grudin, J. (1996). The organizational contexts of development and use. *ACM Computing Surveys*, 28, 1, pp. 169 - 171.
- Ilyenkov, E. V. (1977). *Dialectical logic. Essays on its history and theory*. Moscow: Progress Publishers. 372 pp.
- ISO (1997a). ISO/FDIS 9241-11:1997(E) Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability. International Organization for Standardization. 22 pp.
- ISO (1997b). ISO/DIS 13407:1997 Human-centred design processes for interactive systems. International Organization for Standardization. 32 pp.
- Kuutti, K. (1990). The activity theory as an alternative framework for HCI research and design. In: *European Conference on Cognitive Ergonomics (ECCE-5)*. Urbino, Italy, September 3-5, 1990. European Association of Cognitive Ergonomics. Pp. 347-358.
- Lopez-Garay, H. (1987). A holistic, interpretive concept of systems design. Ph.D thesis. University of Pennsylvania. 633 pp.
- Nardi, B. A. (ed.) (1996). *Context and consciousness: Activity theory and human-computer interaction*. Cambridge, MA: MIT Press. 400 pp.
- Septulin, A. (1977). *Dialektiikan kategoriat ja lait (The categories and laws of dialectics)*. Moskova: Edistys. 344 pp. (In Finnish)