Intelligent Modeling with Agent-Based Fuzzy Cognitive Map

Maja Stula,* Darko Stipanicev, Ljiljana Bodrozic
Department of Modelling and Intelligent Systems, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, R. Boskovica bb, 21000 Split, Croatia

This article presents an agent-based fuzzy cognitive map (ABFCM) developed injecting the concept of multi-agent system (MAS) into the fuzzy cognitive map (FCM). Fuzzy cognitive map is used for qualitative modeling and simulation. Compared to the FCM, the ABFCM enables different inference algorithms in each node enabling simulation of systems with diverse behavior concepts. Each map node can exhibit individual, more or less intelligent, behavior and still can interact with other nodes to conclude on system behavior. Resulting method also enables automatic results interpretation adding additional intelligence to a classic FCM. Explanation of the obtained system architecture with FCM and MAS integration is presented in the article. The experimental results in the article are obtained with the ABFCM prototype, developed on the basis of ABFCM structure given in the article. Multi-agent technology can bring new properties into existing fields and methods, like in the ABFCM case.

1. INTRODUCTION

Qualitative modeling is an approach when system complexity prevents obtaining explicit mathematical model of a system. System complexity is a consequence of the system magnitude or insufficient system knowledge. Fuzzy cognitive map (FCM) is one of the qualitative modeling techniques used in such cases. FCM could be used for system modeling and system behavior simulation. It represents a dynamic system emphasizing the causal relations among the system elements.[1] System elements are the map concepts, sometimes directly analogues to an appreciable physical value and sometimes representing aggregate or even the abstract system values. Fuzzy cognitive maps are applied in decision support, strategic planning, virtual worlds modeling, and so on.[1–5] In short, a FCM is used for representation and simulation of complex dynamic systems. Autonomous agents and multi-agent systems (MAS) fields include synergic ideas from different areas, especially from the distributed computing, object-oriented systems, software engineering, artificial intelligence, and the organizational science.[6] Multi-agent systems are applicable to

*Author to whom all correspondence should be addressed: e-mail: maja.stula@fesb.hr.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 25, 981–1004 (2010)

View this article online at wileyonlinelibrary.com. • DOI 10.1002/int.20435

a variety of distributed, complex software systems.[6,7] Simplified, MAS is an assembly of interconnected agents, peaces of software working together (or working against each other). Fusion of the FCM and the MAS has resulted in a new FCM type thereby extending classic fuzzy cognitive map characteristics. The new FCM type, called the agent-based fuzzy cognitive map (ABFCM), or the short agent map, enables the use of a different algorithm for the new state values calculation for each concept. The Section 2 provides introduction to the FCM theory and comparison of the classic FCM and ABFCM. It also contains overview of FCM extensions and variations and comparison of them with the ABFCM. Definition and mathematical formalization of the ABFCM is given in the Section 3. This section also provides short subsection on agent theory emphasizing agent properties like autonomy, data decentralization, and asynchronous calculations. The FCM is a synchronous and centralized system. Concepts have to change state synchronously. The knowledge about concepts values is located in one place. Since MAS is asynchronous, synchronization is implemented through agent behavior and ontology described in the Section 4. In MAS, data is decentralized and, each agent has a part of the knowledge of the overall MAS, so detecting FCM stable state is not centralized and is made through agent behavior and ontology. An ABFCM behaves identically to a classic FCM if the system observed by means of a FCM model conducts in such a way that the concepts do not require the different inference algorithm. But if the map concepts (or the real world elements that they represent) require different inference process, then the ABFCM is the answer. Two examples of such system are presented in Section 5 with ABFCMs used to qualitatively model systems. Chosen systems concepts properties require different inference process in map nodes. The classic FCM defaults to the same inference process, whereas the ABFCM stimulates the use of a different algorithm for the new state values generation of each concept.

2. FUZZY COGNITIVE MAP

Fuzzy cognitive map is a qualitative modeling and simulation technique, developed by Bart Kosko, which models system as the group of concepts and cause-effect relations among concepts.[1] A FCM is a combination of a conceptual map and fuzzy logic providing more realistic and more accurate real word portrait than a conceptual map. A FCM is represented as a weighted, directed graph.[2] The map concepts are graph nodes, whereas the cause-effect relations are depicted as graph directed edges. Cause-effect relations are regarded as linguistic variables with membership functions defined by the expert designing the system map.[8] The fuzzy logic is usually implemented directly by the expert using process of fuzzification and defuzzification to convert real system values to a FCM concept and FCM cause-effect relations weight factors and back. It is implied that the absolutely large cause-effect relation weight factor means the strong cause-effect impact among concepts.[1,9] Fuzzy cognitive map is a method of representing the knowledge. It is also an inference process that generates conclusion about the system behavior.[10] Inference process puts the FCM in the area of the artificial intelligent systems. The FCM inference process is similar to the neural network inference process; therefore, FCM can be considered as the combination of the fuzzy logic and neural networks. In its simplest form, the inference process is defined with the inference algorithm defined in Equation 1. It is a manipulation of two matrices, the concept vector and the adjacency matrix. The concept vector contains FCM nodes values, whereas the adjacency matrix carries cause-effects values among nodes.[11] If A_i^t is the value of ith node at the discrete time step t, and w_{ji} is ith column element in the adjacency matrix, then A_i^t is calculated according to the Equation 1.

$$A_i^t = f\left(\sum_{j=1, j\neq i}^{n} A_j^{t-1} w_{ji}\right) \tag{1}$$

A_j^{t-1} is the value of jth node at the discrete time step $t-1$; f is transformation function, used to normalize the value of a node to the interval $[-1, 1]$; n is the total number of map nodes. The expert, building the map, has to identify map nodes, cause-effect relations and their weights, and also mapping functions between real system values and interval $[-1, 1]$, since the node values are limited to the $[-1, 1]$ interval. The inference algorithm can use different transformation functions depending on the modeled system. The transformation function can be discrete or continuous. The value of the observed concept depends on other concepts and cause-effect relations. The transformation function also strongly affects the inference algorithm.[12]

When a FCM map is used for system simulation, the map is started with the initial nodes values. Inference process generates new values. Conclusion on system behavior is made when a stabile state is reached. In the stabile state, nodes values remain the same if the map fixed-point attractor is revealed. The stabile state is also reached if the nodes' values repeat in limit cycle. Continuous transformation function can result in unstable state, called chaotic attractor.[13] If a map is in unstable state, the conclusion on system behavior can not be made.[13] The unstable state is undesirable map behavior, because it does not provide conclusion about behavior of the system modeled with map. Developed ABFCM exhibits same stability issues like the classic FCM. The stability issue is not in focus of this research. Similar solutions[12] to stability issues in the classic FCM can be applied to the ABFCM.

What if map nodes need different transformation functions to simulate the real concept behavior? For example, one concept is discrete (air condition turned on or off) and another is continuous (room temperature). The classic FCM does not support variation of the inference algorithm among map concepts. All concepts use the same inference algorithm. It is hard to use even different transformation functions in inference algorithms of map concepts, nevertheless to use completely different inference algorithms. Agent-based fuzzy cognitive map enables the use of a different algorithm for the new state value generation of each concept. This ABFCM characteristic supervenes from the agent autonomy characteristic and is direct outcome of injecting the concept of MAS to the FCM.[14]

2.1. Fuzzy Cognitive Map Extensions and Variations

Different extensions and variations of the classic FCM already exist, providing upgrades of the classic FCM properties. FCM extensions and variations are defined as new FCM types. Certain FCM types enable combination of several maps into one; others take into account time discrepancy among concepts, while others include concept history into inference algorithm. But none of the FCM types, known to the author, supports completely different inference algorithms in the each map node for the new node state calculation.

Some of the defined FCM extensions and variations are:

- Combined fuzzy cognitive map.[5]

Combined fuzzy cognitive map merges knowledge from several fuzzy cognitive maps obtained by different experts according to:

$$W = \frac{1}{\sum_{k=1}^{n} j_k} \sum_{k=1}^{n} j_k W_k \tag{2}$$

where j_k is kth expert credibility weight because experts don't have to be equally scored; W_k is the adjacency matrix of the kth fuzzy cognitive map; n is the number of combined fuzzy cognitive maps; and W is the adjacency matrix of the resulting fuzzy cognitive map. The expert credibility weight is usually same for all the experts, that is, $j_k = 1$. The combined fuzzy cognitive map can provide better results because the map knowledge is combined from several experts.

