Conservation Based Information System for Agrifood Process Network Interoperability

Bela Csukas, Monika Varga, Sandor Balogh

Conservation Based Information System for Agrifood Process Network Interoperability

Bela Csukas[1], Monika Varga[1], Sandor Balogh[1]

[1] Kaposvár University, Research Group on Process Informatics,
40 Guba S,7400 Kaposvár, Hungary
{csukas.bela, varga.monika, balogh.sandor}@ke.hu

Abstract. Based on the stoichiometric structure of conservational processes, a general methodology has been developed for the computer aided generation and operation of network interoperability services. An appropriate set of unified building elements, having autonomous programs and communicating with a dynamic simulating kernel has been elaborated for the qualitative and quantitative tracing and tracking of trans-sectorial processes. The GNU-Prolog implemented method is scalable and makes possible the ad hoc extension of the models with the actually interesting components to be investigated. The system can also be applied for the identification of hidden resources and wastes, as well as for the analysis of the value chains. This outlines a straightforward cooperative architecture of services between the planned interoperability center and the actors, supervised by the responsible authorities. The method will be illustrated by the example of agrifood processes.

Keywords: stoichiometric processes, dynamic simulation, trans-sectorial problem solving, agrifood process networks, interoperability services

1 Introduction

The engineer designed and controlled processes in the almost closed, finite space of resources and reservoirs seem to play an essential role in the solution of the present and forthcoming economical and ecological crisis. The necessary long term, and large scale, hybrid models claim for new, computer oriented frameworks that help to manage extendable simple skeleton of process systems, case specifically.

For example, agrifood processes are built from complex, multiscale, time-varied networks that span many sectors from cultivation, through animal breeding, food industry and food trade to the consumers. Also public health and public administration are interested in agrifood management. Recently, motivated by the food scandals' initiated legislation, many powerful identification and measurement methods, standardized communication protocols have been evolved [1]. The inner traceability of the actors has also been developed, associated with the various ERP systems. However, the sector spanning traceability has not yet been solved, because neither the "one-step backward, one-step forward" passing of IDs, nor the large,

central databases, prepared for the numerous possible situations give a feasible solution.

A paradox, but powerful concept is that let us solve an apparently more difficult task. This task is the dynamic simulation of the simplified, stoichiometric mass balances that provide us the extendable transparency of the whole network. Agrifood networks can be described by process systems, characterized by the inherent feedback structure between the states and transitions. The general formal models, described by the output and state functions of the process systems, had been developed by Kalman [2], before the powerful Information Technology appeared. The General Net Theory [3] proposes a net model for the description of the respective structure of states and transitions. Many net models, like the early appeared and very innovative Petri Net [4], as well as the various State Transition Nets belong to the above family. The net models do not distinguish between the model specific conservation law based properties and the signs, corresponding to the information processes. State-of-art of process modeling was analyzed by Marquardt [5], who reviewed the methodologies and tools, developed for simulation based problem solving. The significant evolution of process modeling methodologies is determined by the process industries [6].

2 Model Specific Conservation Based Stoichiometric Processes

The notion of the measure can be understood simply as an additive quantity. In the scientific context we use Halmos's definition [7] of the measures. Accordingly, ***measure*** is an extended real valued, non-negative, countable, and additive set function over a ring.

First we define a special class of measures that fulfils the model specific conservation. The only way to interpret this general and plausible, but ill-defined physical notion is the axiomatic approach.

Let C be a measure in the space of the geometric and property co-ordinates that can change in the continuous or discrete time t. Let us denote a finite, closed region in the above space with v. The not necessarily finite and closed "environment" of this region v will be indicated with u\v and called universal complement. Let $C_V(t)$ and $C_{U\setminus V}(t)$ denote the measure C associated with the region v and with its universal complement at time t. The ***model specific conservation measures*** are characterized by the axiom that the change of the measure in any finite and closed region, v during any time interval $[t_i, t_j]$, is accompanied by the identical change of the same measure in the universal complement with an opposite sign, i.e.

$$\forall_V \forall_{i,j} \left\{ \left(C_V(t_i) - C_V(t_j) \right) = -\left(C_{U\setminus V}(t_i) - C_{U\setminus V}(t_j) \right) \right\} \quad (1)$$

Fig. 1 shows an illustration to the notion of the model specific conservational measures.

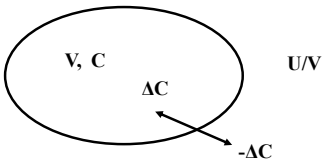

Fig. 1. Illustration of model specific conservational measures.

