

# Conceptual Change in System Understanding: Towards Creative Software Design

R. Oehlmann<sup>1</sup> and K. Shima<sup>2</sup>

<sup>1</sup>Kingston University  
School of Computing  
and Information Systems  
Kingston upon Thames, KT1 2EE, UK  
R.Oehlmann@kingston.ac.uk

<sup>2</sup>Nara Institute of Science and Technology  
8916-5 Takayama, Ikoma,  
Nara, 630-0101, Japan  
+81-743-72-5312  
shima@computer.org

## Abstract

This paper views conceptual change as an element of creative software design. It argues that the process of changing conceptualizations is socially constructed and emphasizes the social origins of knowledge. In support of this view, results from psychology and artificial intelligence are presented. The psychological studies consider conceptualizations as mental representations that describe partial knowledge of a task domain (task conceptualization) as well as knowledge about other participants (social conceptualization). The findings indicate some limitations of conceptual change in collaborative settings and how to overcome these by externalizing models of the social context. Based on these studies a computational model has been implemented that simulates the limitations as well as a method of externalizing social models. The results suggest that the constraints imposed on conceptual change are caused by the process of social concept construction and the relaxation of these constraints are facilitated by externalization of social models.

## Keywords

Software Engineering, Conceptual Change, Creativity, Social Models, Collaborative Workgroups

## 1. Introduction

It has been generally recognized that the software developer has to consider not just the software product itself but the entire socio-economic system in which the software will be developed and deployed. Therefore the perspectives or conceptualisations of the various stakeholders have to be taken into account. Stakeholders have a vested interest but often only a partial understanding of the system and the problem the software development is addressing [5]. The term conceptualization is here used to describe mental representations that represent partial knowledge of a task domain, including causal links between knowledge components (task conceptualization), and representations about other participants (social conceptualization) including knowledge about how these participants might see the world. Negotiating the different conceptualisations may lead to a change of the developers initial conceptualization. Note that in the literature this phenomenon has been described by using various terms, such as changing the viewpoint [11,18], perspective change [9], reconceptualization [12,13], and concept articulation [15] have been used. We use here the terms conceptualization and conceptual change that originated from developmental psychology to subsume all these terms acknowledging that the relation between the various terms still requires clarification. If the conceptual change is suitable,

such change may lead to an improved design. It has been argued that conceptual change may establish creative design solutions [8,9].

Investigations of task conceptualizations have mainly considered an individual reasoner rather than groups and have focused on the role of conceptualizations in creativity and problem solving. Selective encoding and selective constructions are among those processes that have been considered as crucial for creative insight [17]. Selective encoding involves distinguishing between relevant and irrelevant information and hence enabling a conceptualization. Selective combination is concerned with the synthesis of isolated pieces to a unified whole. Often the same pieces can be combined and recombined in different ways and establish different conceptualizations.

Ohlsson [12,13] has argued that an impasse can occur on a problem even if the problem solver is competent to solve it. This happens because the initial encoding of the problem does not activate the relevant competence in memory. Constructing a different representation of the problem can break the impasse. If the new representation is sufficiently different from the initial representation, memory activation will involve different memory components. The impasse is broken by changing the initial conceptualization. This is consistent with views put forward by Gestalt psychologists who described this process as restructuring [12, 16]. In addition to psychological studies, conceptualizations have been utilized to guide a student's learning process. For instance, Ackermann [1] interpreted conceptualisation as constructionist activity. Following Piaget, she argued that people actively construct knowledge rather than passively taking up preformed concepts. Therefore in her view people are able to construct new conceptualisations as well.

More recently, conceptualisations have been discussed in creativity research. Sugimoto, Hori, and Ohsuga [18] described a system that supported creative insight by visualizing different conceptualisations referred to as views. Shoji and Hori [15] investigated conceptualizations and conceptual change by observing sales interactions related to consumer goods. In their investigation they used the notions of *concept articulation* and *communication patterns*. Concept articulation describes a process of refining initially vague purchasing requirements based on information provided in the actual purchasing situation. This could be considered as a refinement of conceptualizations. Communication patterns include a salesperson's unexpected reactions that provide a customer with a different viewpoint. If this viewpoint is adopted, the customer has performed a conceptual change. Based on these investigations, Shoji and Hori developed the S-Conart system that supports concept articulation.