- Adaptive fuzzy cognitive map.[15]

Adaptive fuzzy cognitive map enables adaptation of cause-effect relation weight factors, most often with the Hebbian low learning[5,15]:

$$\dot{w}_{ji} = -w_{ji} + \dot{A}_i \dot{A}_j \tag{3}$$

or in discrete form:

$$w_{ji}(t+1) = \begin{cases} w_{ji}(t) + c_t(\Delta A_i(t)\Delta A_j(t) - w_{ji}(t)) & ako\ je\ \Delta A_i(t) \neq 0 \\ w_{ji}(t) \quad ako\ je\ \Delta A_i(t) = 0 \end{cases} \tag{4}$$

where $\Delta A_i(t) = A_i(t) - A_i(t-1)$ is change of the concept A_i value in t time moment compared to the value in $t-1$ time moment, c_t is learning factor obtained with Equation 5.

$$c_t(t_i) = 0.1\left[1 - \frac{t_i}{1.1N}\right] \tag{5}$$

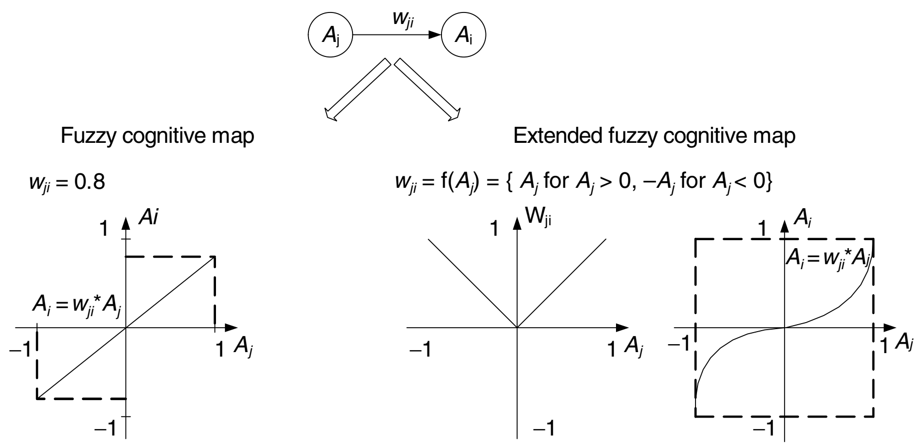

Figure 1. Comparison of fuzzy cognitive map with the cause-effect relation weight factor $w_{ji} = 0.8$ and extended fuzzy cognitive map with the weight factor as a function $w_{ji} = f(A_j) = \{A_j$ for $A_j > 0, -A_j$ for $A_j < 0\}$.

N is number of iteration passed from the map start.

- Extended fuzzy cognitive map.[8]

Extended fuzzy cognitive map introduced function as a cause-effect relation instead of the cause-effect relation weight factor to avoid linearity in a cause-effect relation (Figure 1).

- Fuzzy time cognitive map.[16]

Time lag is supported in the fuzzy time cognitive map. Expert defines time lag as a linguistic variable with accompanying fuzzy values. For example, time lag can be defined with the set {I, N, L}—"immediate," "normal," and "long" that has attached values {1, 2, 3}, where 1 is time lag of 1 time units (for example, around 1 month), 2 is time lag of 2 time units, etc. Time lags are stated in parenthesis in Figure 2.

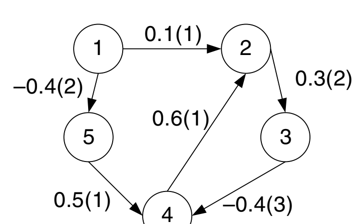

Figure 2. Fuzzy time cognitive map.

Time lags have to be normalized to 1 inserting dummy nodes. "Dummy" nodes can not undermine existing cause-effect relations because time lag does not influence the value or the sign of the cause-effect weight factor. New "dummy" nodes are connected with new "dummy" cause-effect relations with the same weight factor as original cause-effect relation weight factor. Time lag for the map edge e_{ij} of m time units is replaced with the m $-$ 1 "dummy" nodes.

- Rule-based fuzzy cognitive map.[17]

Rule-based fuzzy cognitive map (RBFCM) combines if-then fuzzy rules with the classic map. The fuzzy rule system is extended with feedback and causal-effect relationship. This map supports implication, negation, etc., briefly any relation that can be represented with fuzzy rules. The map is iterative as classic map, so the concept values are calculated through iterative steps. Fuzzy rules base defines relations among concepts.

- Certainty neuron fuzzy cognitive map.[18]

Certainty neuron fuzzy cognitive map (CNFCM) introduced memory into concepts. Each concept has knowledge about previous states. The concept new state value is based on causal-effect relations from other concepts and on concept previous states according to:

$$A_i^{t+1} = f\left(S_i^t, A_i^t\right), \quad S_i^t = \sum_{\substack{j=1 \\ j\neq i}}^{n} A_j^t W_{ji} \tag{6}$$

- Agent-based fuzzy cognitive map.[14]

Agent-based fuzzy cognitive map enables the use of a completely different inference algorithm for the each node. In ABFCM map, one concept can behave like certainty neuron and other one like classic FCM concept. Agent-based fuzzy cognitive map also supports introduction of fuzzy rules into individual concepts' inference algorithm like in rule-based fuzzy cognitive map, cause-effect function relations like in extended fuzzy cognitive map, etc. Agent-based fuzzy cognitive map is a new FCM type that integrates other FCM types into one; ABFCM enables the implementation of different existing inference algorithms in each map node; and ABFCM also supports introduction of new ideas in concepts' inference algorithm. Each concept in an ABFCM map can behave as a classic FCM map concept or can implement some of the existing FCM variations and extensions or can implement completely new inference algorithm. Each causal-effect relation in an ABFCM map can behave as a classic FCM map causal-effect relation with the weight factor or can implement some of the existing FCM variations and extensions.

3. AGENT-BASED FUZZY COGNITIVE MAP

Nowadays, agents are used for different applications ranging from the mail filtering system with just one agent to the air traffic control, industry production optimization control, telecommunication network monitoring and control, Internet data gathering and filtering with large MAS.[19,20] Agent is a software system situated in the environment; it can receive environmental stimulus and can react flexibly and autonomously in pursuing its goals. A MAS can be defined as a network of entities working together on solving the problem that is beyond the individual agent's solving capabilities and knowledge.[19] Multi-agent system properties are: Each agent has a partial information or problem solving capability; there is no global system control; data is decentralized; and calculations are asynchronous.[19]

The agent interaction is one of the main issues concerning MAS. An interaction is a chain of agent's actions pursued in dynamic connection with other agents. The interaction influences the future agent's behavior. Because the interaction demands some kind of a collective knowledge and a vocabulary, each agent implementation domain requires the definition of its joint ontology. The ontology is the explicit knowledge conceptualization.[21] The formal basis for the knowledge representation and description is the conceptualization of the knowledge, containing definitions of the domain entities, concepts, objects (however one wants to call them), and their relationships. A conceptualization is an abstract and simplified view of the real world. Each knowledge-based system or agent possessing some knowledge relies on a knowledge conceptualization.[22] Multi-agent systems has a defined structure according to the Foundation of Intelligent Physical Agents (FIPA—IEEE Computer Society standards organization) specifications.[23] The implementation language is arbitrary as long as the multi-agent architecture requirements are complied. Defined abstract architecture is the basis for developing a concrete MAS. It contains the description of multi-agent services that agents can rely on in a MAS. Since the Java Agent Development Framework (JADE) complies with the set standards,[24] it is used to build the ABFCM prototype used for experimental results in Section 5.

The idea of combining FCP and the agent-based technology resulted in the ABFCM.[14] An ABFCM is a FCM with each concept mapped into the agent. The cause-effect relations manifests as communication messages carrying cause-effect information among the agents (Figure 3).

