Kovács, Sz.: *Fuzzy Behaviour-based Control Techniques in Adaptive System Applications*, Proceedings of the ICCC 2003, IEEE International Conference on Computational Cybernetics, August 29-31, Siófok, Hungary, ISBN 963 7154 18 3, p. 6, (2003). – Draft version.

Fuzzy Behaviour-based Control Techniques in Adaptive System Applications

Szilveszter Kovács
Department of Information Technology
University of Miskolc
Miskolc-Egyetemváros, Miskolc, H-3515
Hungary
szkovacs@iit.uni-miskolc.hu

Abstract – The environment adaptation ability of the behaviour-based control structures – the intelligent adaptation of the system to the current situation, by discrete switching to the most appropriate strategy, or by fusing the strategies appeared to be the most appropriate ones – can be easily extended to other adaptive applications by reinterpreting the sense of the "environment".

In case when the system to be controlled acts as the "environment", the structure can form a fault tolerant control. If the actual user acts as the "environment", the structure can form a user adaptive system. For introducing a flexible platform and some application areas, a fuzzy behaviour-based control structure, and highlights of its application in vehicle control, fault tolerant control and user adaptive systems will be briefly discussed in the followings.

I. INTRODUCTION

In behaviour-based control systems (a good overview can be found in [3]), the actual behaviour of the system is formed as one of the existing behaviours (which fits best the actual situation), or a kind of fusion of the known behaviours appeared to be the most appropriate to handle the actual situation. This structure has two main tasks. The first is a decision, which behaviour is needed in an actual situation, and the levels of their necessities in case of behaviour fusion. The second is the way of the behaviour fusion. The first task can be viewed as an actual system state approximation, where the actual system state is the set of the necessities of the known behaviours needed for handling the actual situation. The second is the fusion of the known behaviours based on these necessities.

The applicability of the behaviour-based control structures is based on the premise, that all the situations could possibly occur can be handled by a behaviour formed as a convex combination of the known (existing) behaviours. This case having the relevant behaviours (control strategies in case of control application, or user models in case of user adaptive systems) the behaviour-based control structure has the chance to form a suitable actual behaviour (control strategy or emotional model).

In the followings, first a flexible fuzzy behaviour-based control platform and then some of its possible application areas in vehicle control, fault tolerant control and user adaptive systems will be briefly introduced in this paper.

II. THE APPLIED BEHAVIOUR-BASED STRUCTURE

The first task of the behaviour-based control is to determine the necessities of the known behaviours needed for handling the actual situation. In the behaviour-based control structure applied in the examples of this paper, for this task the finite state fuzzy automaton [4] is adapted (Fig.1.). This solution is based on the heuristic, that the necessities of the known behaviours for handling a given situation can be approximated by their suitability. And the suitability of a given behaviour in an actual situation can be approximated by the similarity of the situation and the prerequisites of the behaviour. (Where the prerequisites of the behaviour is the description of the situations where the behaviour is valid (suitable itself)). This case instead of determining the necessities of the known behaviours, the similarities of the actual situation to the prerequisites of all the known behaviours can be approximated.