All these approaches consider a single reasoner who as part of the reasoning process performs conceptual change. Unfortunately, it did not take into account that people live in a social context, which affects the way they think about the world around them. In contrast, social-constructionists argue that people perceive the world in the way they do because they participate in socially shared practices and interact with the world in terms of meaning systems which are simultaneously transmitted, reproduced and transformed in direct and symbolic social interchanges [4]. This applies particularly for software development processes, because the tasks usually are too complex to be performed by individual developers. In development teams, engineering problems and their solutions become meaningful as they are constructed during interactions with team members as well as with other stakeholders such as clients.

The concept of social conceptualisation is based on Mead's [7] suggestion that we are able to take the perspective of other people, to imagine how they see the world, and to anticipate and explain their behaviour. It also has been noted that children are already able to take the perspective of others (see the comment in [10]). In contrast, autistic children often lack this ability [2]. Therefore taking the perspective of others appears to be an essential element of normal social interaction.

The group aspect has particularly been considered in work on computer-supported design. Nakakoji et al. [8] used the concept of collective creativity to argue that intelligence and creativity depend on collective memories rather than an "unaided, individual mind" ([8], p. 452). In a similar spirit, Fisher [5] proposed the term social creativity recognizing to some extent the social dimension of creative activities. For instance, complex design projects cannot be completed by single designers, but have to rely on the knowledge and the experience of many "stakeholders" who have a vested interest in the project. Therefore systems that support social creativity are required to overcome the discrepancies between the perspectives of different designers [5]. Knowledge creation in the context of social creativity requires the externalization of tacit knowledge [14] cited in [5]. Fisher lists the following reasons for the need for externalizing knowledge: the shift from vague conceptualizations to concrete representations, enabling stakeholders to negotiate knowledge, and to develop a common language for such interactions. Furthermore, such externalization may aid focussing the discussion ([3] cited in [5]). The concepts of collective creativity and social creativity rightly focus on complex design tasks, which require the collective design experience of many designers. In this sense they extend Ackerman's constructionist view that was just based on the individual reasoner's interaction with the world. However, the study described in this paper suggest that these approaches should be complemented by emphasizing the interactions between different stakeholders. This should include social perspectives, which focus on different participants who are involved in the task. The remainder of this paper will first summarize our psychological studies of individual and collective conceptual change. This will be followed by a description of a collaborative software architecture that simulates the social-constructionist aspect identified in the results of the psychological studies. The paper is completed with a discussion of this approach that will indicate some future applications of this result in areas such as design, creativity and software engineering.

## 2. Social Construction of Conceptual Change

Next we will contrast psychological results related to individual conceptual change with those related to collaborative

conceptual change, to reveal the social-constructionist aspects of the reasoning process. These results were obtained when participants attempted to understand the mechanism of a mechanical clock. Therefore first a description of this task domain is given.

A mechanical clock was used after the case was removed so that subjects were able to observe every single wheel of the clock. The clock was equipped with two springs, one for the chime and one for the actual clock mechanism. Figure 1 shows the structure of the chime (Chime), the mechanism for moving the hands of the clock (Clock), and the components of the clock, which connect the chime and this mechanism (Chime Control). The wheels of the clock are represented by circles, while the hammer of the chime, the hands of the clock, and levers are represented by rectangles. Arrows are used to indicate the transfer of force between wheels or between levers and wheels. The pin-wheel is crucial for the conceptual change described below and is therefore depicted by a shadowed circle. The sequence of wheels that control the chime can be viewed as a branched path. One branch leads from a *spring* to the *hammer*, whereas the other branch leads to a vertical wheel (pin-wheel) with a horizontal pin. A lever in contact with the pin stops the wheel. This lever is released when the large hand of the clock points upward (12 o'clock position).

Similarly, the arrangement of the wheels in the actual clock mechanism can be viewed as branched path. One branch leads directly from the spring to the hands of the clock, whereas the other branch originates at the large-hand wheel and leads to the ratchet.

In a first study, 10 subjects were separately presented with the same mechanical clock after the case of the clock was removed. Verbal protocols of all interactions with the clock were recorded. In three stages, subjects had to understand the chime of the clock, the mechanism for moving the hands, and finally the part of the lock that connects hand movement and chime.