DEFINITION 1. *The set of all FCM concepts is denoted as* $A = \{A_1, A_2, \ldots, A_n\}$. *The set of the FCM cause-effect relations is denoted as* $W = \{W_{1k}, W_{1n}, \ldots, W_{nm}\}$. *The set of agents is denoted as* $Ag = \{Ag_0, Ag_1, \ldots, Ag_{n-1}\}$. *The ABFCM is obtained as the one-way mapping function from set A to the set Ag, thereby providing two-way communication among agents with a cause-effect relation:*

$$f_{ABFCM} : A \to Ag \tag{7}$$

An ABFCM behaves the same as a classic FCM if map concepts don't require the different inference algorithms. When the real system concepts modeled with the map concepts, the behavior is similar, that is the case.

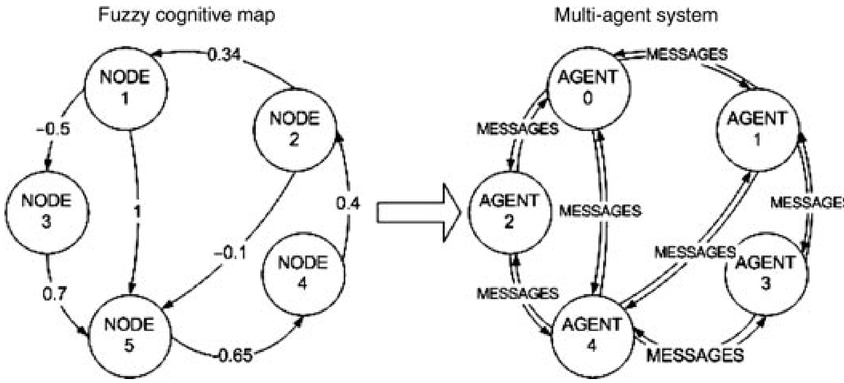

Figure 3. Mapping of the FCM into the ABFCM.

To obtain a MAS that behaves like a FCM the agent behavior modeling, the ontology definition, the synchronization of the agent map, and the agent coordination had to be designed. Developed design is explained in the next sections.

Real systems, observed by means of a FCM model, often conduct in such a way that map concepts require the different inference algorithm. The classic FCM uses the same inference algorithm in each map node. The ABFCM intrinsically supports the use of a different algorithm for the new state values generation of each concept. The Equation 1 for calculating the node value, in the classic FCM, becomes Equation 8 in the ABFCM.

$$A_i^t = F_i\left(A_1^{t-1}, \ldots, \quad A_j^{t-1}, \ldots, \quad A_n^{t-1}, \quad w_{11}, \ldots, \quad w_{ji}, \ldots, \quad w_{nn}\right) \qquad (8)$$

The A_i^t is the value of ith node at the discrete moment t; A_j^{t-1} is the value of jth node at the discrete moment $t-1$; w_{ji} is jth element in the ith column of the adjacency matrix; n is the total number of map nodes. F_i is the inference process that is using provided information to generate new state of the A_i^t node. The inference process in the ith node, marked F_i, is node individual characteristic. Each node can use different inference process. The difference can be just in different transformation function, like in the example of three-coupled tanks presented in the Section 5, but it can go much further. Example of the inverted pendulum ABFCM, also presented in the Section 5, has four nodes. All four nodes in the map use different inference processes, one node uses classic FCM inference process, other uses CNFCM inference process, etc. The ABFCM supports different inference process in each map node. Real world system often requires different inference process in map nodes and ABFCM makes that possible and simple.

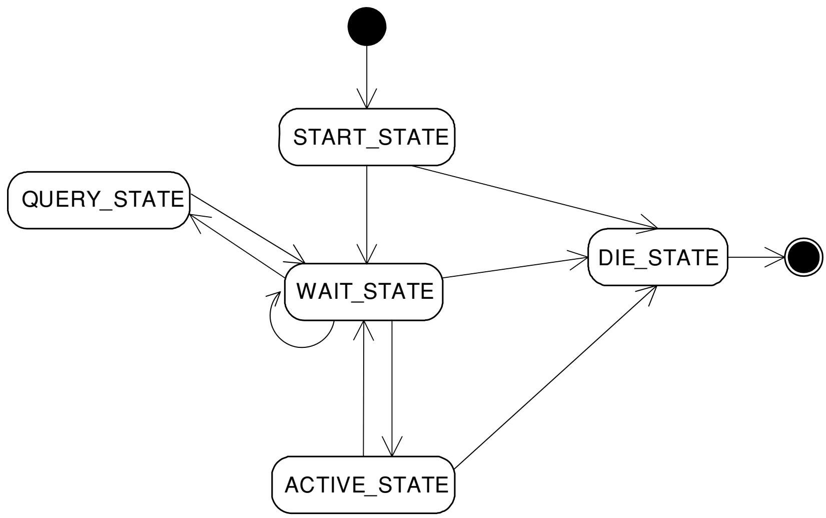

Figure 4. The ABFCM agent behavior states and transitions.

4. AGENT-BASED FUZZY COGNITIVE MAP AGENT

4.1. Behavior

The ABFCM agent needs to emulate a FCM node behavior. Basic features include the initial node state, the new node state generated by the cause-effect relations with other nodes, and the conclusion carried out with the FCM limited cycle or fixed point. The agent behavior is modeled as finite automata with states representing FCM behavior and states that are necessary for MAS functionality (Figure 4).

Figure 4 shows the ABFCM agent's states and transitions. START_STATE is the agent's starting state that includes the initial node state. WAIT_STATE is necessary for the MAS functionality. An agent is in that state when nothing occurs in the system that has any relevance for the agent. If the agent receives a message with a content of interest (the message from another ABFCM agent) it shifts to the other states. QUERY_STATE is the state to which an agent transits upon receiving the message with *Query* action in the content. The agent reply comprises *Answer* action with the current agent/node value in the message content. ACTIVE_STATE is primarily used for calculating the new agent/node value state. An agent enters this state after receiving a message with the *Change* action in the content from another agent that has just changed value and is affecting the agent in stake. The agent needs values from all agents that are influencing it to calculate the new value. Therefore, the agent sends the *Query* message to all agents that are affecting it to obtain those values. DIE_STATE is the agent's final state. An agent enters this state if an error has occurred in the agent behavior or if the map has entered a limit cycle or a fixed point. There is no need for the MAS to continue running after that.

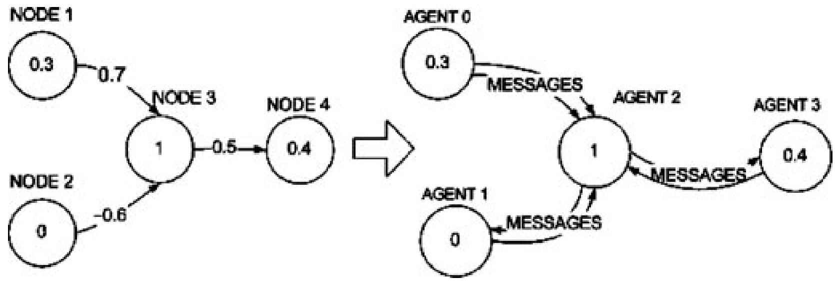

Figure 5. The part of the ABFCM obtained from the part of the FCM.

4.2. Agent Naming

Because an FCM can have different, theoretically unlimited, number of nodes, a unique agent naming convention is defined. This is necessary to enable agents to locate each other in the MAS. An agent in the ABFCM is named concatenating the string *agent* and the agent identifier. Identifier is an integer number starting from 0 and increased by 1 for the next agent (node).

Figure 5 shows the part of the FCM with the four concepts transferred to the ABFCM. Node1 is mapped to agent0, node2 is mapped to agent1, node3 is mapped to agent2, and node4 is mapped to agent3.

