Marek Makowski,[1] Andrzej P. Wierzbicki[2]

Modelling Knowledge in Global Information Networks[3]

Abstract

After recalling the concepts of *networked society, information society and knowledge-based economy,* the paper concentrates on problems of sharing or exchanging knowledge in information society. In order to analyse these problems, the concept of knowledge must be first more precisely defined and distinguished from the concepts of data, arts, wisdom, and skills. In computerised form, knowledge can be represented either as learned texts, or as computerised models or systems of models. Classical models in knowledge engineering take the logical form of patterns discerned in data with the help e.g. of data mining. However, there are also many other mathematical models used to express knowledge in a computerised form: each discipline of science or engineering has its own tradition of representing knowledge with specific class of mathematical models. This fact makes it difficult to exchange or share knowledge in this form. A new possibility of overcoming this difficulty is offered by the concept of grid middleware, a layer of software supporting integration of knowledge developed and stored with diverse standards, and providing for co-operation of diverse software entities. The paper shows an example how the concept of grid middleware can be used to help in exchanging or sharing knowledge in computerised model form. Conclusions of the paper relate to problems of science policy, particularly for countries in economic transition, resulting from the problems of transition towards knowledge-based economy and sharing or exchanging knowledge.

1. Networked society, information society and knowledge economy

There are various perceptions, diagnoses and concepts describing the current *global information revolution*, but it is generally accepted that new *global information infrastructure* will gradually result in *knowledge-based economy* and in *information society* or even in *networked information civilisation*. Actually, the concept of

---

[1] International Institute for Applied Systems Analysis, Schlossplatz 1, A-2361 Laxenburg, Austria.
[2] National Institute of Telecommunications, Szachowa 1, 04-894 Warsaw, Poland.
[3] Draft version of a chapter included in the Proceedings of the 4th Global Research Village Conference: *Importance of ICT for Research and Science: Science Policies for Economies in Transition,* pp. 173 – 186, KBN (the Polish State Committee for Scientific Research), and OECD (the Organization for Economic Co-operation and Development), published by Information Processing Centre, Warsaw, 2003.

information society is oldest, starting in Japan about 30 years ago; but it was about 10 years ago adopted by European Union in response to the concept of global information infrastructure promoted by United States. OECD tried to combine both these concepts by advancing the notion of knowledge-based economy. M. Castells in his monumental work[4] used the concepts of information age and networked society. For all these diverse approaches to the current information revolution, there is also a common basis. There is no doubt that *information* and *knowledge* are becoming essential economic assets with either private or public character and that it is necessary to develop either rules of their sharing or business models of their selling and exchange.

For countries in transition towards market economy and democratic society, a basic problem of science policy is whether to aim the transition targets towards classical industrial market economy or towards knowledge-based market economy. Most of advice coming from international institutions stresses the former option; however, this results in neglecting the development of science and education by most countries of transition economy. The resulting crisis of systems of education and science that can be observed in many such countries destroys the foundations of possible fast transformation towards knowledge-based economy. Therefore, it is necessary to reflect in some more detail on the conditions and forms of exchanging or sharing knowledge and on resulting conclusions for science policy of transition economy countries.

2. How to distinguish knowledge

While there is an increased interest and work on *data and information sharing* or *exchange,* less attention is given yet to *exchanging* or *sharing knowledge.* One of the reasons is that the concept of knowledge is less sharply defined and standardised than the concept of data and information. Increased interest in defining knowledge comes recently from management science, along with the concepts of computerised *knowledge management.*[5] However, definitions of knowledge used in management science are rather comprehensive, actually including such characteristics as skills or even wisdom. We need here a sharper definition.

For the purpose of the paper, we define knowledge as *a synthesis of information aimed at a selected field of applications, testable in practice and presented in a communicable form.* The requirement of a synthesis aimed at definite purpose

[4] See M.Castells (2000).
[5] See F. Gao *et al.* (2002)

distinguishes knowledge from information. The requirement of practical testability distinguishes knowledge from *arts*. The requirement of communicable form distinguishes knowledge from *wisdom that* is more personal and intuitive, results from experience and might be difficult to communicate, as well as from *skills that* are also personal and result from experience and training. In *arts, skills* and *wisdom,* a considerable role can be played by *intuition.* According to a rational theory of intuition proposed by one of the authors of this paper,[6] intuition is based on a pre-verbal, subconscious activity of our brains, much more powerful than verbal and logical activity. This is because modern telecommunications and complexity theory indicate that processing images might be about 10^4 times more complex than processing words. Intuition is acquired by training and by subconscious synthesis of knowledge and information. In its operational form, intuition is thus strictly related to skills; in its creative form, to arts and wisdom.