The protocols revealed that participants attempted to identify a chain of consequences responsible for the hammer movement. In the course of these considerations they performed "experiments". They set the large hand in the 12:00 position to start the chime. All subjects started with the conceptualization that the spring drives the hammer only at certain times. 8 participants discovered the wheel/lever arrangement and performed a conceptual change. During the experiment, these participants observed that the two-wing wheel was turning and noted that a lever was moved into the orbit of the pin-wheel. The lever touched the pin and interrupted the movement of the pin-wheel. The unexpected observation led to a conceptual change. Rather than reasoning about the consequences of the movement of the pin-wheel, the 8 participants focused now on the consequences of the interruption of these movements. Subsequently, they considered the mechanism from the perspective that the spring would drive the hammer all the time and only the lever that is in contact with the pin stops the mechanism from doing so. Next, a study was conducted to investigate whether the ability to perform a conceptual change can be found in collaborative settings as well. 10 pairs of participants were presented with the same mechanical clock. Verbal protocols of all interactions with the clock were recorded. Again participants had to understand the three parts of the clock in three subsequent stages.

Under these conditions, none of the participants in this collaborative setting changed the initial conceptualization. All the protocols have the same structure as the protocols of those participants in the first experiment who did not change their perspective.

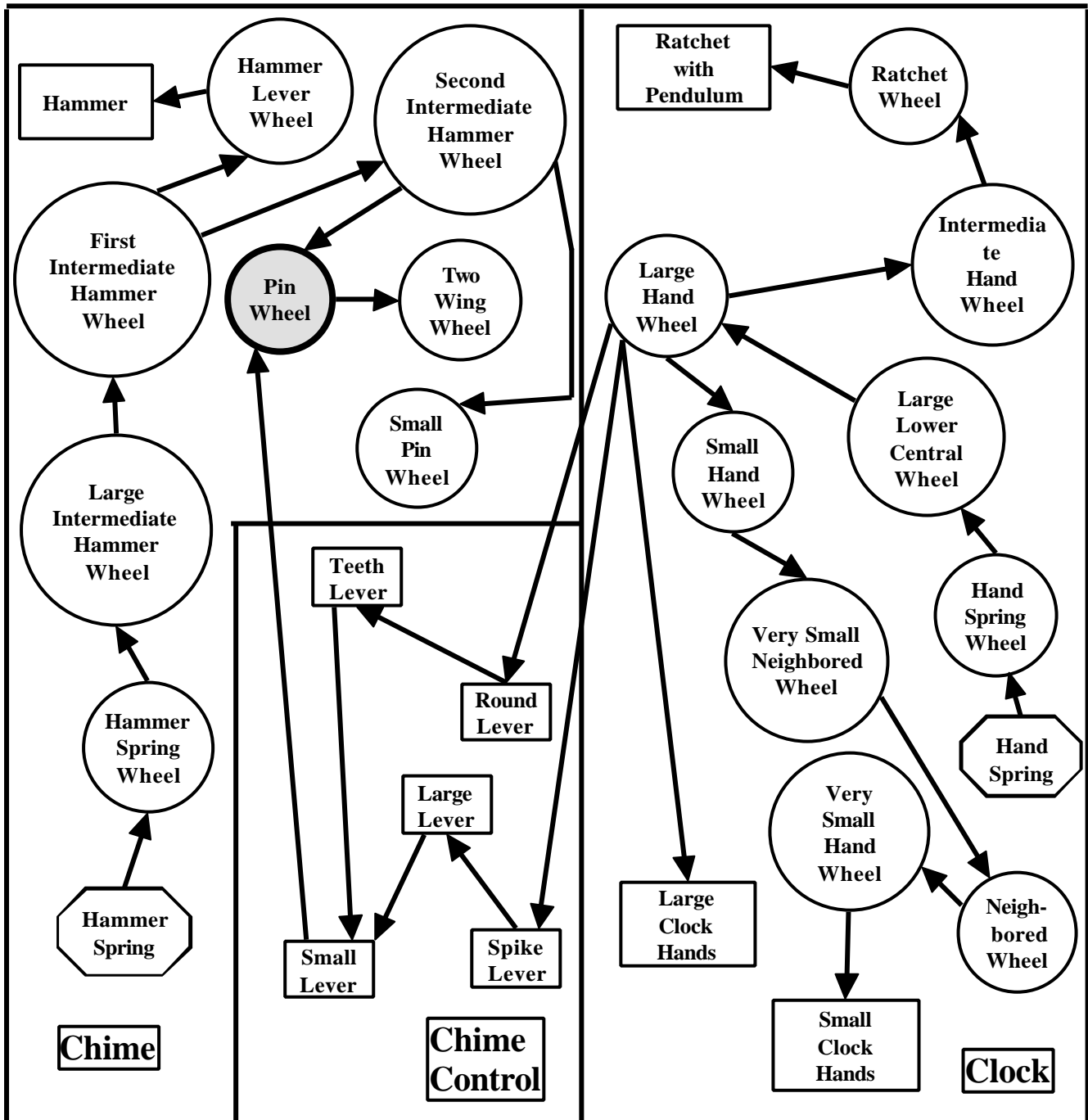


Figure 1: The Mechanical Clock Domain

A possible explanation of this result is based on the concept of cognitive overload, which means that the cognitive task demands are too high given the limited process resources of the human brain. Participants had to form models of the clock behaviour at the same time when they had to form models about the behaviour of the other participant. This might constitute too high cognitive demands. Therefore the third experiment investigated whether conceptual change would be affected by an externalisation of the models maintained by the participants. The experiment was conducted in two stages. One day prior to the experiment, participants received training in diagrammatic representation of conceptualizations. This included a given set of graphical objects to represent social as well as task conceptualizations. The training focussed on mapping perceived conceptualizations into sets of abstract diagrammatic elements such as circles, rectangles, triangles, or irregular shapes and their associated meaning. The second stage included essentially

the same procedure as in the second experiment. However, each member of a pair –group was allowed to call a break at any time to work on their graphical representations of conceptualizations. They were encouraged to exchange their representations with each other. However, if they wished to keep a given representation private, they could choose to do so.

