

To Whom Belongs Conceptual Design?

J. Bíla

The field of Conceptual Design is very alive and is rapidly developing. This paper investigates the disciplines and domains which substantially form its profile. There are considered disciplines such as Semiotics, Formal Logic, Evolutionary analogies, Qualitative Modelling, Ontologies, Artificial Intelligence and Emergent Synthesis. The answer to the question posed in the title lies nowadays in disciplines related to Cognitive Science.

Keywords: conceptual design, semiotics, ontologies, UML, artificial intelligence.

1 Introduction

Conceptual Design remains a very attractive field of research. There are two reasons for this: a free space for modelling of creativity, and the opportunity to apply novel means of Artificial Intelligence.

Preparing this paper, use was made of information from many sources of Conceptual Design Support and Conceptual Design Process (CDP) modelling (and we apologise to all which are not introduced in the References).

The essence of CDP near to the context of this paper is available, e.g., in [3, 4, 5, 19]. Attempts at constructing a deeper formal description of CDP were done, e.g., in [13, 20]. The means of verification of CDP results were suggested, e.g., in [1, 2, 7]. (In [11] there were discussed means of verification of SW products for design in the pre-implementation phase). Support for conceptual design of artefacts and ways for modelling creativity in CDP were presented, e.g., in [3, 9, 10, 13, 28, 29, 35]. The application of structural analysis methods in CDP were used, e.g., in [18] (Yourdon Structural Analysis) and in [21] (UML), [20] (OMT-UML). Special computer CDP support systems were presented, e.g., in [3, 6, 7, 8, 12, 15, 24].

2 Semiotics, formal logic and evolutionary models

The outcome of the process of Conceptual Design (in technological fields) is usually understood to be a *scheme*. The *scheme* has substantial features of a product or system which is designed but need not necessarily contain geometrical and quantitative data. Many Conceptual Design activities may be studied in the field of Semiotics, in a small field which is investigating the specific cognitive relations between the *signed* and the *signing*, and their reflective and cognitive functions. Comparing figures from Leonardo da Vinci's "notebook" with Olesen's figures from [19], we have to admit that the techniques of conceptual reasoning and of expressing conceptual ideas and their essence have not changed fundamentally in a period of 400 years. As conceptual designs there might be considered not only schemes of Leonardo's submarines but also the schemes of Dürer's figural compositions [14], the schemes of Michelangelo's constructions and also the scheme of Alessandro Marcello's (1686–1739) oboe concerto. In all these examples, the scheme lives within a *level of conceptualisation* which drives the understanding of the scheme. What is the purpose of the *scheme*? The *scheme* presents the form of the designed system and explains its function. How deep this

presentation and these explanations are depends on the ontology within the framework of which the *scheme* was formed.

Formal logic has tried to help in many aspects of conceptual design theory. This paper mentions only two:

- the formation and processing of concepts (Frege, Tichý, Materna),
- ways of transferring truth (Gödel, Gentzen, Robinson).

The first line of research established concepts such as specific structures, which are composed in conceptual constructions. The second line of research discovered the fact that the deeper semantics and pragmatics of conceptual constructions may be described by a formal logic system, by a system which is able to control what is possible, what is impossible and what is correct. (Using such a system we can stipulate that legs are not part of the head and that the wheels are not situated on the roofs of cars).

In conclusion, the contribution of Semiotics and Formal Logic to Conceptual Design is natural and not too sophisticated. Semiotics tries to grasp the process that takes place between the model and the brain (mind) of the designer during designing, but its formal means were adapted to a static investigation of language rather than design issues. The automatic mechanisms which assign sign formations to ideas during language phenomena are unfortunately not visible by the formal means of Semiotics. Formal Logic, however, helps Semiotics, but *post factum*.

Hypotheses investigating the "paths" of design ideas which are not necessarily mediated *through language* are fresh and alive nowadays. In this context, the research lines of BioSemiotics [9] and Evolutionary Analogies [33] which induce images about a more natural formation of artefacts are very interesting. Fig. 1 illustrates the interaction between Niche Space and Design Space, as introduced by Sloman in his "speculation" about Evolution [33].