Constant conservational measures correspond to the quantities obeying to the conservation laws, existing in the investigated system, within the given model hypothesis. For example, within the model hypothesis of chemistry the number and the mass of the atoms are constant conservational measures. Similarly, there is conservation for the wheels, motors, windscreens, chairs, etc. in an automobile factory. Although all of these model hypotheses have a limited validity, they give a constructive, sound basis for the problem solving within the scope of the given model, because, all of the measures describing the given class of processes can be derived from these model specific conservational measures. The measures that can be derived as the homogeneous linear combinations of the constant conservational measures are called ***conservation based stoichiometric measures***.

Thinking about the example of atoms, many chemical compounds can be built from the known atoms. The number and the mass of the molecules do not satisfy the axioms of the conservation measures, because they can transform in chemical reactions. However, we can write balance equations for the reacting systems, with the knowledge of the stoichiometry that determines these secondary measures from the primary, constant ones. Similarly we can speak about the stoichiometries of the cars, or of the animals. Nevertheless, there are special additive measures (like entropy, profit) that cannot be derived from the constant measures without additional source terms.

Consider a finite closed region within a given model hypothesis. The model hypothesis can be characterized by the model specific conservational measures $\underline{C} = \{C_1, C_2, ..., C_m\}$. Designate $\underline{M} = \{M_1, M_2, ..., M_n\}$ the set of measures in the same region. Measures $\underline{M}$ are called the ***stoichiometric measures,*** derived from the constant measures iff for any time t there is a matrix $\underline{\underline{S}}$ of the stoichiometric coefficients $s_{i,j}(t)$ that satisfies the equation

$$\underline{M}(t) = \underline{\underline{S}}(t)\underline{C}(t) \qquad (2)$$

Stoichiometric balance models makes possible to develop a general methodology for the trans-sectorial interoperability for the various (e.g. agrifood) process networks. Accordingly, the essential features of the process network can be described in the special model database of the underlying mass balances in TRUs, i.e. in the unambiguously identifiable and traceable units. The actually investigated intensive parameters (e.g. concentration of the various useful or harmful ingredients, prices, etc.) can be carried with the mass batches or mass flows plausibly. The exact definition of TRU (Traceability Resource Unit) was elaborated by Kim [8] in the language of

predicate logic, considering the temporal transportations and transformations, that is familiar with the dynamic processes.

The dynamic mass balance of the input intermediates and output TRUs contains also the necessary and sufficient information about the network structure. Along the simulation of the processes, we can start from the actual states, while the functioning of the processes (i.e. extension of the database) can be solved by stepwise simulation, in line with the data supply.

In this way the various task specific intensive parameters can be carried with the mass flows, e.g. in an associated list. The respective stoichiometries can be derived exactly, or can be estimated by the experts. This solution supports the tracing and tracking of the *ad hoc* appearing problem specific components by the easy extension of the simple mass balances.

The suggested methodology claims for an IT solution, that offers the model generation from unified building elements, helps the scalable storage of the model files in databases, makes possible the case specific extension of the models, and supports the development of the effective multiscale tracing and tracking algorithms.

3 Unified Structural Model of Dynamic Processes

The development of the methodology [9] had been motivated by various practical problems [10, 11] that could not be solved with the available tools in that time.

Recently we have been applying the methodology for the sector spanning quantitative tracing and tracking of the agrifood process networks [12].

Quite different process models can be built from the developed toolbox, containing the meta-prototypes of the same building elements. The meta-prototypes (see Fig. 2) are the followings:

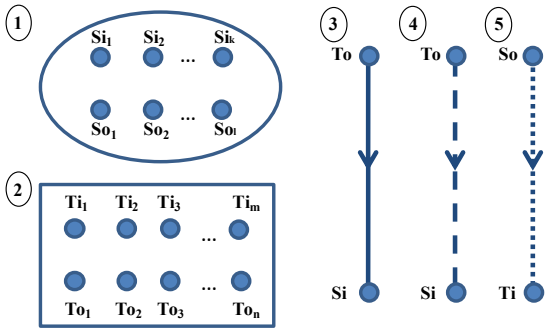

Fig. 2. Meta-prototypes of the building element.

•1. State elements, characterizing the actual state of the process (ellipses in the graphical representations);

•2. Transition elements, describing the transportations, transformations and rules, corresponding to the time-driven or event-driven changes of the actual state (rectangles in the graphical representation);

•3-5. Connection elements, designating the directed transport of the respective measures or signs between the state and transition elements (different lines correspond to the different changes in the graphical representation).