4.3. Synchronization

The MAS is an asynchronous system because the message exchange and the speed of the agent behavior execution are not synchronous. There is no way to know in advance when the agent will start or finish an action, when the agent will create message and send it to the other agents, and so on. The FCM node state calculation in a discrete moment is based on the nodes state in the previous discrete moment. A FCM is basically a synchronous system because nodes values have to change synchronously in a discrete time steps. This means that the two nodes having a cause-effect relation cannot be in the discrete time steps that vary for more than 1 step. If the successive discrete time steps are annotated with $\{k_0, k_1, \ldots, k_n\}$, for example, in Figure 5, it is not possible for node1 to be in step k_4 and for node3 to be in step k_8. To obtain FCM properties in MAS, it is necessary to implement the synchronization. Synchronization is implemented through the agent behavior and the ABFCM ontology.[14] Each agent possesses the synchronization information called the agent tact. An agent uses that information to adjust the behavior and to harmonize the time moment with the other agents in the ABFCM.

4.4. ABFCM Ontology

Every MAS uses some kind of a ontology. Any software that does anything useful cannot be written without a commitment to a model of the relevant world, to entities, properties, and relations in that world.[22] General ontology does not exist so it is common to develop the special case ontology.[21] The ABFCM ontology is designed by inspecting domain of discourse by the knowledge engineer.[25] ABFCM ontology is developed using ontology development tool PROTÉGÉ-2000. The main concepts are identified and incorporated into modeling concepts of the PROTÉGÉ framework. PROTÉGÉ-2000 knowledge model is based on frames and first order logic. The main modeling components are classes, slots, facets, and instances.[26,27] The ABFCM ontology was defined with concrete classes and their slots. The ABFCM ontology contains the four agent action classes translated from PROTÉGÉ-2000 concrete classes:[14] *Change*, *Answer*, *Query,* and *Die*.

Change is the action the agent uses to inform other agents about changes of the node that agent is representing. The PROTÉGÉ-2000 slots *value* and *tact* are translated to the Java class properties. The slot *value* of the *Change* class holds the value of the FCM concept. The slot *tact* of the *Change* class holds the tact of the FCM concept used for the synchronization of ABFCM agents. When an agent receives message with *Change* action in the message content, it transits to ACTIVE_STATE and process received information to calculate the new state. *Answer* is the action that agent uses to answer queries about its value. This class contains the same properties, *value* and *tact*, as the *Change* action. *Query* is the action that enables the agent to put the query to the other agents about their state. This class, like the *Answer* class, contains the same properties as the *Change* action, *value* and *tact*. When an agent receives message with *Query* action in the message content, it transits to QUERY_STATE and generates the message with *Answer* action in the message content to answer the agent querying him. *Die* action agents use to stop the agent map. The *Die* class slots, *name* and *tact*, are translated, during the conversion, into Java class properties. The *name* slot holds the name of the agent that has generated the message with the *Die* action. The agent, generating message with the *Die* action, has concluded that ABFCM map has to terminate because an error has occurred. When agent receives message, from other agent, with *Die* action in the message content, it transits to DIE_STATE and terminates its behavior. The ABFCM ontology also contains one predicate, *LimitCycle*, which agents use to exchange the message about the limit cycle in the ABFCM. When an agent receives message with *LimitCycle* predicate in the message content, it transits to ACTIVE_STATE and tries to detect if the map has reached the limit cycle. The JADE platform defines it own content reference model shown in Figure 6[24] along with elements used in the ABFCM ontology. Each content reference model element has its particular role in the content model. Concepts are type of terms, entities with a complex structure that can be defined in terms of slots. Agent actions are special concepts that indicate actions that can be performed by some agents. Communicative acts of FIPA ACL messages are themselves agent actions.

5. EXPERIMENTAL RESULTS

Experimental results are presented with two nonlinear systems modeled with the ABFCM, an inverted pendulum and a three-coupled tanks system. The inverted

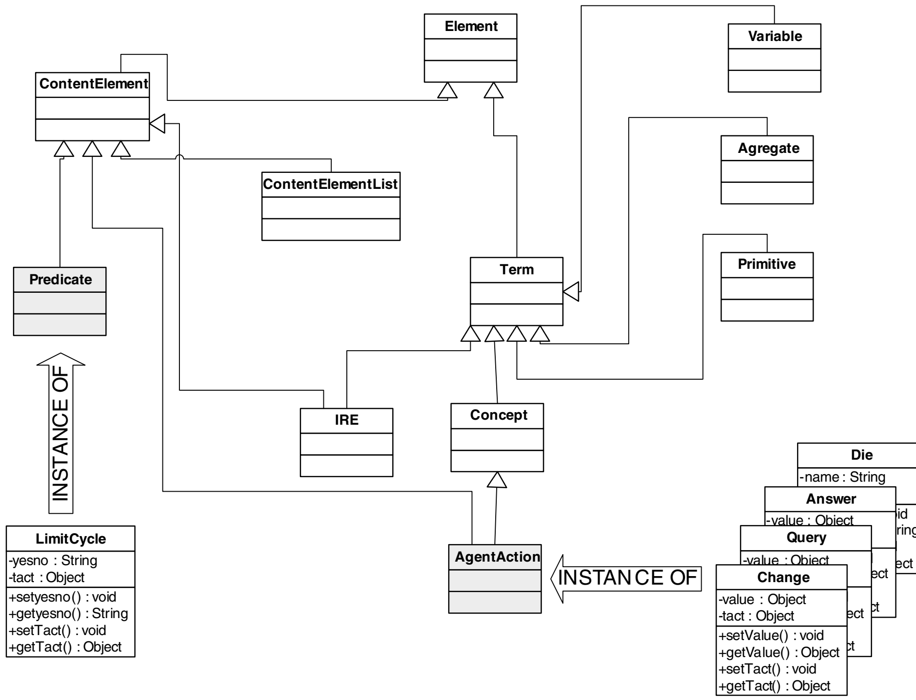

Figure 6. JADE content reference model.

pendulum system is often used with different simulation methods,[28] so results can be compared with ABFCM results. These examples are also chosen because nonlinear system is more likely to have different inference algorithms in the map nodes than a linear system because it is more complex.[29] Chosen systems can also be modeled mathematically according to motion Equation 9 for the inverted pendulum and Torricelli Equation 10, defining the tank fluid outflow based on gravity force and mass preservation low, for the three-coupled tanks system. Matlab models, developed according to Equations 9 and 10 are used to obtain simulation results for the comparison with ABFCM results (Figures 9 and 10).

$$(M+m)\ddot{y}+ml\cos\Theta\,\ddot{\Theta}-ml\,\dot{\Theta}^{2}\sin\Theta=f$$

$$ml\cos\Theta\,\ddot{y}+ml^{2}\,\ddot{\Theta}-mgl\sin\Theta=0 \tag{9}$$

The inverted pendulum system with accompanying ABFCM is shown in Figure 7. There are four main concepts:

- A_1—applied force f.
- A_2—stick offset Θ.
- A_3—platform horizontal offset y.

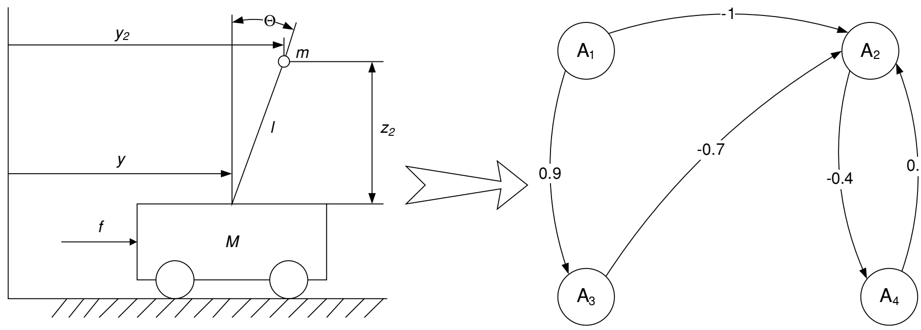

Figure 7. The inverted pendulum with accompanying map.