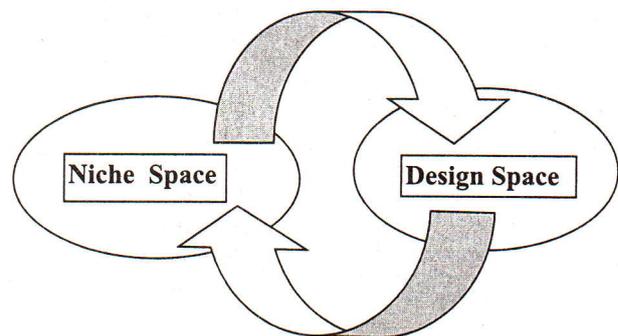


Fig. 1: Interaction between Niche Space and Design Space

Niches are the carriers of “requirements” in living elements which “may produce pressure for evolutionary change”. However, they are more than static lists of requirements. They also own reflective functions and a very useful complex of properties which Sloman called *affordance* (the ability to decide the actual role (of the niche)). In other words, affordance induces variants of functions. Though Sloman’s paper [33] considers real long time evolution, the analogy for CDP is very attractive, and an especially important feature is the process between the Niche Space and the Design Space, which is a process between *the signed* and *the signing*, a metaphorical image of a special co-evolution: “Possible designs and possible niches are linked by descriptions of ways in which different designs match a particular niche and the same design matches different niches. Since mismatches can produce pressures for changes in designs, and this can produce new niches, leading to new kind of matches and mismatches, we have interacting systems concurrently tracing trajectories through design space and through niche space with complex interacting feedback loops”, [33]. Sloman differentiated three essential types of feedback loops (Fig. 1): *i-loops* (individual learning and development), *e-loops* (evolutionary development) and *r-loops* (repair loops: an external agent replaces, repairs or adds a new feature. It may then jump to a new part of design space and niche space.)

The fragment of an evolutionary analogy introduced above contains mainly novel images and terms, it turns our attention to a process the goal of which is different from “correct understanding” of signs. In other words the communication aspects play a less important role (in the process illustrated in Fig. 1) than in the classical framework of Semiotics.

3 Concepts and qualitative modelling

This section will sketch a small “ontology” for work with concepts and related categories. After explaining that conceptual design is based on operations with intentions rather than with concepts, a short reasoning about a calculus for Conceptual Space will be presented.

First, it is necessary to differentiate between *concepts* and *intentions*.

Concepts are abstract ideal categories, and according to their use they belong to knowledge. From the procedural point of view they are *identification procedures* which identify objects. (Objects are used here as entities outside the subject.

There is no relation here to the Object Oriented approach.) Each concept has expression, content (substrate), structure (according to Bolzano) and meaning. A typical example of a concept is: “Primes” [36]. If we know this concept, we also know the procedure for identifying possible numbers as primes or not primes. What is important is that this procedure is our internal knowledge, we do not need any empirical facilities or external assistance. If, for the identification procedure, we need some empirical facilities and operations, e.g., for identifying the situation “actual temperature in block A”, we speak about *empirical concepts*. However, the results of identifying empirical concepts are not objects but *intentions*.

Intentions (as introduced, e.g., in [26, 36]) represent roles which may be played by objects. (E.g., “to be a support for”, “to be an engine of”, ...). *Intentions* are mappings from time-space states of possible worlds into a space of values (and they have no internal structure in general). It is important to emphasise that whilst the definition domain of these mappings is the same (time-space states of possible worlds, but depending of course on the actual instance of time-space states in a possible world), there are four *basic* spaces of values: “space of truth values”, “space of individuals”, “space of classes of individuals” and “space of numbers”. Corresponding with these value spaces there are four basic intentions: *propositions*, *offices*, *properties* and *quantities*. After this small excursion into the background of conceptual constructions it is not surprising that the categories by which we operate in Conceptual Design are more intentions than concepts.

Fig. 2 introduces a simple image of the evolution of a Conceptual Space (a space with concepts and intentions) and a Calculus (in our case a calculus for modelling and supporting CDP). A mental image from the external world is not necessarily the principal category in this scheme. Its main task is to navigate attention during the structuralisation of Conceptual Space. Of great importance are Semiotic Activities, which execute the relations between concepts and intentions and control their evolution. Experience with various formal means has shown that though there are some general relations between Conceptual Space and Calculus, there are strong limitations in the development methodologies, which must be avoided with the help of *additional domain knowledge*.

Empirical experience with operations involving intentions and concepts leads to qualitative objects, and as a

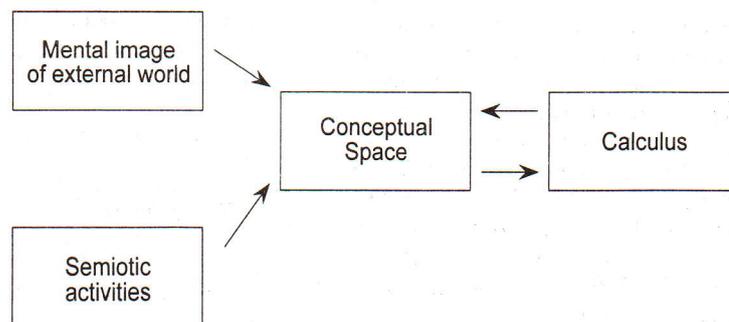


Fig. 2: Conceptual Space and Conceptual Calculus

consequence to qualitative operations. It is interesting to consider the nature of such operations (returning to the issue of calculus (from Fig. 2)).