The state and transition elements contain lists (i.e. arbitrary number) of input (Si or Ti) and output (So or To) slots. The identifier and the type of slots must match to the sending (input) and receiving (output) end of the connections.

The simplified syntax of the state and transition elements is the same, as follows:

```
element(Name,Coord,ProgramName,ParameterList,InputList,
        OutputList,Timing).
```

Both kinds of elements are characterized by the following major attributes: Name: identifying name; Coord: coordinates, determining the scale and place of the given element in the geometrical and parameter space; ProgramName: identifying name of the program; ParameterList: parameter slots, prepared for the storage of the local data, associated with the given elements; InputList: input slots, prepared for receiving data from the containers of the designated connections; OutputList: output slots, prepared for sending data to the containers of the designated connections; Timing: instructions about the temporal behavior of the given element (see later).

The slots of ParameterList, InputList and Outpulist (symbolized by dots in Fig. 2) are described by the following properties: SlotName: determines the local identifier of the given slot; SlotType: gives instructions to the interpretation of the value, associated with the slot; SlotValue: contains the list of data, e.g. in the form of

```
d(DataName,DataValues,Dimension)
```

functors, where DataName: identifies the individual data set; DataValues: is the list of data (numbers or atoms); Dimension: determines the measurement unit or n/a.

The local functionalities of the state and transition elements are described by the program code, identified by the respective ProgramName. Usually many elements use the same program, declared by the prototype of the given subset of elements. In the local execution the elements receive input, execute program and send output.

The programs, referred by ProgramName from the data of InputList and ParameterList calculate the values of OutputList according to the

```
program(ProgramName,InputList,ParameterList,Outputlist):-
ProgramCode.
```

clause, where ProgramCode may be any program in the body of clause that binds the free variables of OutputList with the knowledge of the bound variables of InputList and ParameterList.

In the general case, the state and transition elements may contain both conservational and informational slots. Conservational input slots can receive data only from the increasing and decreasing connections, coming from conservational output slots. Informational input slots can receive data only from the signaling connections, coming from informational output slots. In contrary, conservational output slots can send data only via increasing and decreasing connections to the conservational input slots, as well as informational output slots can send data only via signaling connections to the informational input slots.

There may also be pure conservational and informational state and transition elements, as special cases. The syntactically identical state and transition elements can be distinguished structurally and functionally. The structural difference means that, in

the sense of the General Net Theory, only the state → transition and transition → state connections are allowed. The functional difference, in the sense of the State Space Model, is rather semantic than syntactic. Regardless to the fact, that both kinds of elements are associated with programs, at a given point of time the actual state of the process is described by the state elements, alone. In contrary, the dynamic behavior of the process is determined only by the transition elements. Accordingly, the functioning of the state elements is limited to the collection, interpretation and distribution of the static characteristic, while the transportations and transformations are modeled by the transitions.

The syntax of the

```
connection(SendOperator,SendElement,SendCoord,SendSlot,
           ReceiveOperator,ReceiveElement,ReceiveCoord,
           ReceiveSlot,DataType,DataSet,Timing).
```

is general for all increasing, decreasing and signaling connections. All of them carry data in the container of DataSet from a sending slot to a receiving slot and they are characterized by the following major attributes: SendOperator: determines the action to be done at sending slot (e.g. read, etc.); SendElement: identifies the sending element; SendCoord: refers to the (scale and place) coordinates of sending; SendSlot: defines the sending slot of the SendElement at SendCoord; ReceiveOperator: determines the action to be done at receiving slot (e.g. write, decrease, increase, remove, extend, etc.); ReceiveElement: identifies the receiving element; ReceiveCoord: refers to the (scale and place) coordinates of receiving; ReceiveSlot: defines the receiving slot of the ReceiveElement at ReceiveCoord; DataType: gives instructions to the interpretation of the DataSet; DataSet: contains the list of data, e.g. in the form of the functors: `d(DataName,DataValues,Dimension);` Timing: contains instructions about the temporal behavior of the given connection.

Increasing and decreasing connections transport DataSet from transition to state elements. Signaling connections can transport DataSet both from state to transition elements and *vice versa*. The special reading connections of the conservational substructure transport intensions (intensive parameters) from the output slots of state elements to the input slots of transition elements.