However, by this very definition of intuition, it is extremely difficult to communicate it by words. Thus, knowledge is in a sense opposite to intuition: it must be communicable, by words of either natural or artificial languages – such as computer programming or computer modelling languages. If some elements of wisdom and skills can be objectively communicated, they become part of knowledge. Thus, a learned text e.g. on software engineering is a part of knowledge, although it can be argued that producing good software requires considerable skills and even wisdom, has even some aspects of art (all creative activities require such a mixture of knowledge and intuitive abilities).

3. Models as a form of knowledge

As we indicated above, an essential form of knowledge is a learned text. During the history of printing books, we developed various ways of distinguishing fiction from learned treatise, diverse mechanisms of knowledge certification. In contemporary global information networks, however, this distinction and related certification mechanisms are usually lost. Obviously, not all computerised texts available in global information networks represent knowledge and we need to re-establish appropriate certification mechanisms.

For the purpose of this paper, however, we are more interested in another form of knowledge: that of *models*. Naturally, a learned text expressing knowledge can be also considered as models. In order to convey knowledge, a text must contain some structure that is actually a form of a (verbal) model. In the network of global

[6] See A.P. Wierzbicki (1997).

information infrastructure, however, knowledge has been also communicated in the form of computerised models, and given the fast growing complexity and amount of knowledge, models will become an important form of knowledge management. Models can have diverse interpretations. In so-called *knowledge engineering*, models of knowledge are defined very narrowly as patterns found in large data sets. In broader fields of scientific computing or applied systems analysis, knowledge is formulated as diverse types of computerised models. Models can be categorized according to their mathematical properties (linear and nonlinear, static and dynamic, continuous or discrete, deterministic, stochastic, fuzzy, rough etc.) or according to a modelled phenomena/problem.

Even for a well-defined problem one can develop very different models depending on the purpose for which a model is built. The elements of knowledge represented by models are of two types. First, reflecting the types of relations between a selected set of variables (which constitutes a symbolic model specification), and second, various model instances composed of the model specification and a selected set of data (often coming as result of analysis of other models) defining its parameters. Moreover, there are various paradigms[7] of computerised modelling, related either to a distinct scientific discipline or to traditions of a specific school of thought. Due to the unquestionable success of modelling in problem solving, various modelling paradigms have been intensively developing over last few decades. In this, to a great extent case study driven process, a growing tendency to focus on specific methodologies and tools was observed. Therefore different types of models (characterized by the types of variables and relations between them, as mentioned above) were developed to possibly best represent different problems by a selected type of model. Various abstract (symbolic) model representations, and corresponding formats of their computerized implementations have been developed within diverse modelling paradigms. Moreover, different methods of model analysis (e.g. simulation, optimisation, soft simulation, multicriteria model analysis) have been developed to best support various types of model analysis for different purposes and/or users. Various types of analysis require solving corresponding underlying computational tasks. Due to the growing complexity of such tasks, solvers have become more and more specialized for specific subtypes of mathematical programming problems (e.g. linear programming problems that used to be solved only by the simplex algorithm, are now solved by diverse algorithms depending on a more detailed characteristics of the problem, such as sparsity, a structure of the underlying matrix, etc.).

[7] A modeling paradigm embodies the consensus of a scientific community on development and analysis of a model, and consists (see Hloyningen-Huene, 1993) of the theories, laws, rules, models, concepts, and definitions that are generally accepted in science and practice, as well as of corresponding modeling tools.

All these developments have been rational from the point of view of providing more efficient solutions for specific types of models or elements of modelling process. However, as a consequence of this long-term development, it has become increasingly difficult to apply all pertinent modelling paradigms to a problem at hand. Models have become complex and/or large, therefore development and use of a model with even one specific paradigm is a costly and time-consuming process. Moreover, incompatible model representations used by different paradigms imply that resources (especially various model instances, and corresponding data) developed for modelling with one paradigm can hardly be reused when another paradigm is applied to the same problem.