(In order to avoid the necessity to specify when we are working with intentions and when with concepts (for each case) we will speak about members (x, y, z, \dots) of Conceptual Space (CS). These members do not represent expressions, contents, structures, ... of concepts, nor values of intentions, but they represent their special semantic content. They represent all consequences which are relevant to the specification of a Conceptual Design problem and which can be derived from the considered members.) Some basic implications are available even in a very weak structure of Conceptual Space

$$CS = \langle \Xi, \cup, \sim \rangle, \quad (A1)$$

where Ξ is a carrier of CS , \cup is a binary operation of CS member synthesis and \sim is a relation of strong similarity which is considered as a relation of tolerance. (For "not \sim " we use symbol " \neq ".) The following axioms corresponds to empirical experience:

$$\forall x \in \Xi, (x \cup x) \sim x, \quad (A1)$$

$$((\exists x, y \in \Xi) \text{AND}(x \neq y)) \Rightarrow ((x \cup y) \neq (y \cup x)), \quad (A2)$$

$$((\exists x, y, z \in \Xi) \text{AND}((w \neq y) \text{AND}(x \neq z))) \Rightarrow \\ \Rightarrow ((w \cup y) \neq (x \cup z)). \quad (A3)$$

From these axioms we can derive the following theorems:

T1: Operation \cup is not associative (with regard to relation \sim):

$$((\exists x, y, z \in \Xi) \text{AND}((x \neq y) \text{AND}(y \neq z))) \Rightarrow \\ \Rightarrow (((x \cup (y \cup z)) \neq ((x \cup y) \cup z))). \quad (1)$$

T2: Operation \cup is not bisymmetrical (with regard to relation \sim):

$$\exists w, x, y, z \in \Xi, (((w \cup x) \cup (y \cup z)) \neq ((w \cup y) \cup (x \cup z))). \quad (2)$$

As consequences of A1 – T2 we can find that operation \cup is neither *additive* nor *metric* (in terminology introduced in, e.g., [25]). This implies that for Conceptual Space there is no general method for constructing an appropriate metric function for measuring the "distance" between the members of CS . And from this ensues that there is no general method for

designing the feedback control loop needed for programmable development (evolution) of CS members.

The above thoughts about the nature of a Calculus for Conceptual Space can be understood as a very small contribution to the discussion of why the operations in Conceptual Design are rather qualitative, why quantitative methods in control of CDP are unnatural, and why computer support for CDP needs special approaches and means.

4 Ontologies and conceptual structures

Research on Ontologies nowadays belongs in the field of Artificial Intelligence, but it also plays a significant role in Conceptual Design. The term ontology has been used in the following senses [30, 31]:

- a philosophical approach to the investigation of "being",
- an informal conceptual system,
- a formal semantic account,
- a specification of a conceptualisation,
- a representation of a conceptual system via a logical theory,
- a vocabulary used by a logical theory,
- a meta-level specification of a logical theory.

For this paper the most convenient interpretation is "Ontology is a specification of a conceptualisation". From the knowledge representation point of view, ontologies are semantic networks, very appropriate for conceptual modelling. The main objectives pursued by research on ontologies are:

- sharing and interchange of knowledge,
- management of knowledge,
- data retrieval.

It is clear that ontologies participate in all conceptual designing and their main merit is the opportunity to combine different professions, expert knowledge and points of view in the required domain. Research on ontologies is at present focused on representation and on semantic modelling. In addition to Ontologia [34] there are a few serious candidates for this place. A promising candidate seems to be a combina-