The temporal behavior of the elements and connections is declared by the associated Timing list, containing the

```
t(From,To,[When1,When2,…,WhenM],Step)
```

functors, where From: is a possible starting time; To: is a possible ending time; When1, When2,…,WhenM: are prescribed discrete times of the execution; Step: is the individual time step of the repeated execution.

The multi-scale modeling is supported also by the arbitrary number of integer coordinates given in the lists of Coord. Say, Coord = [3,7,5] refers to the fifth element in scale III, contained by the seventh element in scale II, being in the third element of scale I. The connections can be interpreted both within a scale and between scales. The model is extendable, because the number of functors d(.) on the lists at conservational slots, as well as in the containers of conservational connections isn't prescribed.

The only convention is that the optionally single first element (first functor) refers to mass, while the following ones refer to the intensions, carried with the given mass

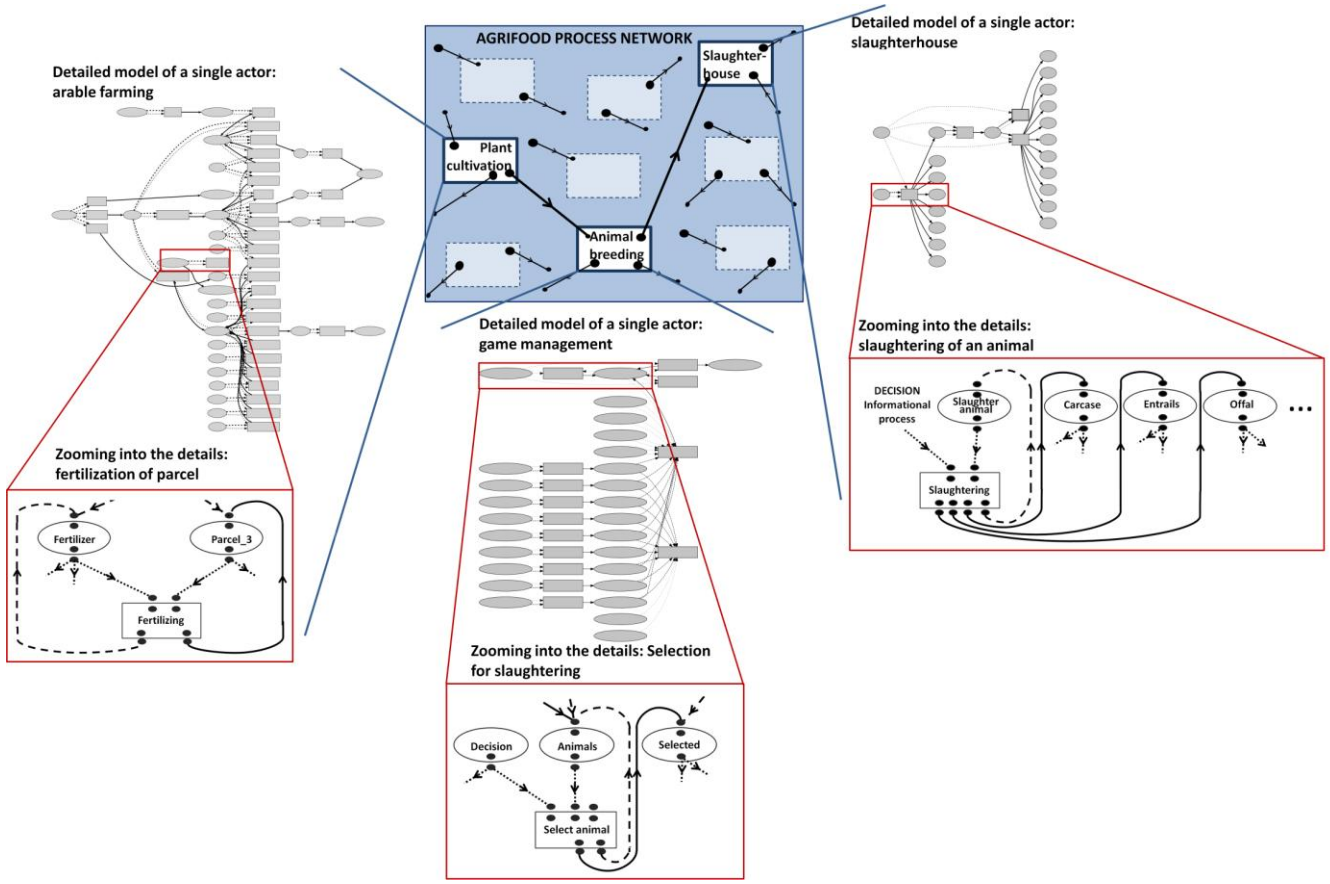

Fig. 3. Example for a multiscale model of an agrifood process network.

batch or mass flow. The conservational state elements receive extensive changes and send intensive properties, while the conservational transition elements receive intensive characteristics and send extensive changes, *vice versa*. The increasing and decreasing of the measures, as well as the extending and removing of functors can be initiated by the increase and decrease, as well as by the extend and remove operators, respectively. The distribution of the investigated new components is calculated by the exactly known or estimated stoichiometries.