In contrast to the second experiment, eight pairs of participants changed the perspective in the third experiment. Participants in this experiment followed the same path from the initial conceptualization to the changed conceptualization as participants in the first experiment. Again a conceptual change was achieved by noting the contradiction between an observation and the underlying task conceptualization.

### 3. Modelling Socially-Constructed Conceptual Change

Based on the results given in the previous section, the implementation of a software architecture, referred to as COLLABORATOR, will be described. The system simulates the socially constructed conceptual change and its constraints. The system consists of two separate agents that interact with one another. Following the description of the components of a single agent, the interaction between two agents is addressed.

Protocols obtained in the experiments described above have been simulated in the computer model. The program uses planning of sentences such as questions, answers, and factual statements to model reasoning. In addition, it is capable of planning experiments. The main task of the system is the generation of new explanations to revise an initial theory. Figure 2 shows the main modules of the system: *sentence planner*, *sentence strategy planner*, *experiment planner*, *introspection planner*, *model planner*, and *model scratch pad*. All these modules are implemented as case-based planners [6]. Ellipses in Figure 2 represent data in the form of input, output or stored data structures, whereas modular processes are shown as boxes. Thin arrows indicate data transfer and thick arrows represent the control structure in the form of process calls. Double arrows suggest that a process calls a second process that returns control to the calling process after it is executed.

The sentence planner module accepts a problem description as input and generates a protocol to address the input problem. Depending on the input, it has several options such as generating a hypothetical statement or a question about the problem followed by an answer. Often answers and factual statements are generated after performing experiments and using the obtained information for the answer. Sometimes a statement cannot be generated because information is missing. Also this situation is addressed by performing additional experiments. Generally, the generation of sequences of sentences is controlled by goals, i.e., goals are used to retrieve a sentence plan and the execution of the plan generates new goals. Before an experiment is performed, the sentence planner generates sentences that hypothesize the experimental result. The hypothesis is stored in the form of objects and relations between these objects. When the actual result is generated, an expectation failure is stated as the difference between the hypothesis and the result. Such an expectation failure is addressed by generating additional sentences and experiments. During this process of sentence-based reasoning and experiment-related activity, sentences focus on objects of the domain to be investigated such as pin-wheel and small lever (Figure 1). In addition, the sentence planner generates sentences that focus on the system's reasoning process. This type of sentence generation requires additional meta-knowledge that is not available to the sentence planner. Acquiring this knowledge is the task of the introspection planner [11].