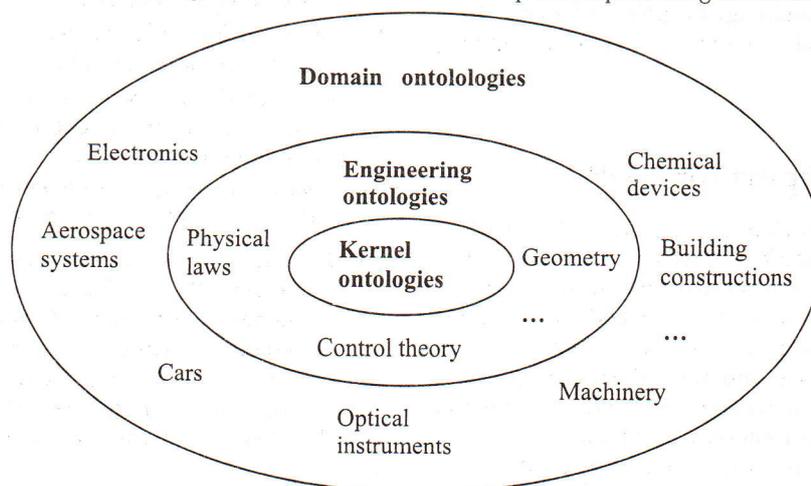


Fig. 3: Example of a hierarchical structure of ontologies

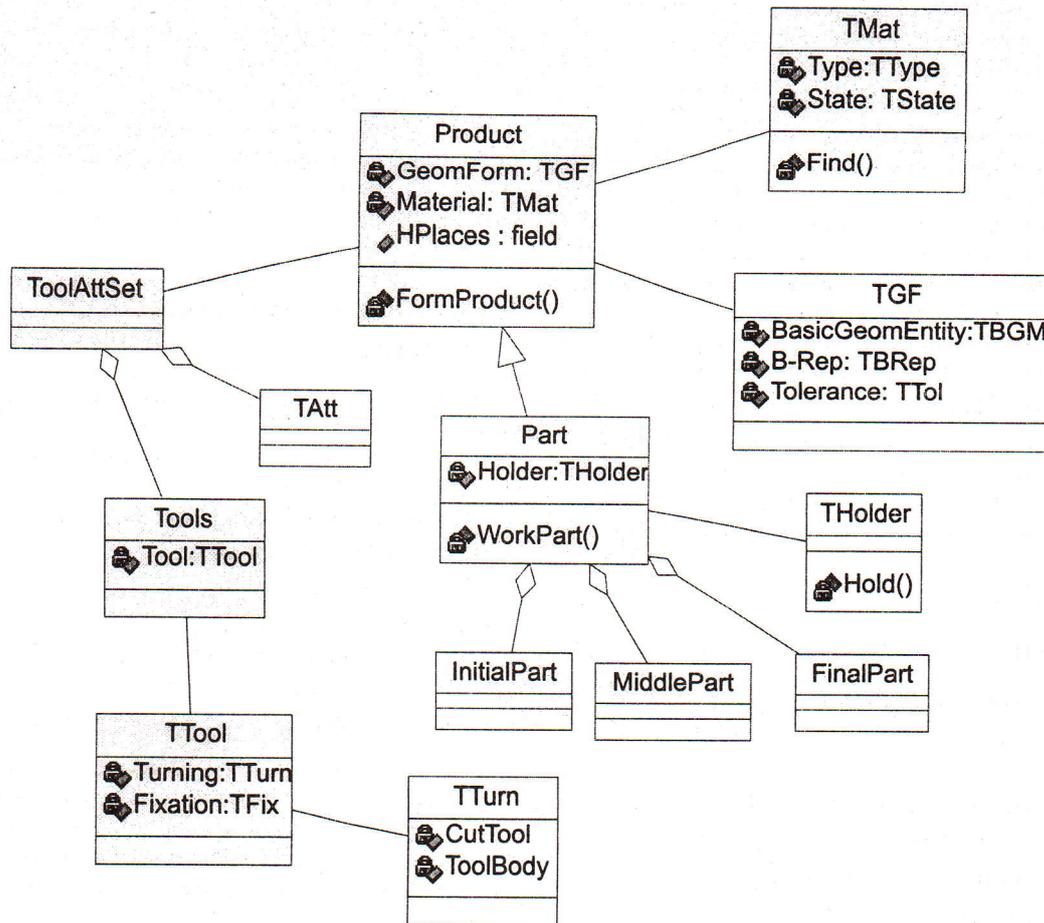


Fig. 4: Fragment of a multi-view ontology of a product

tion of **OMT** methodology (Object Modelling Technique) [16] and the **UML** language (Unified Modelling Language) [17]. UML is able to represent semantic networks and OMT is important for developing a factual ontology. Fig. 3 shows various types of ontologies neighbouring on engineering ontologies. Fig. 4 shows a fragment of the ontology [32] of a product description (combining the points of views of the product designer, technologist and CNC software engineer) developed in UML.

5 Computer support for CDP

From a purely white-collar point of view, CDP could go on some database in a network of retrieval and composition procedures. However, this is a rather administrative image. CDPs differ in their internal structure and the computer support for their components also varies. The analysis of conceptual design processes underlines the following criteria by which various CDPs can be compared with each other by means of their most important characteristics:

- P1. Translation of the initial specification into CDP.
- P2. Decomposition of Functions and Structures.
- P3. Proper method for forming conceptual constructions.

- P4. Verification of the correctness of the CDP result function.
- P5. Way of modelling the emergence of novel solution.

In order to concentrate information about the level of CDPs and about the type of designed systems, the whole field of various CDPs will be considered decomposed into three classes of Conceptual Designs:

- A) *Conceptual design of Configurations* (flats, buildings, parks, allocation of machines in halls, ...).
- B) *Conceptual design of technological components, machines and devices* (holders, attachment tools, frames, bicycles, cars, paragliding sets, refrigerators, heat pumps).
- C) *Conceptual design of systems* (control systems, technological systems, transport systems, telecommunication systems).

Note 5.1: The above decomposition is one of many. It is conditioned by criteria P1, ..., P5 and by a certain temporary interest of designers. It is an example of a decomposition, and it could not induce a discussion of the type "Is the paragliding set a machine?". A, B, C, represent certain ontologies as levels of conceptualisation. (It is clear that a refrigerator may be considered as an element in all classes A, B, C, according to need.)

The orientation results of the evaluation of classes A–C by criteria P1–P5 are as follows:

Conceptual design of Configurations

- P1A:** The translation of the specification is performed by a graphic interface. The programming environment may be of the Prolog type.
- P2A:** Decomposition of the structures is determined by the content of the library of structural elements and parts.
- P3A:** The composition operations for elements and parts are defined.
- P4A:** Direct verification of the specification without the need for any additional computation.
- P5A:** Visual interaction of shapes – occasional emergence. (The Magic of M.C. Escher, [37].)

Note 5.2: If we consider the design of configurations of machines in a hall the conceptual know how is the order and arrangements of the machines, not the solvability and possible productivity of the designed allocation set. If we need simulations the problem belongs to class C.