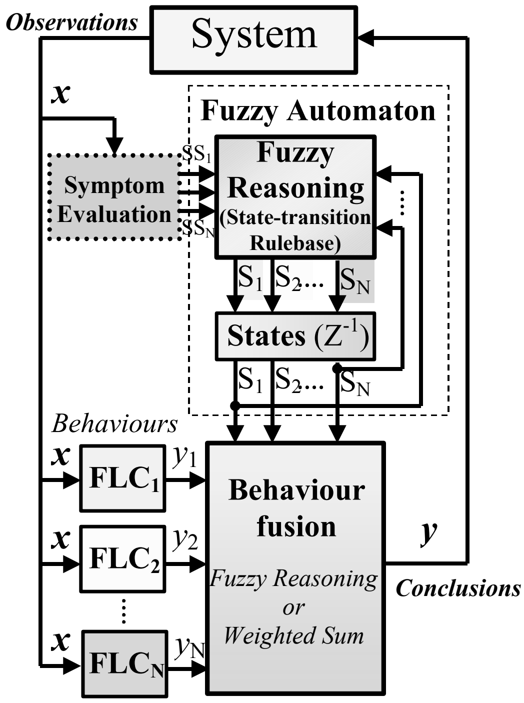

Fig. 1. The applied behaviour-based control structure

Thus the first step of the system state approximation is determining the similarities of the actual situation to the prerequisites of all the known behaviours – applying the terminology of fault classification, it is the symptom evaluation (see e.g. Fig.1.). The task of symptom evaluation is basically a series of similarity checking between an actual symptom (observations of the actual situation) and a series of known symptoms (the prerequisites – symptom patterns – of the known behaviours). These symptom patterns are characterising the systems states where the corresponding behaviours are valid. Based on these patterns, the evaluation of the actual symptom is done by calculating the similarity values of the actual symptom (representing the actual situation) to all the known symptoms patterns (the prerequisites of the known behaviours). There are many methods exist for fuzzy logic symptom evaluation. For example we can adopt fuzzy classification methods e.g. the Fuzzy c-Means fuzzy clustering algorithm [1], where the known symptoms patterns are the cluster centres, and the similarities of the actual symptom to them can be fetched from the fuzzy partition matrix. On the other hand, having a simple situation, the fuzzy logic symptom evaluation could be a fuzzy rule based reasoning system itself. One of the main difficulties of the system state approximation is the fact, that most cases the symptoms of the prerequisites of the known behaviours are strongly dependent on the actual behaviour of the system. Each behaviour has its own symptom structure. In other words for the proper system state approximation, the approximated system state is needed itself. A very simple way of solving this difficulty is the adaptation of fuzzy automaton. This case the state vector of the automaton is the approximated system state, and the state-transitions are driven by fuzzy reasoning (Fuzzy state transition rulebase on Fig.1.), as a decision based on the previous actual state (the previous iteration step of the approximation) and the results of the symptom evaluation. The basic structure of the rulebase applied for the state-transitions of the fuzzy automaton (rules for interpolative fuzzy reasoning) for the ith state S_i (R_{Ai}) can be the following:

If S_i=One **And** S_i-S_i=One **Then** S_i=One **(1)**
If S_i=One **And** S_i-S_k=One **Then** S_i=Zero
If S_k=One **And** S_k-S_i=One **Then** S_i=One
If S_k=One **And** S_k-S_i=Zero **Then** S_i=Zero

where S_i-S_k is the conclusion of the symptom evaluation about the state-transition from state i to k, $\forall k \in [1, N]$, N is the number of known behaviours (state variables). The structure of the state-transition rules is similar for all the state variables. Zero and One are linguistic labels of fuzzy sets (linguistic terms) representing high and low similarity. The interpretations of the Zero and One fuzzy sets can be different in each S_i, S_i-S_k universes. The reason for the interpolative manner of fuzzy reasoning is the incompleteness of state-transition rulebase [2].

In case of having a simple situation, where fuzzy logic rule based symptom evaluation can be applied, the fuzzy symptom evaluation (rulebase) could be integrated to the state transition rulebase of the fuzzy automaton (as it was done in the first example application of this paper).

The conclusion of the system state approximation (the approximated state itself) is a set of similarity values, the level of similarities of the actual situation and the prerequisites of the known behaviours. Applying these similarities as the level of necessities for fusing the known behaviours, the actual behaviour can be formed.

In case of fuzzy behaviour fusion, the following rulebase can be used for the fusion of the conclusions of the different behaviours: **(2)**

If S_1=One **And** S_2=Zero **And...And** S_N=Zero **Then** $\boldsymbol{y}=y_1$
If S_1=Zero **And** S_2=One **And...And** S_N=Zero **Then** $\boldsymbol{y}=y_2$
...
If S_1=Zero **And** S_2=Zero **And...And** S_N=One **Then** $\boldsymbol{y}=y_N$

where S_i is the ith state variable, y_i is the conclusion of the ith behaviour and $\boldsymbol{y}$ is the fused conclusion. Zero and One are linguistic labels of fuzzy sets (linguistic terms) representing high and low similarity. The interpretations of these fuzzy sets can be different in each S_i universes.