- A_4—elastic bond intensity between the platform and the stick.

$$A_1 \frac{dh_1}{dt} = q_1 - q_2 = q_1 - A_{01} \times \sqrt{2 \times g \times h_1}$$
$$A_2 \frac{dh_2}{dt} = q_2 + q_3 - q_4 = q_2 + q_3 - A_{02} \times \sqrt{2 \times g \times h_2} \tag{10}$$
$$A_3 \frac{dh_3}{dt} = q_4 - q_5 = q_4 - A_{03} \times \sqrt{2 \times g \times h_3}$$

The three-coupled tanks system with accompanying ABFCM is shown in Figure 8. There are eight main concepts:

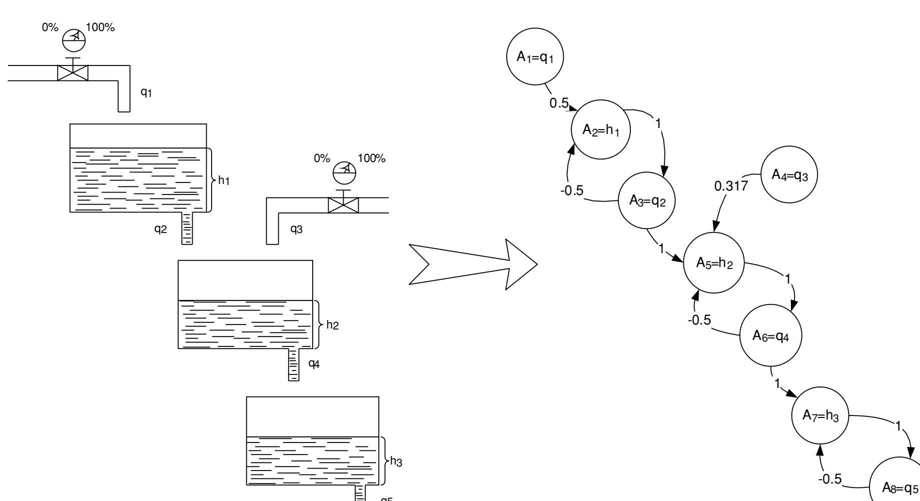

Figure 8. The nonlinear three-coupled tanks system with accompanying map.

- A_1—first tank input flow q_1.
- A_2—first tank flow height h_1.
- A_3—first tank outflow and second tank input flow q_2.
- A_4—second tank input flow q_3.
- A_5—second tank flow height h_2.
- A_6—second tank outflow and third tank input flow q_4.
- A_7—third tank flow height h_3.
- A_8—third tank outflow q_5.

Map concepts, cause-effect relationships, and weight factors strongly depend on the system physical values and are identified by the system expert as in classic FCM.[2,13,30]

For the inverted pendulum system, physical values used are $M = 20$ kg, $m = 1$ kg, $l = 2$ m, $g = 9.81$ m/s^2, $f = 10$ and 0.1 N (step impulse with the increase time $T = 1$ s). When the force is acting on the pendulum, the pendulum moves in the direction of the force so that cause-effect relationship is positive and is weighted with the 0.9 factor by the expert. The same force causes stick offset in the opposite direction so that cause-effect relationship is negative (-1). The pendulum offset affects the stick offset in the same manner and that cause-effect relationship is also negative (-0.7). The relationship between the stick offset angle and the elastic bond intensity between the platform and the stick contains a so-called deviation counteracting cycle. The increased angle causes the bond intensity to decrease (-0.4). The bond-intensity increase increases the angle (0.6). This leads to the adjacency matrix:

$$W = \begin{bmatrix} 0 & -1 & 0.9 & 0 \\ 0 & 0 & 0 & -0.4 \\ 0 & -0.7 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \end{bmatrix} \tag{11}$$

The concepts in the map are limited to the $[-1, 1]$ interval, therefore, the real concept values have to be mapped to the $[-1, 1]$ interval. The mapping between real values and the interval $[-1, 1]$ is also defined by the expert. The mapping function g_{A_2} between real concept value and interval $[-1, 1]$ is defined in Equation 12.

$$\begin{aligned} g_{A_2} : IM_{A_2} \to RC_{A_2}, \quad IM_{A_2} &= \{x : -1 \leq x \leq 1\}, \\ RC_{A_2} &= \{x : -90^0 \leq x \leq 90^0\} \\ g_{A_2}(x) &= 90x, \ -1 \leq x \leq 1 \end{aligned} \tag{12}$$

The force concept has a nonlinear mapping function g_{A_1}, defined in Equation 13. The negative function values refer to the direction of the function.

The function g_{A_1} is also defined by the expert and fine tuned experimentally comparing the map results with the Matlab model results. The mapping is relatively complex although the force real values are limited between minimal value 0.1N and maximal value 10N with inclusion of the value 0N. This is a consequence of the real concept characteristics. They are what they are and the task of the expert, developing the map, is to identify concept characteristics. Presented map with identified concepts, cause-effect relations, weight factors, and mapping functions between real concept values and interval $[-1, 1]$, is used in this example and results are obtained with that map.

$$g_{A_1} : IM_{A_1} \rightarrow RC_{A_1}, IM_{A_1} = \{x : -1 \leq x \leq 1\},$$

$$RC_{A_1} = \{x : 0.1N \leq x \leq 10N \ \& -10 \leq x \leq -0.1N \ \& \ x = 0\}$$

$$g_{A_1}(x) = \begin{cases} -0.9783x - 1.0783, -1 \leq x \leq -0.08 \\ -180x - 15.4, -0.08 < x \leq -0.03 \\ -10, -0.03 < x < 0 \\ 0, x = 0 \\ 10, 0 < x < 0.03 \\ -180x + 15.4, 0.03 \leq x < 0.08 \\ -0.9783x + 1.0783, 0.08 \leq x \leq 1 \end{cases} \tag{13}$$

Other concepts are not interesting for the observation, as it is custom to observe just the stick angle according to the applied force; therefore, other concepts mapping functions are not discussed. Results obtained with the map are translated to the real system values using Equations 12 and 13 and are compared to the results obtained with the Matlab model.

The defined map is tested with the ABFCM prototype* developed to test the ABFCM characteristics. The map is first tested for the force $f = 0.1$. The concept A_1 starting value is set to 1, whereas all the other concepts are set to 0. These values are obtained from the real values using inverse functions of the functions defined in Equations 12 and 13.

The transformation function used in this example is *tanh*. This function is selected experimentally according to the real concepts values that are continuous. Results obtained with the map with the starting concept vector $A^{t=0} = [1\ \ 0\ \ 0\ \ 0]$ are presented in the Table I. In the stable state, at the 8$^{\text{th}}$ map step, the concept A_2

Table I. Results obtained with the concept vector $A^{t=0} = [1\ \ 0\ \ 0\ \ 0]$.

	A_1	A_2	A_3	A_4
Starting state (tact k = 0)	1.0	0.0	0.0	0.0
	1.0	−0.7615	0.7162	0.0
	1.0	−0.9053	0.7162	0.2955
	1.0	−0.8677	0.7162	0.347
	1.0	−0.8599	0.7162	0.3337
	1.0	−0.862	0.7162	0.331
	1.0	−0.8624	0.7162	0.3317
	1.0	−0.8623	0.7162	0.3318
Ending state (tact k = 8)	1.0	−0.8623	0.7162	0.3318

has a value -0.8623. The real stick offset Θ value, represented with concept A_2, is 77,607°. This means that the stick offset Θ rises to the value 77,607° when applied force f is 0.1N. The negative sign of the concept A_2 (-0.8623) implies direction of the offset angle opposite to the direction of the force.

To establish the stick offset Θ behavior in the next, qualitatively different, behavior step, the map is started with the concept vector $A^{t'=0} = [0 \quad -0.8623 \quad 0.7162 \quad 0.3318]$, that is the stable state from Table I. The stable state in the previous simulation is used as the starting vector for the next simulation. The force f does not affect the platform any more. Therefore, the concept A_1 is 0 in the next simulation steps.

$$A_1^t = 0 \tag{14}$$

The concept A_3 (platform horizontal offset y) has a starting value 0.7162. The movement of the platform is a straight-lined uniformly accelerated motion because the friction is omitted. Therefore, the value of the concept A_3 has to remain constant.

$$A_3^t = A_3^{t-1} \tag{15}$$

This condition is set only for this concept. Other concepts behave differently. The concept A_2 (the stick offset Θ) behaves like CNFCM and differently from other concepts. The moving of the stick affects the system until it equilibrates with the elastic bond intensity between the platform and the stick (A_4) because of inertia. The inference process is given in Equation 16. It includes previous states of the concept to model inertia that affects behavior of the stick offset angle.

$$A_2^t = \tanh\left(\sum_{j=1, j\neq i}^{4} A_j^{t-1} w_{ji} + A_2^{t-1}\right) \tag{16}$$

Again, this is true only for the concept A_2, whereas other concepts behave differently. The concept A_4, the elastic bond intensity between the platform and the stick, has inference process like the classic FCM.

$$A_4^t = \tanh\left(\sum_{j=1, j\neq i}^{4} A_j^{t-1} w_{ji}\right) \tag{17}$$

All four nodes in the map use different inference processes given in Equations 14–17. The next, qualitatively different, behavior step of the map is presented in Table II.