Conceptual design of principles in technological components, machines and devices

- P1B:** Graphic interface. The specification contains static, kinematic and dynamic parameters of the designed target. We assume the translation of the specification into some available programming language (e.g., Delphi, C++, Prolog).
- P2B:** Decomposition of the functions is a substantial task. It is necessary to define the functions of the components. Decomposition of the structures is easy. We assume a developed database (library) of design elements and parts.
- P3B:** Composition operations which realise a design as the composition of principles, functions and structures of elements and parts into wholes are performed by rules of composition and restrictions. We assume a developed rule-based knowledge base.
- P4B:** We take into consideration small verification computations or small simulation experiments.
- P5B:** The modelling of the emergence of a novel solution is not assumed.

Conceptual design of systems

- P1C:** The translation of the specification is performed by a graph-symbolical interface. The specification contains behavioural description, many functional and structural parameters of components and wholes. The application of special methodologies is assumed (e.g., OMT) and translation of the specification into special languages and program environments (e.g., UML, STEP 7).
- P2C:** The decomposition of the complex function of the designed system into sub-functions is performed by special rules and by means of the database of elements and subsystems. There are supposed a developed knowledge base and a library of elements and sub systems.
- P3C:** The proper method for forming conceptual structures only extends (by rules for compositions and restrictions) the decomposition process from P3B.

P4C: No detailed verification of the functional correctness of the system is available within the framework of CDP. This would require large simulation models and many experiments. There are three ways to approach this important issue:

- by approximating the behaviour and the properties of the developed prototypes.
- by using some novel method of verification without experiments.
- by utilising UML representation of the system and by means of some appropriate CASE system to generate the code of simplified simulation and visualisation programs.

P5C: Modelling the emergence of a novel solution is not assumed.

5 Artificial intelligence in conceptual design

This section will introduce examples of AI systems developed to support CDP.

A. "Standard" AI CDP Support Systems

This class includes systems which use decomposition and composition operations in the Function-Structure platform and which work with a pre-formed dedicated database and knowledge base. Examples of such systems are Galileo [6], AIDA [8] and GPAL [15].

GALILEO is a knowledge-based CDP support system. [6] presents examples of its application for conceptual design of two classes of devices (which were commercially interesting) but we can imagine how difficult (or how easy) it would be to develop its application for the conceptual design of an another similar system. The kernel of the system is a knowledge base which contains atomic and partial essential structures of the type "*Required function* \Rightarrow *Means of its realisation*". The principle of CDP lies in decomposing the global function of the designed system (included in the specification) into subfunctions – Fig. 5.

The verification process takes place during designing, and its efficiency is limited by the content of the library of elements and parts and by the implemented constraints.

AIDA is described in [8] as an AI system for computer support for Conceptual Design of complex systems. It is based on a combination of three AI tools: Case Based Reasoning for suggesting the initial proposals, Rule-Based Reasoning to assess these proposals and their functional qualities (i.e., small computations and checking points), and Constraint-Based Geometrical Modelling for visualisation of the proposals. These tools are developed as independent modules. One of published applications of AIDA is dedicated to Conceptual Design of Aircraft.

GPAL (Green Product All Life-Cycle) is a CAD system that integrates conceptual and detailed design. The kernel of the system consists in four modules:

- functional element library,
- a knowledge-based "Function to Form" mapping mechanism,
- an assembly model,

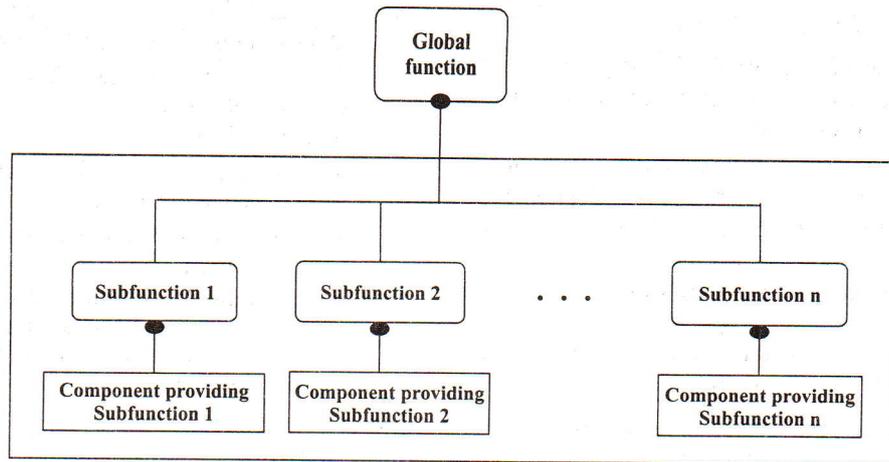


Fig. 5: Decomposition of a global function into sub-functions

- a module for geometric reconstitution of functional carriers based on default geometric reasoning [15]. The verification algorithms work with abstract features (conceptual geometric data structures are defined).

All the above systems demonstrate formal facilities which are added to the “classical” loop of conceptual design (“Proposals-Evaluation-Corrections”) and show still continuing differentiation of the abilities of such systems.