Instead of fuzzy reasoning a kind of weighted average, (where the weights are functions of the corresponding similarities) is also applicable (even it is not so flexible in some cases). E.g.:

$$\boldsymbol{y} = \frac{\sum_{i=1}^{N} w_i \cdot y_i}{\sum_{i=1}^{N} w_i}, \qquad (3)$$

where $w_i = S_i$ is the weight of the i^{th} behaviour.

III. APPLICATION EXAMPLES

For introducing some of the possible application areas of the proposed fuzzy behaviour-based control structure, a vehicle control, a fault tolerant control application and a user adaptive emotion-based selection system are shortly introduced in the followings.

A. Vehicle navigation control example

The first application example is a simulated steering control of an automated guided vehicle (AGV) [6], [7]. In the example application the steering control has two main goals, the path tracking (to follow a guide path) and the collision avoidance. The simulated AGV is first trying to follow a guide path, and in the case if it is impossible (because of the obstacles) leave it, and as the collision situation is avoided try to find the guide path and follow it again. A simulated path sensing system senses the position of the guide path by special sensors (guide zone) tuned for the guide path. The goal of the path tracking strategy is to follow the guide path by the guide zone with minimal path tracking error on the whole path (see Fig.2.).

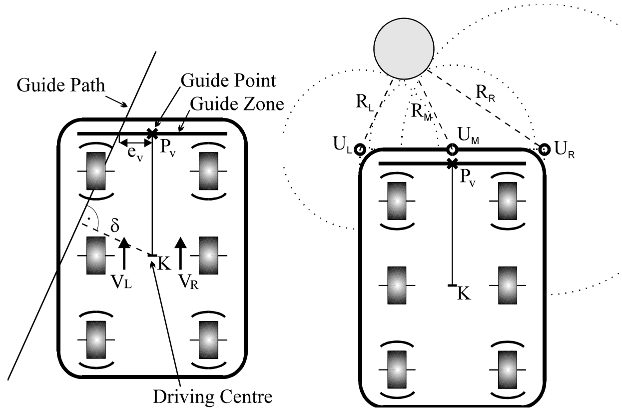

Fig. 2. Differential steered AGV with guide zone, δ is the path tracking error, ev is the distance of the guide path and the guide point, Pv is the guide point, K is the driving centre, RL, RR, RM are the distances measured by the left, right and middle ultrasonic sensors (UL, UR, UM).

In the collision avoidance strategies, two different collision situations, the frontal and the side collision are distinguished. Having the preconditions of motionless obstacles, it is sufficient to have three ultrasonic distance sensors (on the front of the AGV, one in the middle (U_M) and one-one on both sides (U_L, U_R) (see Fig.2.)) to approximate both the collision conditions [7]. Having the preconditions of motionless obstacles, the obstacle distance measurements of the near past can be used for scanning the boundaries of the obstacles. Collecting the previous measurements of the left and right obstacle sensors and the corresponding positions of the AGV (measured by the motion sensors on the wheels), the boundaries of the obstacles can be approximated [7].