Nevertheless, we observe recently the development of an interdisciplinary field of *computational science*, related to various approaches of using computerised models of knowledge. We shall not discuss here in detail the knowledge and the art of computerised modelling.[8] More important is the fact that there is a great diversity of various forms and standards of computerised models of knowledge. This diversity makes difficult the objective of sharing or exchanging knowledge represented by models, and potentially available from model-based data analysis. Each modelling paradigm embodies much accumulated knowledge, expertise, methodology, and modelling tools specialized for solving many of the problems belonging to each modelling paradigm. However, these resources are fragmented, and using more than one paradigm for a problem at hand is too expensive and time consuming in practice. In other words, a great amount of knowledge is to a large extend fragmented and therefore cannot be efficiently used. However, such knowledge can become an important asset of enterprises, organizations, and societies, if its integration, further development, maintenance, and sharing will be made efficient.

4. Sharing and exchanging data, knowledge and models

The data management revolution occurred in response to severe problems with data reusability associated with file-processing approaches to application development. The need to share data resources resulted in the development of DBMSs, which separate the data from the applications that use the data. Advances in database technology have been propelled in the past decades primarily by the development, refinement, and efficient implementations of the relational data model. DBMSs make it possible to efficiently share not only databases but also tools and services

[8] See A.P. Wierzbicki (1984), A.P. Wierzbicki *et al.* (2000).

for data analysis that are developed and supplied by various providers and made available on computer networks. By providing a platform for modelling that combines the DBMS techniques for data management with several leading modelling paradigms, modelling can make a similar jump forward, resulting in a breakthrough in modelling technology that is necessary for an efficient integration and management of knowledge (both computerized in form of data and models, and tacit knowledge of modellers and users of models), which in turn is a prerequisite for model-based analysis and solving of complex problems.

A lesson needs to be learned from the qualitative jump in data processing made about a half of century ago: a structured model representation, generic enough to accommodate leading modelling paradigms is a prerequisite not only for multiparadigm modelling of complex system, but also for efficient modelling support within one modelling paradigm. However, another element (that has not been yet developed for DBMS) is also needed. Namely, knowledge-engineering-based support for management of modelling resources, and guiding a user through the whole modelling process that should efficiently use distributed modeling resources (composed of abstract model specifications, data, and modeling tools) available from diverse sources in a global network.

A structured approach to distributed modelling should contain two parts. One is management of resources (modelling middleware) necessary for distributed use of modelling resources in a global network, and for guiding a user through the whole modelling cycle while supporting a number of diversified modelling paradigms. Due to the complexity of multiparadigm modelling process that uses a great variety of heterogeneous modelling resources, such a management has to be based on a knowledge-engineering approach adapted for supporting model developers and users in an entire process of model building, validating and using models for various purposes (the whole modelling cycle). Such a support is in fact more and more important even for a selected modelling approach applied for complex problems in a specific domain.

The second key part is a structured model representation capable of handling diversified modelling paradigms, and supporting consistent and efficient model representations in formats suitable for diversified software that is specialized for various modelling tasks. A structured representation will exploit all commonly known advantages of object-oriented approach (especially the power of inheritance, which allows for sharing resources for common tasks, and for specializing methods when required for specific tasks). It will also implicitly enforce key elements of good modelling practice, many of which cannot be effectively supported by existing modelling environments. These elements include: separation of symbolic model specification from handling data needed for definition of a model parameters, checking a correctness of both syntax and semantics of a model specification, assuring consistency of a model specification with its actual implementation, automatic generation of the documentation of the whole modelling process.

These two parts will make it possible to dynamically compose applications that will support modelling activities according to the needs and preferences of developers and users of models, and exploiting the richness of all relevant paradigms.

5. A modelling Grid

The paper presents an outline of research leading to the development of such distributed modelling environment called AMEPS (Advanced Modelling Environment for Problem Solving). We concentrate here on the two key elements of AMEPS, namely the knowledge-based modelling support, and the structured model representation (called AMUR). However, before characterizing these elements we briefly outline a modelling process, which is necessary to provide a context of advanced modelling.

5.1. Modelling process

Modelling is a network of activities, often referred to as a modelling cycle, or a modelling process. Typically, such a process starts with an analysis of the problem, including description of objectives and questions to be answered. Subsequently, a conceptual (qualitative) version of a model is set up to support further discussions between modeller and user. In this phase selections have to be made of types of variables and the mathematical relations between them, to be used for calculating answers with the model.

In the next step this conceptual model directs the modeller to translate the ideas into a model specification, which is composed of mathematical (symbolic) relations, and implemented using either a general-purpose modelling tool, or by developing a problem specific model generator. Different types of variables and relations are used depending not only on the kind of problem modelled, but also on the choice of model type that is relevant to its future use, available data, and resources for model development, analysis and maintenance. For any non-trivial problem, model specification is an iterative process, which involves a series of discussions between developers (typically OR specialists) and users until a common understanding of the problem and its model representation is agreed. Substantial changes of model specification are usually made during this process.