B. “Prototyping” AI CDP Support Systems.

The prototyping approach was frequently quoted and applied, especially in the late 1980s. The kernel of this ap-

proach consisted in an image of a functionally representative but uncompleted product of CDP which was sufficient for verification and for iterative corrections of the design. The greatest advantage of prototype methods was in verifying the specification requirements, which was independent of the completion of the detailed design stage.

The prototyping approach was implemented very widely from CAD/CAM systems with physical prototypes of products to sophisticated program prototypes in the Software Engineering field. Fig. 6 illustrates an application where OMT – UML – CASE are linked in CDP. Working with OMT methodology, the formulation of the problem environment and the

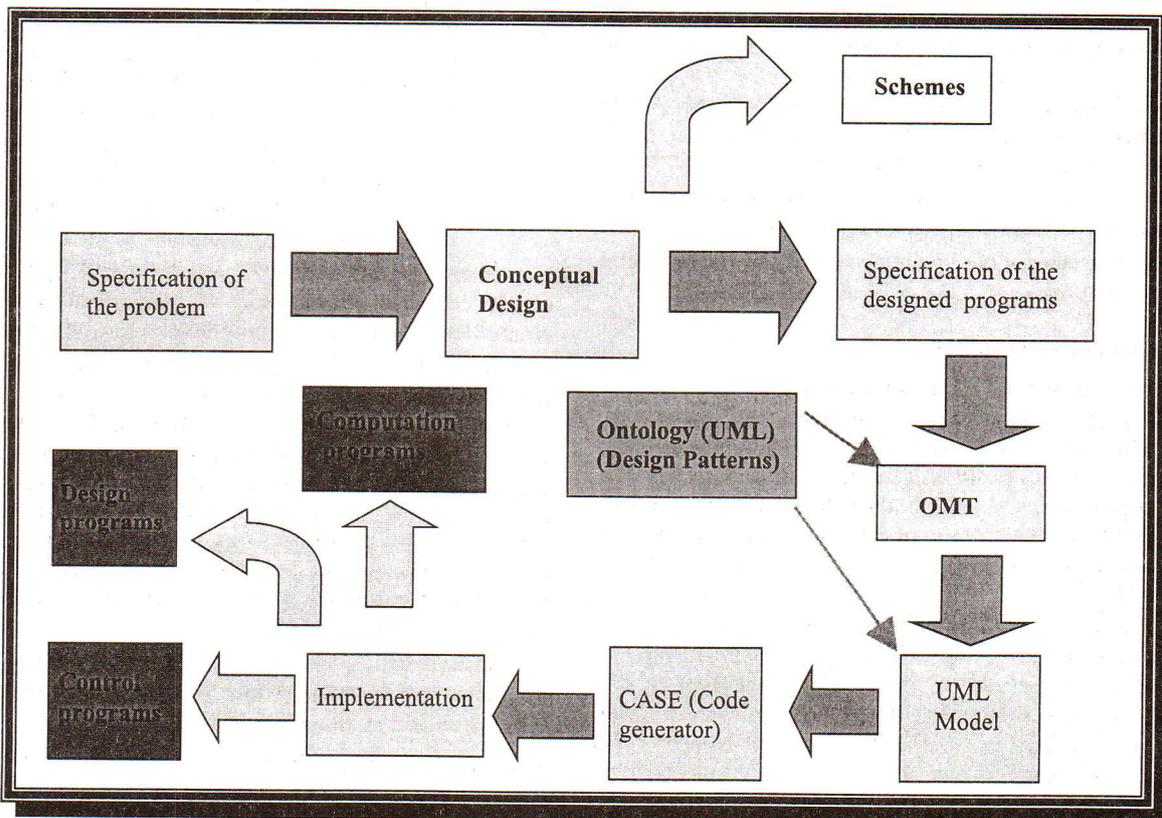


Fig. 6: Ontology – OMT – UML – CASE

problem solution goals are “translated” into the UML model. This model may be considered as a preliminary stage of the conceptual design process. If the rules for developing the formulation are detailed at the necessary level (e.g., in the specification of simulation, visualisation and measurement tools) and an appropriate CASE system with a good generator of code is used for UML model, the results of CDP are represented by means of the functions and the results of the generated programs. (This way is surprisingly easily accessible, and it is not limited only to the conceptual design of systems which end at the level of programs.)

C. Interpretation AI CDP Support Systems

A traditional form of systems which support interpretation is a set of rules. In a more detailed proposal it is better to speak about a multi-view approach combined with a gradual knowledge acquisition procedure.

One such system is **MMforTED** [24], which was developed to support the acquisition of several ontologies for reasoning about an artefact from different viewpoints. The system works with a hypertext browser which enables the designed object (situation, system) to be identified with pre-formed models. Browsing through the network of models induces simultaneous changes of views. The system integrates model-based reasoning, ontological engineering and hypermedia and web-based instruction. The system facilitates the exploration of design situations from different conceptual perspectives and supports problem setting and design development.

D. Support for Emergent Phenomena within CDP

Though the emergence of novel solutions as outcomes of CDP is widely expected, the known means for formalisation and computer (or other) support for emergence phenomena in the field of CD have not been too effective. Promising projects in this field are usually covered by research support for creative phenomena.