An introspective sentence refers to the reasoner's internal knowledge and internal processes. A sentence plan that has to generate an introspective sentence contains of several steps. Executing such a step results in a call of the introspection planner. This planning process provides the information needed by the sentence planner, which then can complete the sentence. Introspection plans address different metacognitive tasks such as assessing goals, reasoning strategies, resources needed to perform a given reasoning strategy, reasoning failures that occurred previously, and conditions that have to be satisfied in order that a strategy can be executed. The basic knowledge structures of the system are experiments and plans that are used as cases. An experiment consists of two components: an experimental setting (e.g. a description of a mechanical clock with wheels, springs, and hands) and the result and the

experimental result such as the statement that "the hammer of the chime is moving when the large hand is in upright position." Experiments are represented by objects and relations between objects. Objects are represented as *memory units*, which contain an *object frame*, a *content frame*, and a *context frame*. The *object frame* contains general information about the object. The *content frame* stores several sets of intentional descriptor values referred to as *view*, whereas relations with other objects are described in the *context frame*.

Sentence plans are used to apply case-based planning techniques to the generation of single sentences. For instance, the question "What does the PIN-WHEEL turn?" can be built by combining the substructures "What", "does", "the OBJECT1", and "turn". OBJECT1 is a variable which is instantiated with the string "PIN-WHEEL" during plan execution. A question plan has two main parts: a set of descriptors used for indexing the plan and a sequence of steps. A plan is retrieved by matching its index with the current situation; this is characterized by goals the system pursues in asking the question. If a plan execution fails, the usual explanation-based repair mechanism is employed [6]. It is an important advantage of the case-based planning approach that new sentences can be learned by modifying previous sentence plans.

The case-based planning approach to sentence generation is highly flexible because it only depends on the current situation and the goals the system is attempting to pursue. Moreover, new plans can be generated by adapting existing plans to new situations.

In addition, the system uses experimentation plans to perform experiments. Experimentation plans describe the steps, which have to be executed in order to perform an experiment. Executing an experimentation plan results in calls to a rule-based simulator that simulates the experiment and returns the experimental result. The experimental setting and the result of a plan execution are stored as new experiment. The same basic plan structure used for sentence plans has been employed for experimentation plans, although the index vocabulary differs.

The model planner is an extension of the approach to generating hypotheses. Similar to hypotheses, models in this context are net structures consisting of objects and relations between objects. In contrast to hypotheses, models have a wider scope in that they describe a more complex situation rather than just predicting the outcome of a given experiment. The model planner functions in a way similar to the sentence planner. However executing the steps of a model plan generates and stores the elements of a model rather than those of a sentence. Basically the same technique is used to generate models that describe the task domain and to generate models that describe the behaviour of other agents.

Whereas the first type of model can be generated based on experimental results and the knowledge about experimental plans, the second type of model is more difficult to generate because the first agent has no knowledge of the plans stored in the second agent. Therefore the generation of social models is based on the initial state before plan execution, the result of a plan execution, and the plan stored in the first agent that matches initial state and result of an experiment. If a match is found, goals, tasks, and emotions under which the matched plan is stored can be used to build the social model. If the matching operation fails, questions need to be generated, which retrieve information about intentions and emotions from the second agent. This approach can be summarized in the following algorithm.