The A_3 concept value has to remain constant, but it has a little change because of the *tanh* function rounding. This is corrected in the next step of establishing the system behavior. The system qualitative behavior is caught repeating the procedure of starting the map with the previously obtained steady state map. In the next step of establishing the system behavior, the map is started again with the concept vector

Table II. Results obtained with the concept vector $A^{t'=0} = [0\ \ -0.8623\ \ 0.7162\ \ 0.3318]$.

	A_1	A_2	A_3	A_4
Starting state (tact k = 0)	0.0	−0.8623	0.7162	0.3318
	0.0	−0.8225	0.6145	0.3318
	0.0	−0.7831	0.6145	0.3176
	0.0	−0.7709	0.6145	0.3033
	0.0	−0.7694	0.6145	0.2989
	0.0	−0.7699	0.6145	0.2983
	0.0	−0.7703	0.6145	0.2985
	0.0	−0.7704	0.6145	0.2987
Ending state (tact k = 8)	0.0	−0.7704	0.6145	0.2987

$A^{t''=0} = [0\ \ -0.7704\ \ 0.7162\ \ 0.2987]$. That is, steady state from Table II (last row), with the corrected A_3 ($0.6145 \rightarrow 0.7162$) concept error caused by *tanh* function rounding. Stable states, obtained in the four steps of establishing the system behavior, and initial concept vector are shown in Table III. In each step of establishing the system qualitative behavior, the map is started with the steady state obtained in the previous step of establishing the system behavior. The system qualitative behavior implies the behavior of the system when concepts in the system qualitatively change. For example, the Table I contains the map results for applied function $f = 0.1$N, that is, qualitative behavior of the system for applied function $f = 0.1$N. The behavior of the system, qualitatively speaking, is that stick offset Θ rises to around 77°. Table II contains map results when the function stops. The behavior of the system, qualitatively speaking, is that stick offset Θ lingers around 69°.

Results from Table III are translated to the real values in Table IV just for A_1 and A_2 concepts that are observed.

Results obtained with the ABFCM are compared with the Matlab model results in Figure 9a for the force $f = 0.1$N and in Figure 9b for the force $f = 10$N. Figures show behavior of the Θ depending on a time after the force initial action.

The Matlab results are that Θ almost immediately starts to oscillate around -1° after initial value around -3° for the force of 10N; and for the force of 0.1N, the offset Θ of the stick rises to -75° and then oscillates around that value. Negative

Table III. Stable states of the ABFCM map started with the initial concept vector $A^{t=0} = [1\ \ 0\ \ 0\ \ 0]$.

A_1	A_2	A_3	A_4
1.0	0.0	0.0	0.0
1.0	−0.8623	0.7162	0.3318
0.0	−0.7704	0.6145	0.2987
0.0	−0.7703	0.6145	0.2987
0.0	−0.7703	0.6145	0.2987

Table IV. Real values of concepts A_1 and A_2 obtained from the ABFCM results in Table III.

A_1 [N]	A_2 [°]	A_3	A_4
0.1	0.0	Not mapped	Not mapped
0.1	−77,607	Not mapped	Not mapped
0.0	−69.336	Not mapped	Not mapped
0.0	−69.327	Not mapped	Not mapped
0.0	−69.327	Not mapped	Not mapped

Θ values relate to the stick offset direction, that is, opposite to the direction of the force f. The results obtained with the ABFCM are qualitative and discrete. For comparison with the quantitative and continuous Matlab results, they have to be adjusted. The map results were expanded over the time line to conform to the corresponding Matlab results. For the force $f = 0.1$N, the ABFCM model initially shows increase of the stick offset Θ in the opposite direction of the force, hence the negative value. Then the ABFCM model shows the angle stabilization around some value. For the force $f = 10$N the ABFCM model shows the increase of the stick offset Θ and then reaching value retention. Qualitatively observed, similar results are obtained with the Matlab model.

The more timely behavior, such as the detailed oscillations of the stick-offset angle, cannot be acquired with the map, because the map is a qualitative model of the system. However, qualitative behavior of the system is rather accurately captured with the ABFCM. The inverted pendulum system can be modeled quantitatively; therefore, there is no need for a qualitative model. The inverted pendulum system has been used to compare the qualitative results obtained with the ABFCM with the real, quantitative Matlab model system behavior.

The second system modeled with the ABFCM is the three-coupled tanks. The observed system values are h_1, h_2, and h_3 liquid height in the tanks; and the input system values are first tank input flow $q_1 = [0–25]$ m^3/s and second tank input flow

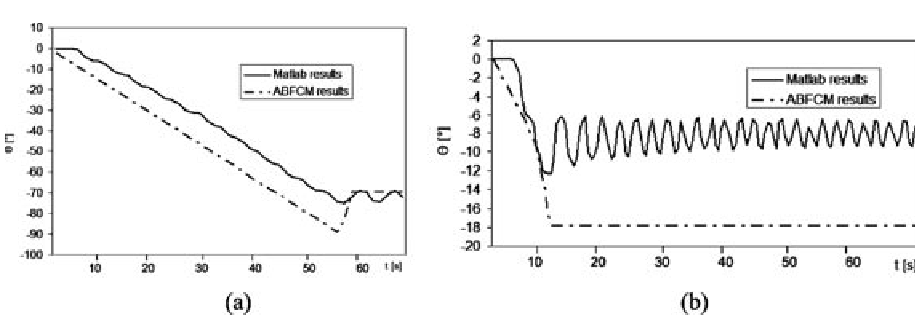

Figure 9. (a) The inverted pendulum ABFCM results compared with the results obtained with Matlab system model for $f = 0.1$N and (b) the inverted pendulum ABFCM results compared with the results obtained with Matlab system model for $f = 10$N.

$q_3 = [0\text{–}25]$ m^3/s. Physical values affecting cause-effect relationships are tanks sectional area (25 m^2 for all three tanks) and outflow pipes sectional area (2 m^2 for all pipes). Map concepts, cause-effect relationships, and weight factors strongly depend on the system physical values and are again identified by the system expert. This leads to the adjacency matrix:

$$W = \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.317 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.5 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.5 & 0 \end{bmatrix} \tag{18}$$

The mapping between real values and the interval $[-1, 1]$ is also defined by the expert.

$$\begin{gathered} q_1(x) = q_3(x) = \frac{1}{25} \times x, x \in [0, 25]; \quad q_2(x) = \frac{0.32}{25} \times x, x \in [0, 25]; \\ q_4(x) \approx \frac{0.26}{31.3} \times x, x \in [0, 31.3]; \quad q_5(x) \approx \frac{0.17}{31.3} \times x, x \in [0, 31.3]; \\ h_1(x) \approx \frac{\sqrt{x}}{12}, x \in [0, 15.9];\ h_2(x) \approx \frac{\sqrt{x}}{18.9}, x \in [0, 25]; h_3(x) \approx \frac{\sqrt{x}}{28.3}, x \in [0, 25] \end{gathered} \tag{19}$$

In three-coupled tanks system, the difference in inference algorithm for input flows and outflows and flow heights is exhibited only in different transformation functions. Nodes representing flows require a transformation function like in Equation 19 while nodes representing flow heights use a transformation function like in Equation 20.