A relatively old approach (still quoted and still being developed) which may be included in the field of Conceptual Design on the left side is *formal and computer support for TRIZ methodology* [10]. This method works with a set of rules which have a formal representation of a technical problem (one of the formal tools for describing the problem is the “Substance-Field” language and calculus) and on the right side a description of the solution operations. The operations are described as general heuristics, and the solution is achieved by interpreting these heuristics in the conceptual environment of the solved problem. This interpretation procedure represents a special semiotic process which may be associated with an *emergence phenomenon*. This approach and method may serve as an experimental space for investigating emergence phenomena, though the set of rules (acquired by analysing an enormous number of creative technical solutions) is rather large (more than 1200 in [10]).

A *formal description of emergence conditions* within CDP was introduced in [13]. The authors of this work described the emergence phenomenon by a context-preserving morphism from an original semiotic algebra to another semiotic algebra and by a condition claiming that the discovered solution is not in the co-domain of actions of the interpretation of the original algebra. This contribution focuses on explaining the

emergent results acquired in an intuitive creative way, and the user application of this approach it is only illustrative.

Another formal approach leading to a computer support tool for *the emergent phenomena within a co-evolutionary variant of CDP* has been presented in [28, 29].

Finally, a very promising line of research into emergent phenomena within CDP is the *multi-agent approach*, where the emergence of a novel conceptual solution is achieved as a result of information interaction of agents.

The field of Conceptual Design was included as an experimental platform in the project “Methodology of Emergent Synthesis”, as one of four parts of a project on “Science of Synthesis” (Hi. Yoshikawa, Japan, 1996). The essential ideas of *Emergent Synthesis* are explained, e.g., in [35]. The general direction of Emergent Synthesis is slightly different from the approaches to the processing of emergence phenomena introduced above.

7 Conclusions

The answer to the question “To whom belongs Conceptual Design?” is that present-day research is concentrated on *novel principles of designing* (related to modelling of the mind), on *ontologies* (for semantical modelling), on *intelligent tools for computer support of CDP* and on *emergent solution theory*.

Acknowledgements

This research has been conducted at the Institute of Instrumentation and Control Engineering, FME, CTU in Prague and has been supported by Research Grant GACR 102/01/0763.

References

- [1] Deng, Y. M., Britton, G. A., Tor, S. B.: *Constraint-based Functional Design Verification for Conceptual Design*. Computer-Aided Design, Vol. 32, 2000, p. 889–899.
- [2] Jansson, J., Horvath, I.: *Behavioral Simulation of Incomplete Representations in Conceptual Design*. In: Proc. of the 15th European Simulation Multiconference (Ed.: Gerril, P.), Delft: SMS Press, 2001, p. 321–325.
- [3] Roy, U., Pramanik, N., Sudarsan, R., Sriram, R. D., Lyons, K. W.: *Function-to-form Mapping: Model, Representation and Application in Design Synthesis*. Computer Aided Design, Vol. 33, 2001, p. 699–719.
- [4] Taura, T.: *Design Science for Functional Design Process Modelling*. In: Proc. of 10th Int. Conf. on Engineering Design – ICED 95. (Ed.: Hubka, V.) Edition Heurista 95, Vol. 2, 1995, p. 456–464.
- [5] Suh, N. P.: *The Principles of Design*. Oxford Series on Advanced Manufacturing. New York: Oxford University Press, 1990.
- [6] O'Sullivan, B.: *Constraint-Aided Conceptual Design*. London (UK): Engineering Research Series, 2002.
- [7] Tay, F. E. H., Gu, J.: *Product Modelling for Conceptual Design Support*. Computers in Industry, No. 48, 2002, p. 143–155.
- [8] Rentema, D., Jansen, E.: *An AI Tool for Conceptual Design of Complex Products*. In: Design Research in Netherlands