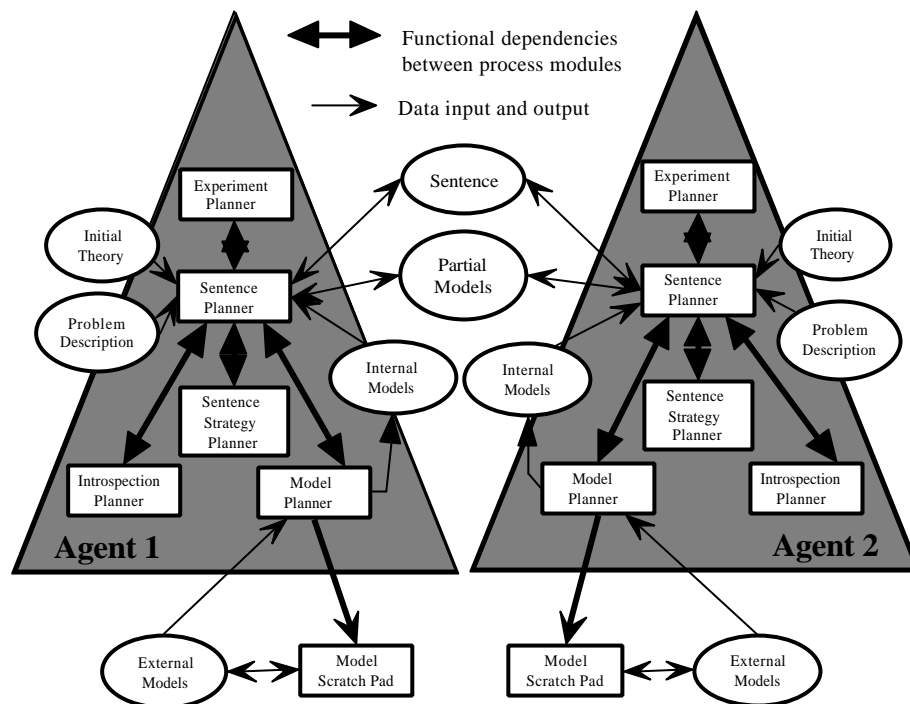


Figure 2: The COLLABORATOR Architecture

- Agent 1 attempts to assign own intentions to the observed behaviour by matching its observation (results generated by the simulator) with results stored in its plan library from previous plan executions.
  1. If the matching process fails
    - A. Agent 1 generates a question to ask Agent 2 about its intentions
    - B. Agent 1 uses the information given by Agent 2 and the observation of Agent 2's behaviour to enhance or modify the social model.
  2. If the matching process succeeds
    - A. Agent 1 enhances the social model using its own plan and the intentions stored in that plan.

Finally the *model scratch pad* is used to externalize models. This is again implemented as planner and therefore uses plans that are indexed by "intentions". It is important to note that storing a model in the scratch pad does not involve simple copying of an already existing model. It rather means to generate a new model that typically is more detailed and more explicit than the original model. This again reflects findings from the protocol analysis mentioned above. The generation of a model in the scratch pad may even lead to restructuring the internal model, because the more detailed view may include inconsistencies that requires changing the initial model.

Communication between agents mainly takes place on the level of exchanging sentences. In order to avoid the issue of parsing sentences and understanding natural language, which is beyond the scope of this approach, the agents exchange partial sentence plans. This means that the correct sequence of steps is transferred, but not the information about goals and task that is used to index the plan. Also partial models can be transferred between agents. This is considered as a simulation of producing a drawing and showing it to a collaborator, but it avoids the problem of image understanding.

#### 4. Discussion

In Section 2, we have summarized psychological results, which indicate that individuals may benefit from conceptual change, while reasoners in collaborative settings fail to do so. This limitation could be overcome by training participants in externalizing their conceptualizations. The behaviour has been simulated in the COLLABORATOR system. Tests of this architecture have shown that the system changes the reasoning perspective, only if the model scratchpad is used. The psychological findings suggested as one possible explanation that participants experience a cognitive overload which prohibits addressing task conceptualizations and social conceptualizations at the same time. However tests of the COLLABORATOR system indicate another explanation. Conceptual change is socially constructed; i.e. the conceptual change is derived from the interactions between participants. This interpretation reach beyond Ackerman's view of constructing conceptualizations in that it encompasses the social constraints imposed by collaborators. The social constructionist view requires one agent to evaluate the contributions of the other agent. In order to do so, the agents need to have available social models of their collaborators. Moreover, the COLLABORATOR system uses the generation of external models in the model scratch pad to restructure the internal task model. Therefore another possible explanation for the failure to achieve conceptual change might be the interleaving between the generation of task models and social models. These participants may just not have completed their social modelling and therefore could not address the conceptual change. However this hypothesis has to be tested in additional psychological experiments.