A model specification defines symbolic (abstract) relations between model variables. These relations are of a generic (or qualitative) nature. For a quantitative analysis one needs to define a model instance, which is composed of a model specification, and of a selection of data that define values of parameters of its relations. Model instances differ by various selections of data used for instantiations of the model specification, which typically corresponds to various assumptions about the modelled problem. During the model implementation several model instances are created and tested in order to verify that the symbolic model specification is properly implemented. Typically many instances of a model are used for different sets of data corresponding to various assumptions that the user wants to examine in order to check to what extent the model adequately represents the problem. The data typically come from different sources (often also as results of analysis of other models); therefore, assembling and making data complete and consistent (e.g. defined in units consistent with specification of model relations) is a resource consuming process. An instance of the model is also called a substantive model because it represents relations between variables, but does not include preferential structure[9] of the user, which typically induce partial ordering of solutions (characterized by output variables) obtained for different combinations of values of inputs.

The next phase of the modelling process is model analysis. A typical decision problem has an infinite number of solutions, and users are interested in those that correspond to their preferences (assumptions, trade-offs). Therefore, a properly organized analysis of a model is the essence of any model-based problem support. Properly organized means that the user is supported in using all relevant methods of analysis, comparing the results, documenting the modelling process, and also in moving back to the first stage, whenever he/she wants to change the type of the model (e.g. for handling uncertainty, or imprecision of model parameters using a different type of variables or relations). During the model analysis different computational tasks are generated and solved by solvers, which are software specialized for specific types of mathematical programming problems.

Therefore, at least five distinct model representations (some of them also called formats) are used for each model: first, a documentation suitable for end-users;

[9] Preferential structure in a well-organized modeling process is defined during the model analysis phase, when users modify their preferences, typically substantially and often.

second, a symbolic representation used by modellers; third, computer-executable representation of the model specification; fourth, a representation of the model instance; and fifth, representations of various computational tasks in formats that are suitable for corresponding solvers. These representations are often inefficient because of their redundancy, possible inconsistency, and the demand for widely different requirements for handling them.

The modelling process used to be simple at the early period of computerized modelling, when the computing capacity limited the modelled problems to those that were small and well structured. Therefore a model instances used to be defined on few pages of paper, and model implementations were punched on a teletype tape, or on punch-cards. Over last few decades the complexity of models has grown substantially, models are often composed of elements (sub-models, and/or data) developed by multi-disciplinary teams typically working at remotely located institutions. Modelling technology has been incrementally advancing for each of the leading modelling paradigms. However, this development has not resulted in providing an adequate support for multi-paradigm modelling that is necessary for a comprehensive exploitation of the accumulated knowledge for model-based solving of complex problems.

5.2. Knowledge-based modelling support

The complexity of the modelling process is due to the diversity of model types, heterogeneity and amount of the needed data, methods of model analysis, and tools used in the modelling process. These elements need to be configured, to exchange information, to be executed, and to be monitored at various steps of the modelling process. This process is not simple, on the contrary, even some of its components are complex. Consider for example a solver, a piece of software that provides a solution for a given type of computational problem. Solvers, especially for large problems, are specialized for efficiency with a certain type of computational problems. There is no one-to-one correspondence between the type of model, and a solver used for its analysis, because different types of analysis result in various types of computational tasks. Moreover, even for a given type of computational problem, different solvers (or even a given solver run with a different set of parameters for the underlying algorithm) have varying efficacy, depending on more detailed characteristics of the problem. Therefore, applying a suitable solver to a given computational task typically requires specialized skills.

AMEPS will develop a modelling engine that makes tools contained in AMEPS effective and efficient in the whole modelling process. Such an engine will be able to create knowledge, and to use this knowledge to create configurations of computational tasks needed during various steps of model-based problem-analysis. The knowledge to be provided through the AMEPS advisor will be oriented towards the end-user, but it will also provide additional functionality that helps in configuring, running, and monitoring computational tasks for the whole modelling process. AMEPS will also provide modules that can be optionally used by external solvers to provide the information necessary for monitoring the computations, which may also be performed on computers at distant locations.