- 2000 (Eds.: Achten, H., Bauke de Vries and Jim Hennessey). TU/e, 2000, p. 119–132.
- [9] Cottam, R., Ranson, W., Vounckx, R.: *Emergence: Half of Quantum Jump?* Acta Polytechnica Scandinavica, Mathematics, Computation and Management Engineering. Ser., 1998, p. 12–19.
- [10] Savransky, S. D.: *Engineering of Creativity: Introduction to TRIZ Methodology of Inventive Problem Solving*. New York: CRC Press, Boca Raton, 2000.
- [11] Opio, E. Z., Horváth, I., Vergeest, J. S. M.: *Quality Assurance of Design Support Software: Review and Analysis of the State of Art*. Computers in Industry, No. 49, 2002, p. 195–215.
- [12] Kitamura, Y., Sano, T., Namba, K., Mizoguchi, R.: *A Functional Concept Ontology and its Application to Automatic Identification of Functional Structures*. Advanced Engineering Informatics, No. 16, 2002, p. 145–163.
- [13] Kryssanov, V. V., Tamaki, H., Kitamura, S.: *Understanding Design Fundamentals: How Synthesis and Analysis Drive Creativity, Resulting in Emergence*. Artificial Intelligence in Engineering, 2001, No. 15, p. 329–342.
- [14] Panowski, E.: *Meaning in the Visual Arts*. Garden City, New York: Doubleday & Company, Inc., 1955.
- [15] Xu, Z. G., Tang, M. X., Frazer, J. H.: *From Concept to Embodiment: Challenge and Strategy*. In: Proc. of Int. Conf. Artificial Intelligence in Design '02. (Ed.: J. Gero), Dordrecht: Kluwer Academic Publisher, 2002, p. 215–236.
- [16] Rumbaugh, J. E., Blaha, M., Premerlani, W. J., Eddy, F., Lorensen, W.: *Object-Oriented Modelling and Design*. New York: Prentice-Hall International Inc., 1991.
- [17] Rumbaugh, J. E., Jacobson, I., Booch, G.: *The Unified Modelling Language Reference Manual*. Reading, MA: Addison Wesley, 1999.
- [18] Sharpe, J. E., Bracewell, R., H.: *The Use of Functional Reasoning for the Conceptual Design of Interdisciplinary Schemes*. In: Proc. of 10th Int. Conf. on Engineering Design – ICED'95. (Ed.: Hubka, V.), Edition Heurista 95, Vol. 2, 1995, p. 465–470.
- [19] Olesen, J.: *Strengthening the Understanding of Conceptual Design*. In: Proc. of 10th Int. Conf. on Engineering Design – ICED'95. (Ed.: Hubka, V.), Edition Heurista 95, Vol. 2, 1995, p. 471–476.
- [20] Bíla, J., Weiglhofer, W.: *Multimodelling of Large Scale Systems in Pre-Implementation Stage*. In: Proc. of Int. Conf. on Modelling and Simulation 2002. Universidad de Girona Press, 2002, p. 303–310.
- [21] Peckham, J., MacKellar, B.: *Generating Code for Engineering Design Using Software Patterns*. Artificial Intelligence in Engineering, No. 15, 2001, p. 219–226.
- [22] Krishnan, P.: *Consistency Checks for UML*. In: Proc. of 7th Asia-Pacific Software Engineering Conference, IEEE, 2000, p. 162–169.
- [23] Bíla, J.: *The Optimality of UML Model Structure?* Acta Mechanica Slovaca, No. 2, 2002, p. 37–44.
- [24] Toppiano, E.: *MMforTED: A Cognitive Tool Fostering the Acquisition of Conceptual Knowledge about Design Products*. In: AI in Design '02. (Ed.: J. Gero). Dordrecht: Kluwer Academic Publisher, 2002, p. 191–214.
- [25] Pfanzagl, J.: *Theory of Measurement*. Würzburg-Wien: Physica-Verlag, 1971.
- [26] Tichý, P.: *The Foundations of Frege's Logic*. de Gruyter, 1988.
- [27] Jelínek, I.: *Modelling of Design Process with Modal Logic*. In: Proc. of the 15th European Simulation Multiconference (Ed.: Gerril, P.), Delft: SMS Press, 2001, p. 361–365.
- [28] Bíla, J.: *Emergent Phenomena in a Co-Evolutionary Conceptual Design*. In: AI in Design '02. (Ed.: J. Gero). Dordrecht: Kluwer Academic Publisher, 2002, CD-ROM, paper No. 1.
- [29] Bíla, J.: *Degree of Emergence in Evolutionary Synthesis Process*. In: Int. Conference on Neural Networks – NN '2000, (Ed.: M. H. Hamza), Calgary: IASTED Press, 2000, p. 142–147.
- [30] Brasethvik, T., Gull, J.: *Natural Language Analysis for Semantic Document Modelling*. Data and Knowledge Engineering, No. 38, 2001, p. 45–62.
- [31] Guarino, N., Giaretta, P.: *Ontologies and Knowledge Bases, Towards a Terminological Clarification*. 1995, <http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/KBKS95.pdf>.
- [32] Dartigues, CH., Parisa, G.: *Product Data Exchange Using Ontologies*. In: AI in Design '02. (Ed.: J. Gero). Dordrecht: Kluwer Academic Publisher, 2002, p. 617–637.
- [33] Sloman, A.: *Interacting Trajectories in Design Space and Niche Space: A Philosopher Speculates About Evolution*. In: Proc. of 6th Int. Conf. on Parallel Problem Solving from Nature. September 18–20, Paris (France), 2000, p. 196–207.
- [34] Gruber, T. R.: *Ontolingua: A Mechanism to Support Portable Ontologies*. Stanford University, Knowledge Systems Laboratory, 1992.
- [35] Ueda, K.: *Emergent Synthesis*. Artificial Intelligence in Engineering, 2001, No. 15, p. 319–327.
- [36] Materna, P.: *The World of Concepts and Logic*. Philosophy, Prague (Czech Republic), 2000.
- [37] Thé, E.: *The Magic of M. C. Escher*. London: Joost Elffers Books, Thames & Hudson, Ltd., 2000.

Prof. Ing. Jiří Bíla, DrSc.
phone: +420 224 352 534
fax: +420 233 336 414
e-mail: bila@fsid.cvut.cz

Czech Technical University in Prague
Faculty of Mechanical Engineering
Technická 4
166 07 Prague 6, Czech Republic