Among the areas of possible applications of the reported results are design, creativity tools, and project management. In addition to considering the collective design knowledge of the design community, designers need to consider the perspectives of those designers they collaborate with. In the early stages of managing software projects, such as requirements analysis and design, the

different perspectives presented by different stakeholder groups are of particular importance for the success of the project. Software systems that guide the externalization of social models may help to reach a consensus within the project team that would support the project goals in an optimized way.

Managing different conceptualizations is important for gaining new insights as well as for achieving consensus within development teams. The training of participants in externalizing conceptualizations has increased the tendency towards conceptual change. Looking at a problem by using different conceptualizations may lead to more creative solutions. Future computer-based tools for diagrammatic reasoning about the social context may achieve that.

## REFERENCES

- [ 1] E. Ackermann, "Perspective-Taking and Object Construction. Two Keys to Learning." In Y. Kaifai and M. Resnick (eds.), *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*, Chapter 2, Lawrence Erlbaum Associates, Hillsdale, NJ, 1966.
- [ 2] S. Baron-Cohen, A. Leslie, U. Frith, "Does the Autistic Child Have a "Theory of Mind"?", *Cognition*, vol. 21, pp. 37-46, 1987.
- [ 3] J. Bruner, "The Culture of Education", Harvard University Press, Cambridge, MA, 1996
- [ 4] V. Burr, "An Introduction to Social Constructionism", Routledge, London, 1995.
- [ 5] G. Fischer, "Symmetrie of Ignorance, Social Creativity, and Meta-Design". *Knowledge-Based Systems (Journal)*, Special Issue on Creativity and Cognition, 1999, vol. 13 2000, pp. 7-8.
- [ 6] K. Hammond, "Case-Based Planning", Academic Press, New York, 1989.
- [ 7] G. Mead, "Mind, Self and Society", University of Chicago, Chicago, IL, 1934.
- [ 8] K. Nakakoji, M. Ohira, and Y. Yamamoto, "Computational Support for Collective Creativity", *Knowledge-based Systems Journal*, Elsevier Science, Vol. 13, No. 7-8, 2000, pp. 451-458.
- [ 9] R. Oehlmann.: "Perspective Changes Affect Attentional Access to Conscious Experience". In K. Yasue (ed.), *Towards a Theory of Consciousness: Foundations and New Developments*, *Advances in Consciousness Research*, vol. 33, John Benjamins Publishers, 2002.
- [10] R. Oehlmann, "Can Metacognition be Explained in Terms of Perceptual Symbol Systems?" *Behavioural and Brain Sciences*, Vol. 22, No. 4, 1999, 629-630.
- [11] R. Oehlmann, *Metacognitive Attention: "Reasoning about Strategy Selection"*, In J. Moore and J. Lehman (eds.), *Proceedings of the 17th Annual Conference of the Cognitive Science Society*, Pittsburgh, PA., Lawrence Erlbaum Associates, Hillsdale, NJ, 1995, pp. 66-71.
- [12] St. Ohlson, "Restructuring Revisited. I. Summary and Critique of the Gestalt Theory of Restructuring and Insight", *Scandinavian Journal of Psychology*, Vol. 25, 1984, pp. 65-78
- [13] St. Ohlson, "Restructuring Revisited. II. An Information Processing Theory of Restructuring and Insight", *Scandinavian Journal of Psychology*, Vol. 25, 1984, pp. 117-129.
- [14] M. Polany, "The Tacit Dimension", Doubleday, Garden City, NY, 1966.
- [15] H. Shoji and K. Hori, "Creative Communication for Chance Discovery in Shopping", *New Generation Computing (Journal)*, Vol. 21, No. 1, pp. 73-86
- [16] H. Simon, "The Information-Processing Explanation of Gestalt Phenomena", *Computers in Human Behaviour*, 2, 1986, pp. 241-255.
- [17] R. Sternberg "A Three-Facet Model of Creativity". In R. Sternberg (ed.), *The Nature of Creativity*, Cambridge University Press, UK, 1988, pp. 125-147.
- [18] M. Sugimoto, K. Hori, and S. Ohsuga, "A System for Supporting Researcher's Creativity by Visualizing Different Viewpoints". *Workshop on Creativity and Cognition*, Loughborough, UK, 1996.