The main challenge of AMEPS is to provide an adequate modelling infrastructure based on knowledge engineering, and using available software and hardware resources. The knowledge is contained not only in analytical models but also in various methods and tools that support diversified paradigms of model analysis. Part of the knowledge is tacit, i.e. is composed of not structured elements possessed by modellers and practitioners. Supporting the whole modelling cycle with knowledge engineering to arrive at adequate combinations of model specification, data used, various analysis methods, and to provide efficient solvers for each computational task (that results from an applied modelling paradigm) is a primary scientific challenge. However, implementation of knowledge-based management of modelling resources (composed of models, data, solvers) distributed on heterogeneous hardware in distant locations is a question of combining knowledge with skills and wisdom.

The main advantage of knowledge engineering (the process of building a knowledge base) is that it requires less commitment, and thus less work, because a knowledge engineer only has to decide what objects and relations between these objects are worth representing, and which relations hold among which objects. Having specified these items, the knowledge base holds the relevant information about resources available on the local computer or distributed over the Internet. Therefore an inference mechanism can be developed enabling the knowledge base to adequately respond to incoming requests.

Assigning information usage and access rights to a database is important for development of any complex model, for which a team of developers needs to document modelling activities, and to control access to modelling resources. To achieve these goals, a communicating multiple-agents approach might be adapted in order to comply with users' resource requests. Modelling resources (models, data, modelling tools) are agents that communicate with each other directly by messages. The use of modern DBMS technologies provides fast response times and efficient data management. In order to solve diverse types of complex problems an assembly of agents possessing complementary skills is called for.

The availability of resources is not restricted to a local computer. Instead, a client-broker-server architecture on the Internet can be applied to AMEPS (experimental implementations are currently tested), enabling the user to search for specific resources in distant locations or to contribute new resources by easily adding them to the system. Thus, the needed applications consist of dynamically shared and geographically distributed resources (data, models, and modelling tools). Knowledge about existing model specifications and data will be stored in a (distributed) knowledge base, and made available for new applications.

5.3. AMUR – Algebraic Model Unifying Representation

AMUR aims to support most leading modelling paradigms, therefore it will be constructed to handle most of the widely used types of parameters and variables (e.g. static, dynamic, deterministic, stochastic, set-membership, fuzzy) and relations (e.g. linear, non-linear, as well as functions computed off-line, i.e. not explicitly known). In order to conform to these required features, AMUR is designed and built according to strict formal requirements comparable to requirements for a language with a grammar supporting well-defined functionality.

AMUR is not developed from scratch. A good foundation for AMUR is provided by SML[10] that is probably the most successful attempt of a formal representation of mathematical models, in a way that both computers and humans can work with. It provides a framework for many aspects of the modelling process. Attention is given to mathematical structure as well as to representing semantics. SML is compatible with fundamental manipulations applied to models, and it has been developed to provide a theory of model aggregation. SML is applicable to a wide class of models in management science, operations research, decision support, economics and engineering. Although SML has several limitations (e.g. it is rather oriented to optimisation-based paradigms of model analysis, and therefore does not support a number of other paradigms, especially those developed recently in Europe; it is difficult to use SML for models defined with differential equations and time-dependent algebraic equations, or to deal with randomness in models, fuzzy logic), SML has shown that a structured modelling approach can actually be implemented to support multidisciplinary and multiparadigm modelling. Therefore, SML provides a solid basis for the development of AMUR.

[10] See Geoffrion (1987).

AMUR is designed to support *algebraic models*, which is a large class of various types of models. In order to provide not only scientifically founded and precisely defined, but also efficient support, the design and implementation of AMUR conforms to three basic requirements: first, each type of model should be handled efficiently, i.e. without a substantial overhead caused by functionality needed for other types of models; second, common functionality has to be provided without reimplementations; third, AMUR must be open to future extensions.

Object-oriented approach with its inheritance mechanism is used for AMUR implementation. The base-class of AMUR provides the functionality needed for all types of model specifications supported by it. The derived classes use the functionality provided by a base class, and will therefore need to provide only additional specialized functionalities needed for each corresponding type of model.

AMUR will use MathML,[11] the de facto standard for specification of algebraic formulas that is compatible with XML.[12] Fast growing support for MathML and XML makes it possible to combine them with the proven DMBS technology. Such an approach allows for a consistent handling of the key elements of modelling, i.e. model specification, data management, comparative analysis of various model instances, and the documentation of the whole modelling process.