The first stage of building the behaviour-based control structure is to build the component behaviours. The simplest way of defining these strategies is based on describing the operator's control actions. These control actions could form a fuzzy rule base. In the example – using interpolative fuzzy reasoning for direct fuzzy control – constructing the fuzzy rule base is very simple. It is not necessary to build a complete fuzzy rule base; it is enough to concentrate on the main control actions, by simply adding rules piece by piece. Having the simulated model of the controlled system, the performance of the controller can be checked after each step. (See more detailed in [7].) In the example, there are four different known behaviours:

Path tracking and restricted collision avoidance strategy: The main goal of this strategy is the path tracking (to follow a guide path) and as a sub goal, a kind of restricted (limited) collision avoidance [7]. (Here the restricted collision avoidance means, "avoiding obstacles without risking the chance of loosing the guide path".) The base idea of the path tracking strategy is very simple: keep the driving centre of the AGV as close as it is possible to the guide path, than if the driving centre is close enough to the guide path, simply turn the AGV into the new direction. Adding the collision avoidance, this simple strategy needs seven observations: Two for the path tracking, the distance between the guide path and the driving centre (e_v), and the distance between the guide path and the guide point (δ). Five for the collision avoidance, the distances measured by the left middle and right ultrasonic sensors (R_L, R_M, R_R) and the approximated maximal left and right turning angle without side collision (α_{ML}, α_{MR}). Based on these observations two conclusions, the speed (V_a) and the steering (V_d) are calculated (see Fig.2. and [7] for more details).

The collision avoidance strategy: The second known behaviour is a simple collision avoidance steering strategy. Its only goal is to avoid collisions.

The collision avoidance with left/right tendency strategy: The next two behaviours are basically the same as the collision avoidance steering strategy, expect the left or right turning tendencies in case of no left or right turning difficulties. These strategies are needed to aid finding the path after leaving it (because of the fail of the first strategy). Their rulebases are the same as the rulebases of the collision avoidance strategies, except one additional rule, which causes the left/right turning tendencies in collision free situations.

The example application is so simple, that it does not need separate symptom evaluation. The function of the symptom evaluation is built to the state-transition rulebase of the fuzzy automaton. Having four known behaviours, the automaton has four state variables. These are the approximated level of similarity of the actual system to the prerequisites of the path tracking and restricted collision avoidance strategy (S_P), to the prerequisites of the collision avoidance strategy (S_C), to the prerequisites of the collision avoidance strategy with right tendency (S_{CR}), and left tendency (S_{CL}). Having four conclusions, four state transition rulebases is needed. The $\mathbf{R_{SP}}$ state transition rulebase is determining the next value of the S_P state variable, $\mathbf{R_{SC}}$ is for determining S_C, $\mathbf{R_{SCR}}$ for S_{CR}, and $\mathbf{R_{SCL}}$ for S_{CL}. The observations of the state transition rulebases are the observations introduced in the path tracking and partial collision avoidance strategy, the state variables themselves (S_P,S_C,S_{CR},S_{CL}), and a new observation (P_V), signing if the path sensing is available. The state-transition rulebases for interpolative fuzzy reasoning are the following:

$\mathbf{R_{SP}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_P
				Z	*V*			*L*			*L*
				PL	*V*					*S*	*Z*
				NL	*V*				*S*		*Z*
					NV						*Z*

$\mathbf{R_{SC}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_P
					V			*S*			*L*
					V			*L*			*Z*
					NV						*Z*

$\mathbf{R_{SCR}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_P
L				*NVL*	*V*						*L*
		L			*NV*						*L*
				Z	*V*			*L*			*Z*
			L								*Z*

$\mathbf{R_{SCL}}$:

S_P	S_C	S_{CR}	S_{CL}	e_v	PV	R_L	R_R	R_M	α_{ML}	α_{MR}	S_P
L				*PVL*	*V*						*L*
			L		*NV*						*L*
				Z	*V*			*L*			*Z*
		L									*Z*

where *N*: negative, *P*: positive, *VL*: very large, *L*: large, *S*: small, *Z*: zero, *V*: path valid, *NV*: path not valid.

Conclusions of the example: Fig.3. and Fig.4. introduces some results of the simulated application. The results shows, that in the tested situation the suggested fuzzy behaviour-based control structure was able fuse the known behaviours in the expected manner.