The simple example in Fig. 3 illustrates how the multiscale model of an agrifood process network can be built from the above described uniform elements.

The execution of the models consists of four cyclically repeated consecutive steps, as follows: (1) transition elements read the content of the associated state elements through the reading (signaling) connections; (2) brief programs, associated with transition elements calculate the changes; (3) state elements are modified according to the changes carried via modifying connections; (4) brief programs, associated with state elements calculate the new state.

The methodology makes possible the reverse dynamic simulation of conservation based stoichiometric processes, i.e. we can start from any simulated final state and can simulate the process backwards in time. The transitions are calculated causally right, while the increases are replaced for decreases, and *vice versa*. Also the extending and removing of the add-on intensive characteristic can be changed. Consequently, with the knowledge of the stoichiometries, the inverse simulation supports the quantitative tracing of the various problem specific components backwards in space and time.

The recent implementation of the methodology has been written in GNU-Prolog, while the case specific unification of the higher level structures, contained in dynamic partitions, supports the generalized method development. Temporarily an extended GraphViz input interface based model interpreter and a CSV file based Microsoft Excel output interface are used for testing of the methodology. The ongoing new implementation is a platform independent and partly open source tool, with a QT based, interactive GUI. The interface involves a graphical modeling and design environment that allows both the user and the field expert an easy access to the input/output data, while the expert can modify and extend also the field-specific program prototypes.

4 Problem Solving Services of Planned Agrifood Interoperability Centres

The effective implementation of the above described methodology can be solved by the cooperative system of the actors in the process network, coordinated by the Interoperability Centre. The schematic architecture of the Agrifood Interoperability Centres is illustrated in Fig. 4. There are three levels of the tasks to be solved, regarding the Interoperability Center, the authorities and the actors.

In the startup phase, the Centre installs communication interfaces both for actors and authorities. The suggested method of stoichiometric mass balances makes

possible to generate uniform process models (and interfaces) from the same building blocks for the quite different technologies and activities.

For those actors, who have an appropriate ERP system, the model based interface is adapted to the existing software. It is worth mentioning that the required system of data is very familiar with the capability of the usual ERP modules. For the frequently used ERP systems easily configurable and uniform applications can be generated. For the smallest actors (e.g. minor private companies), who do not have ERP systems, a special user-friendly web application is given by the Interoperability Center.

Having installed the models in the Interoperability Center and the interfaces at the actors and authorities, the systematic data reporting, as well as the in-demand problem solving can start. The data reporting from the actors means the reporting about the new transactions, and the upgrading of process models.

It is to be noted that the majority of the systematically reported process data is limited to the new "connections", describing the up-to-date transportations and transformations. Nevertheless the method supports the assertion, modifying and deletion of state or transition elements, too.