5.4. Summary of AMEPS objectives

We described above a distributed modelling environment, actually a modelling grid called AMEPS that is being currently developed[13] and aims at achieving a qualitative jump in modelling technology. Its basic features are:

1. It provides a structured model representation that is generic enough to accommodate leading modelling paradigms.
2. It uses knowledge-engineering methods to manage resources (models, data, modelling tools), and to guide a user through the full modelling cycle (including specification, generation, validation, analysis, maintenance, and documentation).
3. It uses individual models as a part of a system of models, and integrates two or more independently developed models.

---

[11] http://www.w3.org/Math/
[12] http://www.w3.org/XML/
[13] Contact marek@iiasa.ac.at for information about the status of the AMEPS development.

4. It provides access to various tools that support model analysis (solving) based on all relevant modelling paradigms (e.g. simulation, optimisation, multicriteria model analysis, soft simulation).
5. It provides efficient and secure access to shared resources (models, data-bases, software tools, hardware) over the Internet; all of this seamlessly integrated with the information infrastructure of organizations involved in the modelling process.
6. It uses DBMS technology, proven and familiar to many analysts, to integrate the whole modeling process with management of data used for models.

This specific example, however, is presented here as an exemplification of a general thesis: the idea of grid middleware can be extended and specified not only for distributed data access or scientific computation access, but also for distributed knowledge management of quite advanced forms.

6. Conclusions

We can turn now back to the issues of science policy, especially in countries of transition economy, related to the problems of knowledge exchange outlined above. It is clear that a country in knowledge economy must create its own specific parts of knowledge for sharing or exchange. Thus:

the first objective of science policy must be to reverse the trend of neglecting science and education systems in transition economy countries.

However, transition economy countries cannot afford to support all parts of science system, which often necessitates systemic reforms. Thus:

the increased support for science systems must be based on a systemic reform and on a carefully selected system of research priorities, taking into account the tradition and scientific capacities of a given country.

On the other hand, current developments of information and telecommunication technology and, specifically, software development related to computerised knowledge representation, grid and middleware indicate that they should constitute an important part of any system of priorities. Thus:

a priority in science policy should be given to research directly contributing to software development; in particular, efficient development of robust software for complex, distributed systems, especially those related to

computerised knowledge representation, modelling (including data mining) and computational science, grid and middleware.

Bibliography

Applegate, A., B. Konsynski and J. Nunamaker: Model management systems: Design for decision support. *Decision Support Systems* 1986, pp. 81—91.

Borst, W.: *Construction of engineering ontologies for knowledge sharing and reuse.* Tech. rep., University of Twente, Enschede, The Netherlands, 1997.

Castells, M.: *End of Millenium: The Information Age: Economy, Society and Culture, Vol. 1: The Rise of the Networked Society, Vol. 2: The Power of Identity, Vol.3:End of Millenium.* Blackwell, Oxford, 2000.

Drucker, P.: The next society, a survey of the near future. *The Economist,* 361, Nov. 3, 2001, pp. 3-22.

Fourer, R., and J. Goux: Optimisation as an internet resource. *Interfaces* **31,** 2, 2001, pp. 130-150.

Geoffrion, A.: An introduction to structured modeling. *Management Science* **33**, 5, 1987, pp. 547-588.

Gao, M. Li and Y. Nakamori, Y. *Critical Systems Thinking as a Way to Manage Knowledge*, the 46th ISSS'02 Shanghai, 2002.

Gruber, T.: Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies* **43,** 5, 1995, pp. 907-928.

Hloyningen-Huene, P.: Restructuring Scientific Revolutions, The University of Chicago Press, London, 1993.

Jarke, M., R. Klemke and A. Nick: Broker's longue - an environment for multi-dimensional user-adaptive knowledge management. In *Proceedings of the HICSS-34 Conference*, Hawaii, 2001.

Kottemann, J., and D. Dolk: Model integration and modeling languages: A process perspective. *Information Systems Research* **3,** 1992, pp. 1-16.

Tsai, Y.-C.: Comparative analysis of model management and relational database management. *Omega* **29**, 2001, pp. 157-170

Wierzbicki, A.P: *Models and Sensitivity of Control Systems.* WNT-Elsevier, Warsaw-Amsterdam, 1984.

Wierzbicki, A.P.: On the Role of Intuition in Decision Making and Some Ways of Multicriteria Aid of Intuition. *Journal of Multi-Criteria Decision Analysis* **6**, pp. 65-78, 1997.

Wierzbicki, A.P., M. Makowski and J. Wessels: *Model-based Decision Support Methodology with Environmental Applications,* Kluwer, Dordrecht, 2000.