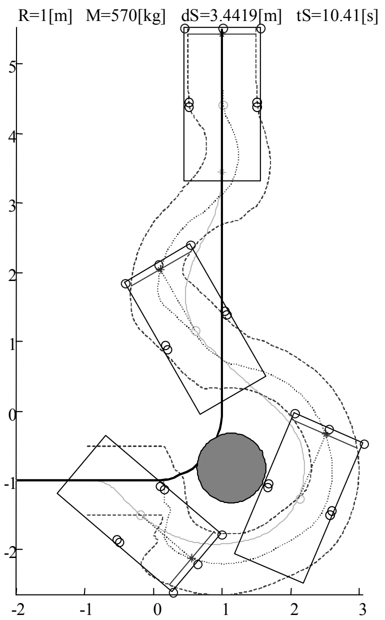

Fig. 3. Track of a single run in case of one obstacle.

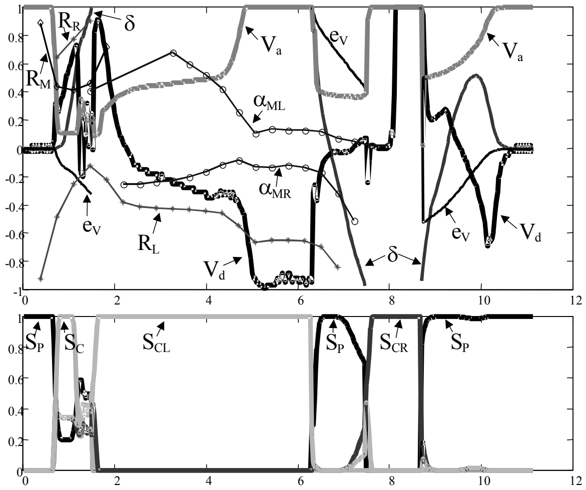

Fig. 4. Time function of observations, conclusions and system state values (S_P, S_C, S_{CL}, S_{CR}) related to the track of Fig.3.

B. Fault tolerant control example

As a second simple application example of the suggested fuzzy behaviour-based control structure, fault diagnosis and reconfiguration of a simplified configuration (two tanks only) of the three tank benchmark [5] (Fig.5.) is introduced in the followings.

The goal of the control system is to keep the water levels in $tank_1$ and $tank_3$ to be h_1=0.5 and h_3=0.1 by controlling the $valve_{13}$ and the $pump_1$ at a constant value of outflow from $tank_3$ (*normal* behaviour of the system).

The example is concentrating of the faults of the $valve_{13}$. Were this valve opened and blocked, the water level in $tank_3$ h_3=0.1 could be controlled by $pump_1$ (this case h_1 is changed) – this is the *fault condition no.1*. Were this valve closed and blocked, the water levels in $tank_1$ and $tank_3$ h_1=0.5 and h_3=0.1 could be controlled by the $valve_1$ and the $pump_1$ – this is the *fault condition no.2*.

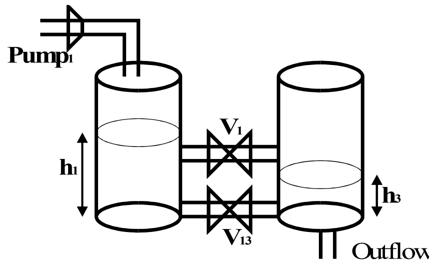

Fig. 5. Simplified configuration of the three tank benchmark.

The way of the implementation of the suggested fuzzy behaviour-based control structure is very similar to the previous example. However having a little bit more complicated symptom structure, after building the component behaviours – the controllers handling the separate situations (one controller for the normal, one for handling fault 1 and one for fault 2) this case, a separate symptom evaluation module is built. In this example for this task the fuzzy clustering was applied. The symptom evaluation module has to be able to characterise all the state-transitions in all the studied behaviours. The state-transition diagram for the studied states of the interpolative fuzzy automaton is shown on Fig.6. (The states are fuzzy membership values.) The structures of the state-transition rulebase and the behaviour fusion rulebase (2) are similar as introduced in section two in (1) and (2).