$$f(x) = \begin{cases} -1 \times \tanh(x), & \tanh(x) \le 0 \\ \tanh(x), & \tanh(x) > 0 \end{cases} \tag{20}$$

$$f(x) = \tanh(x) \tag{21}$$

The three-coupled tanks system input flows and outflows and flow heights modeled with the Matlab reach stable state fast. For example, when the model is started with the $q_1 = 7.5$ m^3/s (30% of the maximal 25 m^3/s) and $q_3 = 12.5$ m^3/s (50% of the maximal 25 m^3/s), the h_1 reaches 0.72 m, the h_2 reaches 5.1 m, the h_3 reaches 5.1 m, the q_2 reaches 7.5 m^3/s, the q_4 reaches 20 m^3/s, and the q_5 reaches 20 m^3/s in less than 0.2 s. Real system results obtained with the Matlab model for different

Table V. Results obtained with the Matlab system model for different input flows. Input values for the observed system are input flows q_1 i q_3. System response values are flow heights h_1, h_2, h_3 and outflows q_2, q_4, q_5.

q_1 [m^3/s]	h_1 [m]	q_2 [m^3/s]	q_3 [m^3/s]	h_2 [m]	q_4 [m^3/s]	h_3 [m]	q_5 [m^3/s]
25	15.9	25	25	25 spill over	31.3	25 spill over	31.3
25	15.9	25	0	15.9	25	15.9	25
0	0	0	25	15.9	25	15.9	25
12.5	4	12.5	0	4	12.5	4	12.5
7.5	0.72	7.5	12.5	5.1	20	5.1	20
12.5	4	12.5	12.5	15.9	25	15.9	25
0	0	0	12.5	4	12.5	4	12.5
25	15.9	25	6.32	25 no spill over	31.3	25 no spill over	31.3
6.32	1	6.32	25	25 no spill over	31.3	25 no spill over	31.3

Table VI. ABFCM results with the starting concept vector $A^t = 0 = [0.3\ 0\ 0\ 0.5\ 0\ 0\ 0\ 0]$.

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
Starting state (tact k = 0)	0.3	0.0	0.0	0.5	0.0	0.0	0.0	0.0
	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
Ending state (tact k = 35)	0.3	0.0999	0.0995	0.5	0.1714	0.1697	0.1082	0.1077

input flows are shown in Table V. For this example only system stable states are observed.

System behavior simulation with the initial values for the input flow value q_1 of 30% of the 25 m^3/s, and input flow value q_3 of 50% of the 25 m^3/s is done by setting the concepts $A_1 = 0.3$ and $A_4 = 0.5$ according to Equation 19. Table VI presents results obtained with the ABFCM, but only the map starting state and ending, stable state.

Results from the Table VI are translated to the real values, rounded to two decimals, in Table VII.

For example, the interpretation of the first tank outflow q_2 value, that is, concept A_3 value, 0.0995 is real value 7.7734375 according to Equation 19.

System behavior simulation with the initial values for the input flow value q_1 of 50% of the 12.5 m^3/s, and input flow value q_3 of 50% of the 12.5 m^3/s is done by setting the concepts $A_1 = 0.5$ and $A_4 = 0.5$ according to Equation 19. Table VIII presents results obtained with the ABFCM for starting and ending state.

Table VII. Real values of concepts obtained from the ABFCM results in Table VI.

A_1 [m^3/s]	A_2 [m]	A_3 [m^3/s]	A_4 [m^3/s]	A_5 [m]	A_6 [m^3/s]	A_7 [m]	A_8 [m^3/s]
7.5	0	0	12.5	0	0	0	0
7.5	1.43	7.77	12.5	10.49	20.42	9.37	19.82

Table VIII. ABFCM results with the starting concept vector $A^t = 0 = [0.5\ 0\ 0\ 0.5\ 0\ 0\ 0\ 0]$.

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
Starting state (tact k = 0)	0.5	0.0	0.0	0.5	0.0	0.0	0.0	0.0
	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
Ending state (tact k = 40)	0.5	0.1661	0.1645	0.5	0.2142	0.2109	0.1401	0.1391

Table IX. Real values of concepts obtained from the ABFCM results in Table VIII.

	A_1 [m^3/s]	A_2 [m]	A_3 [m^3/s]	A_4 [m^3/s]	A_5 [m]	A_6 [m^3/s]	A_7 [m]	A_8 [m^3/s]
Starting state	12.5	0	0	12.5	0	0	0	0
Ending state	12.5	3.97	12.85	12.5	16.39	25.39	15.72	25.61

Results from the Table VIII are translated to the real values, rounded to two decimals, in Table IX.

Results obtained with the ABFCM are compared with the Matlab model results given in Figure 10a for the input flow q_1 of 12.5 m^3/s and input flow q_3 of the 12.5 m^3/s and in Figure 10b for the input flow q_1 of the 7.5 m^3/s and input flow q_3 of the 12.5 m^3/s.

In this example, system transition is not observed. Figure 10 shows only finite states for both Matlab and ABFCM model. There is no need for ABFCM results adaptation, like in previous example where system transition is observed. Although FCM is qualitative modeling technique, results obtained with the ABFCM for all eight concepts are close to real Matlab results due to the different transformation functions used in concepts. Again the ABFCM map correctly concludes the qualitative behavior of the system.

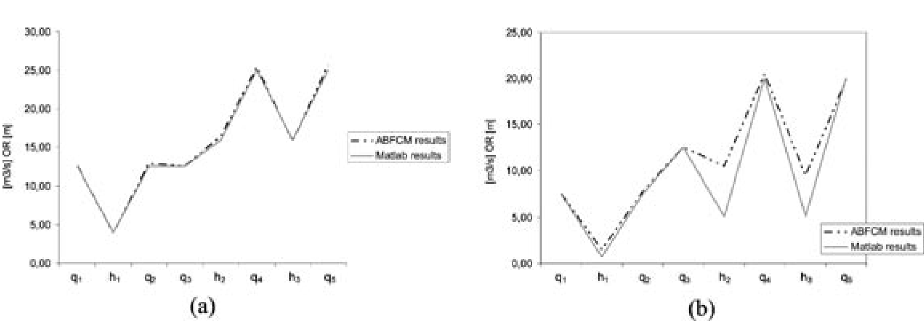

Figure 10. (a) The inverted pendulum ABFCM results compared with the results obtained with Matlab system model for the input flow q_1 of 12.5 m^3/s and input flow q_3 of the 12.5 m^3/s, and (b) the inverted pendulum ABFCM results compared with the results obtained with Matlab system model for input flow q_1 of the 7.5 m^3/s and input flow q_3 of the 12.5 m^3/s.

To relieve the user of the map results interpretation the ABFCM has another new property. The agent interpreter can be introduced in the map to facilitate the use of the ABFCM. The role of the agent interpreter is the automatic interpretation of the obtained results and presentation of the results to the user in graphical or linguistic format. The agent interpreter does the map results interpretation instead of the user, and facilitates the use of the ABFCM for the system modeling and simulation compared to the classic FCM.

Linguistic interpretation of the results from Table VII:

> "If first tank input flow q_1 is 30% of the maximal value (0.3) and second tank input flow q_3 is 50% of the maximal value (0.5), the first tank flow height h_1 will rise to 5.75% of the maximal value, first tank outflow q_2 will be 31% of the maximal value, second tank flow height h_2 and third tank flow height h_3 will be almost the same (around 40%), as well as second tank outflow q_4 and third tank outflow q_5 (around 65.26%, 63.35%)."

Figure 11 shows the graphical interpretation of the results from Table VII obtained with the ABFCM agent interpreter.