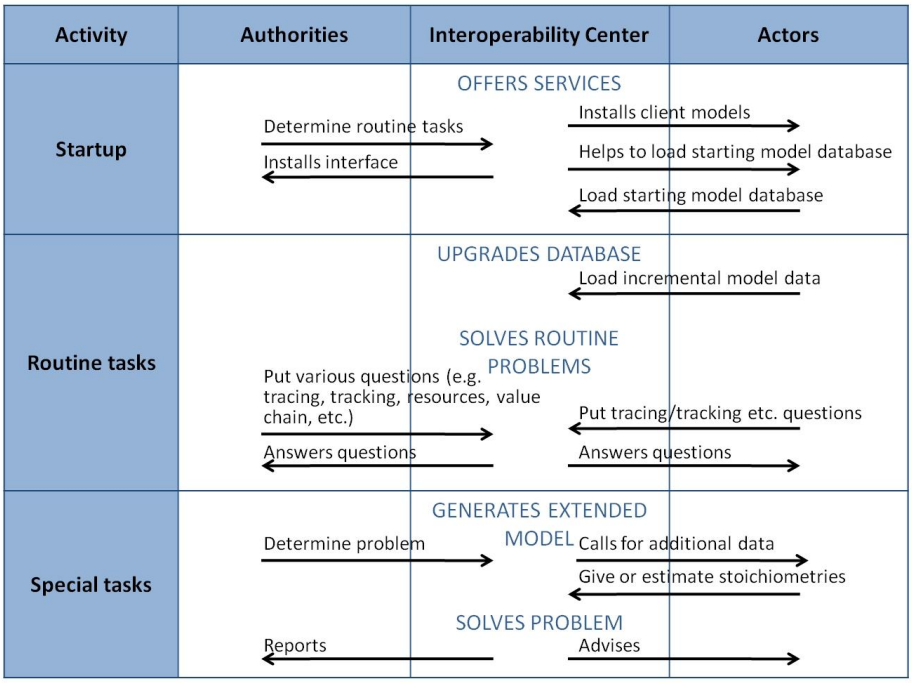

Fig. 4. Architecture of planned Agrifood Interoperability Centres.

In case of special tasks, e.g. when a harmful component appears in the network, the Centre calls for additional data (e.g. for known or estimated stoichiometries) from actors. With the knowledge of dynamic mass balance based "skeleton" of processes and the stoichiometries, regarding the investigated components, Centre runs searching algorithms, and determines possible origins and the suggested measurement points.

The most important tasks, solved by the Centre, are the followings: Qualitative tracking by multiscale search along the forward balance routes; Qualitative tracing by multiscale search along the backward balance routes; Dynamic simulation based quantitative tracking for the concentration of the known or *ad hoc* appearing components to be studied; Backward dynamic simulation based quantitative tracing for the concentration of the known or *ad hoc* appearing components to be studied; Interactive, measurement supported search for the possible origin of the various contaminations (combining the above methods with a genetic algorithm); Reporting about hidden resources or wastes on the basis of balance calculations; Trans-sectorial value chain analysis; Analysis of the basket of typical consumers' groups.

Acknowledgement

The project was supported by Baross R&D Project, #REG-DD-09-2-2009-0101.

References

1. Wolfert, J., Verdouw, C.N., Verloop, C.M., Beulens, A.J.M.: Organizing Information Integration in Agri-food – A Method based on a Service-Oriented Architecture and Living Lab Approach. Computers and Electronics in Agriculture, 70(2), 389--405 (2010)
2. Kalman, R., Falb, P., Arbib, M.: Topics in Mathematical System Theory. McGraw Hill (1969)
3. Brauer, W. (ed.): Net Theory and Applications. Springer Lecture Notes in Computer Science, 84 (1980)
4. Petri, C. A., Kommunikation mit Automaten (Communication with Automatons). Schriften des Institut für Instrumentelle Mathematik, Nr. 2, Bonn (1962)
5. Marquardt, W.: Trends in Computer-aided Process Modeling. Computers chem. Engng. 20(6/7), 591--609 (1996)
6. Yang, A., Braunschweig, B., Fraga, E.S., Guessoum, Z., Marquardt, W., Nadjemi, O., Paen, D., Pinol, D., Roux, P., Sama, S., Serra, M., Stalker, I.: A Multi-agent System to Facilitate Component-based Process Modeling and Design. Comput. Chem. Engng., 32(10), 2290--2305 (2008)
7. Halmos, P.R.: Mértékelmélet. (Measure Theory) Gondolat, Budapest (in Hungarian) (1984)
8. Kim, H.M., Fox, M.S., Gruniger, M.: An Ontology of Quality for Enterprise Modeling. IEEE Proceedings of WET-ICE, pp. 105--116, Los Albamitos, CA, USA (1995)
9. Csukás, B.: Megmaradási és információs folyamatok közvetlen számítógépi leképezése. (Direct Computer Mapping of Conservational and Informational Processes) Manuscript of DSc Theses (in Hungarian) (2001)
10. Csukás, B., Balogh S.: Combining Genetic Programming with Generic Simulation Models in Evolutionary Synthesis. Computers in Industry, 36, 181--197 (1998)
11. Csukás, B., Balogh, S., Kováts, S., Aranyi, A., Kocsis, Z., Bartha, L.: Process Design by Controlled Simulation of the Executable Structural Models. Comput. Chem. Engng. 23, 569--572 (1999)
12. Varga, M., Balogh, S., Csukás, B.: Sector Spanning Agrifood Process Transparency with Direct Computer Mapping. Agricultural Informatics Vol 1, No. 2: 73-83 (2010)