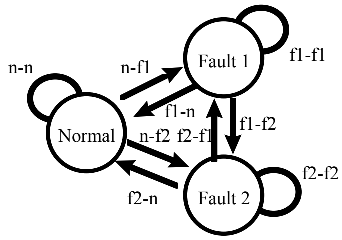

Fig. 6. State-transition diagram of the fuzzy automaton, where e.g. f1-f2 is the conclusion of the symptom evaluation related to the - from f1 to f2 state-transition.

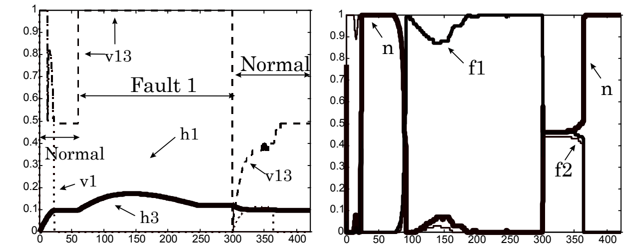

Fig. 7. The simulated results and the approximated fuzzy states of the complete control system (Normal - Fault 1 – Normal behaviour).

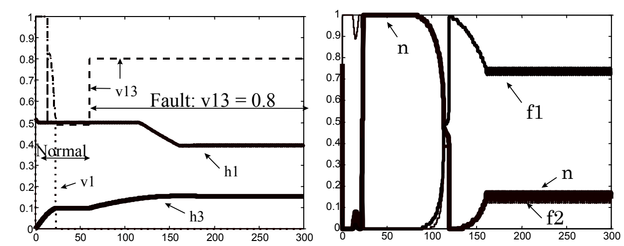

Fig. 8. The simulated results and the approximated fuzzy states of the complete control system (Normal - Fault: v13 = 0.8 unstudied situation).

Conclusions of the example: The simulated example application demonstrated, that the fuzzy automaton is able to follow the studied relevant states and state-transitions (e.g. normal-fault1-normal on Fig.7.). Moreover, because of the fuzzy state approximation and the fuzzy behaviour fusion in some cases the system is able to handle unknown (unstudied) fault situations too (see e.g. on Fig.8., where v_{13}=0.8 open is an unstudied fault situation).

C. User adaptive emotion-based system example

The last simple application example of the suggested fuzzy behaviour-based control structure is a user adaptive emotion-based system – an interactive selection system [8]. The example application is based on the idea, that from the viewpoint of the application, the user adaptivity is similar task as the situation adaptivity, introduced in the previous examples. Forming the emotional user model as on-line variable fusion of some fixed existing (off-line collected) models. In this case the user adaptation itself is handled as a kind of adaptive fusion of existing emotional models in the manner of "the more similar the actual user to one of the existing emotional model, the more similar must be the actual emotional model to that model". In other words, instead of identifying the actual emotional model itself, the user is classified in the manner of existing emotional models. As an analogy to the previous applications, the different known behaviours are the different known emotional models, and the actual situation is the similarity of the actual user to the evaluators, gave the known emotional models. The main benefit of this view is quick convergence, as in the most cases the problem of user classification related to some existing emotional models is much simpler than the identification of the complicated emotional model itself. The ability of proper depiction of user emotion is highly dependent on the number and diversity of the known models available in the system. The implementation of the suggested fuzzy behaviour-based control structure is very similar to the previous examples. The main differences (Fig.1, Fig.9.) are the substitution of the known behaviour controllers (FLC_i) by the known emotional models (Object Descriptor – Emotional Descriptors), and the direct similarity checking (similarities of the actual user opinions to the content of the existing models) instead of symptom evaluation.