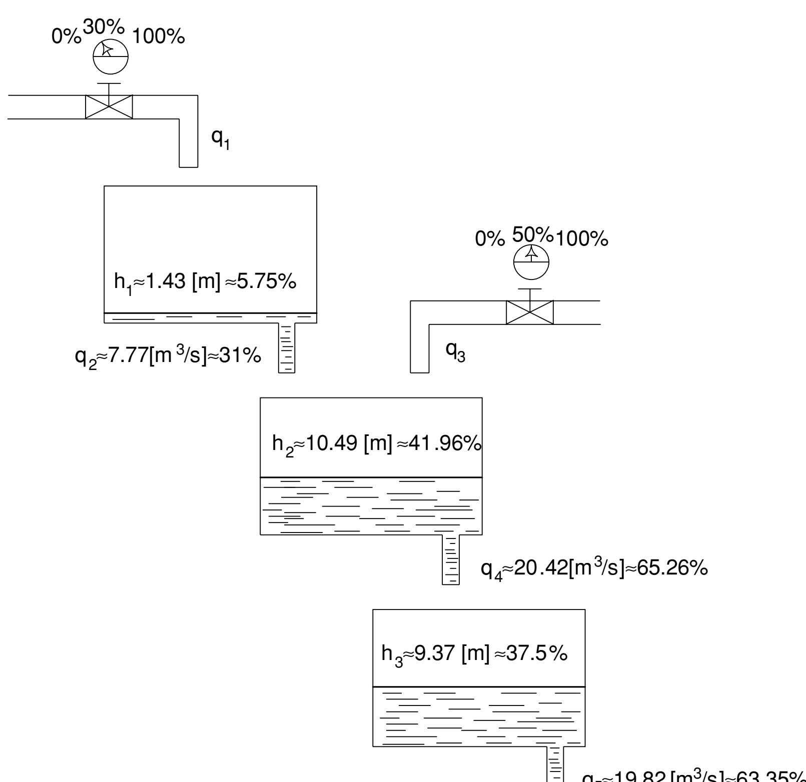

Figure 11. Graphical interpretation of the results from Table VII obtained with the ABFCM agent interpreter.

The agent interpreter does not map to the FCM node like other ABFCM agents, and it may or may not be used in the ABFCM. It is introduced to emphasize how agent technology can enhance the characteristics of the FCM other than by just adding individual characteristics to each concept represented with the agent in the ABFCM.

Another example of the ABFCM use can be found in Ref. 31.

6. CONCLUSION

Fuzzy cognitive map is a qualitative modeling and simulation technique. Real world systems often require different inference algorithm in each map node for the new node state calculation during a system behavior simulation. The classic FCM uses the same inference algorithm in each map node. The agent-based fuzzy cognitive map is a new FCM type developed by combining FCM and MAS technology. Although the definition of the ABFCM is composite and extensive, the use of the ABFCM is simple as the use of the classic FCM. Developed ABFCM software prototype used for examples presented in the article enables the user to define map concepts, relations, and inference algorithms and to start the map and observes obtained results like in the classic FCM.

ABFCM is a fuzzy cognitive map based on a MAS with each concept mapped into the agent. The FCM cause-effect relations are implemented as communication messages carrying cause-effect information among the agents. An ABFCM has all the characteristics of the classic FCM and also has an important, novel characteristic. The novel characteristic supervenes from the multi-agent technology properties that are introduced into FCM. Since an agent is an individual entity, the map node realized as an agent, also possesses individual characteristics. Compared with the classic FCM, ABFCM intrinsically enables each concept to use different inference process, so ABFCM is more appropriate for simulation of real world systems with diverse behavior concepts.

References

1. Kosko B. Fuzzy cognitive maps. Int J Man-Machine Studies No. 24; 1986. pp. 65–75.
2. Koulouriotis DE, Diakoulakis IE, Emiris DM, Antonidakis EN, Kaliakatsos IA. Efficiently modelling and controlling complex dynamic systems using evolutionary fuzzy cognitive maps. Int J Comput Cogn 2003;1(2):41–65.
3. Zhi-Qiang L, Satur R. Contextual fuzzy cognitive map for decision support in geographic information systems. IEEE Trans Fuzzy Syst 1999;7(1):495–507.
4. Carvalho JP, Carola M, Tome JAB. Forest fire modelling using rule-based fuzzy cognitive maps and voronoi based cellulra automata. Fuzzy Information Processing Society NAFIPS 2006. Annual Meeting of the North American; June 2006. pp 217–222.
5. Kosko B. Neural networks and fuzzy systems. USA: Prentice Hall Int Editions;1992, 449 p.
6. Jennings NR, Sycara K, Wooldridge M. A roadmap of agent research and development. Auton Agents Multi-Agent Syst 1998;1:275–306.
7. Petrie C. Agent-based engineering, the web, and intelligence. IEEE Expert 1996;11(6):24–29.

8. Hagiwara M. Extended fuzzy cognitive maps. USA: IEEE Int Conf on Fuzzy Systems; March, 1992. pp 795–801.
9. Miao Y, Liu Z-Q. On causal inference in fuzzy cognitive maps. IEEE Trans Fuzzy Syst 2000;8(1).
10. Taber R. Knowledge processing with fuzzy cognitive maps. Expert Syst Appl 1991;2:83–87.
11. Kim HS, Lee KC. Fuzzy implications of fuzzy cognitive map with emphasis on fuzzy causal relationship and fuzzy partially causal relationship. Fuzzy Sets Syst; 1998;(97):303–313.
12. Tsadiras A. Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps. Inf Sci 2008;178:3880–3894.
13. Stylios CD, Groumpos PP. Fuzzy cognitive map in modeling supervisory control systems. J Intell Fuzzy Syst 2000;8(2):83–98.
14. Stula M. PHD Thesis in Croatian. Croatia: FESB University of Split; 2005. 172 p.
15. Huerga AV. A balanced differential learning algorithm in fuzzy cognitive maps. Barcelona: Sixteenth Int Workshop on Qualitative Reasoning (QR'02); 2002.
16. Park KS, Kim SH. Fuzzy cognitive maps considering time relationships. Int J Human-Comput Stud 1995;42(2):157–168.
17. Carvalho JP, Tome JAB. Rule based fuzzy cognitive maps—qualitative systems dynamics. Atlanta: Proc 19th Int Conf of the North American Fuzzy Information Processing Society NAFIPS2000; 2000.
18. Tsadiras AK, Margaritas KG. Cognitive mapping and certainty neuron fuzzy cognitive maps. Inf Sci 1997;(101):109–130.
19. Wooldridge M, Jennings N. Intelligent agents: Theory and practice. Knowl Eng Rev 1995;10(2).
20. He M, Jennings RN, Leung H-F. On agent-mediated electronic commerce. IEEE Trans Knowl Data Eng 2003;15(4):985–1003.
21. Gruber TR. Toward principles for the design of ontologies used for knowledge sharing. Padova: Int. Workshop on Formal Ontology; 1993.
22. Chandrasekaran B, Johnson TR, Benjamins VR. Ontologies: What are they? why do we need them?. IEEE Intell Syst Appl Spec Issue Ontologies 1999;14:20–22.
23. Bellifemine F, Rimassa G, Poggi A. JADE – A FIPA-compliant agent framework. UK: Proc 4th Int Conf and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents; 1999.
24. Bellifemine F, Caire G, Greenwod D. Developing multi-agent systems with JADE. New York: John Wiley & Sons; 2007. 300 p.
25. Staab S, Schnurr H-P, Studer R, Sure Y. Knowledge processes and ontologies. IEEE Intell Syst 2001;16(1):26–34.
26. Noy NF, Fergerson RW, Musen MA. The knowledge model of Protege-2000: Combining interoperability and flexibility. Juan-Les-Pins: 12th Int Conf in Knowledge Engineering and Knowledge Management (EKAW'00); Lecture Notes in Artificial Intelligence LNAI 1937; Springer-Verlag; Berlin; Germany; 2000. pp 17–32.
27. Gennari JH, Musen MA, Fergerson RW, Grosso WE, Crubézy M, Eriksson H, Noy NF, Tu SW. The evolution of protégé: An environment for knowledge-based systems development. Int J Human-Comput Stud 2003;58(1):89–123.
28. Uchida M, Nakano K. Swing up of an inverted pendulum by simulator-based foresightcontrol. Proc 1999 IEEE Int Conf on Control Applications; Vol 2; 1999. pp 1255–1259.
29. Karam M, Zohdy MA. Modeling a simple inverted pendulum using a model-based dynamic recurrent neural network. Proc 37th Southeastern Symp on System Theory; 2005. pp 78–82.
30. Kandasamy WBV, Smarandache F. Fuzzy cognitive maps and neutrosophic cognitive maps. USA: Books on Demand; ProQuest Information & Learning; 2003. 212 p.
31. Stula M, Stipanicev D, Vuko T. Agent based fuzzy cognitive maps in fire fighting decision support. Split: 14th TIEMS Annual Conf Proc; 2007. pp 529–537.