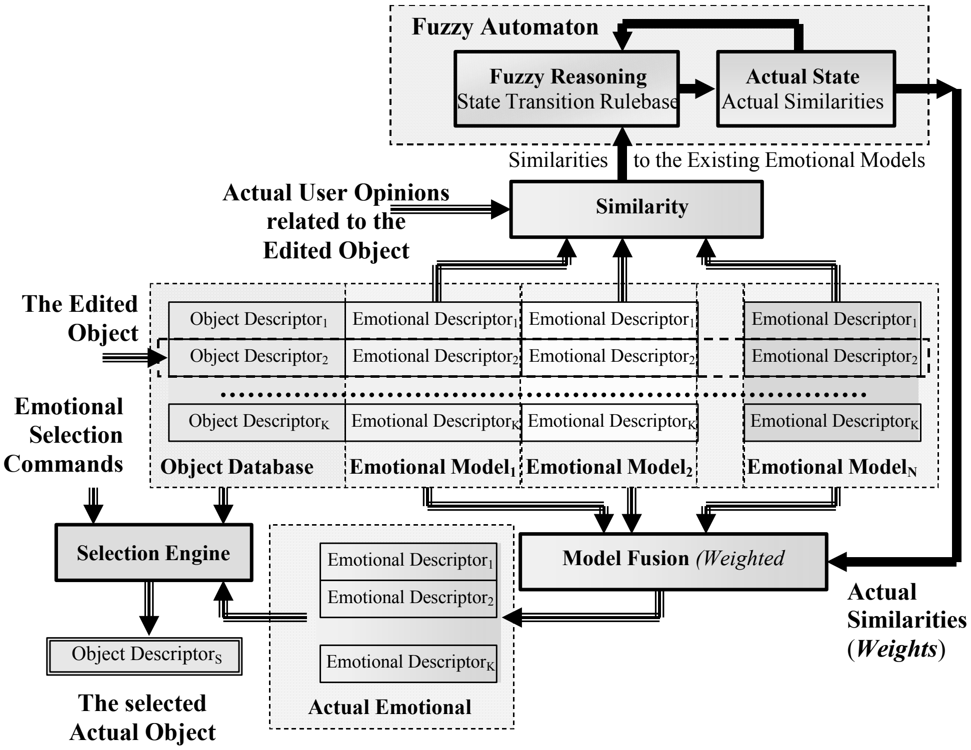

Fig. 9. Structure of the proposed adaptive emotional model generation.

Using the selection system, the user can search the object database by giving emotion-related requests (like "friendly" or "convenient"). These requests are translated to physical parameters (characterising the real objects) by the actual emotional model. The user adaptivity of the actual emotional model (see Fig.9.) (since the physical meanings of the emotional words are highly user dependent) is provided by the suggested fuzzy behaviour-based control structure. This case the state of the fuzzy automaton (actual similarities, see Fig.9.) is interpreted as the actual approximated similarities of the actual user opinions and the known emotional models. In practice the automaton is starting from an initial state (e.g. all the similarities are equal to 0.5), and during the events of the user feedback (e.g. giving his/her opinions related to an "edited object" – see Fig.9.) the actual similarities are recalculated. The state-transitions rulebase is a slightly different than the rulebase (1) suggested in the first section. This is because the direct similarity checking gives the similarities of the actual user opinions to the content of the known models, instead of the state-transition decisions of the symptom evaluation (the similarities are independent from the actual model, or state). The rulebase applied for the state-transitions of the fuzzy automaton for the i[th] state variable S_i, $i \in [1,N]$ of the state vector **S** is the following (rules for interpolative fuzzy reasoning) [8]:

If S_i=One **And** SS_i=One **Then** S_i=One

If S_i=Zero **And** SS_i=Zero **Then** S_i=Zero

$$\text{If } S_i=\text{One} \text{ And } SS_i=\text{Zero} \text{ And } SS_k=\text{Zero} \text{ Then } S_i=\text{One} \quad \forall k \in [1,N], k \neq i$$

$$\text{If } S_i=\text{Zero} \text{ And } SS_i=\text{One} \text{ And } S_k=\text{One} \text{ And } SS_k=\text{One} \text{ Then } S_i=\text{Zero} \quad \forall k \in [1,N], k \neq i$$

$$\text{If } S_i=\text{Zero} \text{ And } SS_i=\text{One} \text{ And } S_k=\text{Zero} \text{ And } SS_k=\text{Zero} \text{ Then } S_i=\text{One} \quad \exists k \in [1,N], k \neq i$$

where SS_i is the calculated similarity of the actual user opinion to the i^{th} known emotional model, N is the number of models (or state variables). The structure of the state-transition rules is similar for all the state variables. Zero and One are linguistic labels of fuzzy sets (linguistic terms) representing high and low similarity. The interpretations of the Zero and One fuzzy sets can be different in each S_i, SS_i universes. The reason for the interpolative manner of fuzzy reasoning is the incompleteness of state-transition rulebase [2].

In the example application, because of the simplicity of the model fusion (there is no need for the flexibility of the rule-based fusion), and the need of the quick response of the interactive program, weighted average (3) were used.

Conclusions of the example: The goal of the actual user model modifications from the actual user side is to tune the system to be closer to his/her opinions. Practically the system is starting from an initial stage (where the similarities to the existing models are equal), and in the case if the user is disagreeing with the evaluation of the actual object given by the system, he/she has the possibility to modify the actual user model by giving his/her opinions. In most cases the given opinions are related to one or a few emotional descriptors of the edited object (see Fig.9.). But because of the suggested structure, all the changes are done globally (all the emotional descriptors of known model has the same weights "globally" in the actual model – not only the descriptor weights related directly to the given user opinion are "locally" modified). Hopefully that this kind of adaptation strategy keeps the actual user model coherent, and able to avoid incoherence could caused by step by step partial modifications. E.g. if one of the users have exactly the same opinions as one of the known model (even his/her opinions were given through some of the emotional parameters only), then (after a few modification, detection steps) as the best fitting known model, the system will use it exactly. Basically the "adaptive knowledge" of the system related to the actual user is not a new adapted emotional model, but the actual system state, a set of approximated similarities of the actual user opinions to the known emotional models. Because of the convex combination way of the emotional model combination, the suggested structure is unable to follow user requirements outside the area covered by the known models. In other words, the system cannot go beyond its existing "knowledge". The only solution of this problem is extending the number and the variety of the known emotional models, to cover the model space as much as it is possible.

IV. CONCLUSIONS

The goal of this paper was to introduce a simple and flexible fuzzy behaviour-based control structure and some of its possible application areas, a behaviour-based vehicle control, a fault tolerant control and a user adaptive emotion-based selection system application. The main benefits, both the simplicity and the situation adaptivity of the behaviour-based control structures are inherited from their hierarchical construction. In case of the first and second application example, this hierarchy has the meaning of building a (more) global strategy from some relevant, but only partially valid (with respect to the state space of the system) strategies. The suggested fuzzy behaviour-based control structure is simply fusing these strategies to form one strategy, which has an extended area of validity. This way a rather complicated strategy can be modularly built. Moreover, because of the fuzzy state approximation and the fuzzy behaviour fusion in some cases the system is able to handle unknown (unstudied) situations too (like in the second example (see Fig.8.), where "the tap is fixed 0.8 open" is an unstudied fault situation). The benefit of adapting fuzzy automaton for system state (similarity) approximation in the proposed structure is to give (state) memory to the system. On one hand this memory is needed for the correct symptom evaluation (see the second example), or able to hold a kind of "history" information as it was introduced in the first example (left, or right turning tendency strategy decision). On the other hand, in case of adaptive applications, the system state can be viewed as the model of the actual situation, or the surrounding environment of the system from the viewpoint of the known behaviours. Like in the case of the third example, where the system state is the model of the actual user with respect to the known emotional models.

V. ACKNOWLEDGMENT

This research was partly supported by the Japan Gifu Prefectural Research Institute of Manufacturing Information Technology and the Intelligent Integrated Systems Japanese Hungarian Laboratory. Szilveszter Kovács is supported by the György Békésy Postdoctoral Scholarship.

VI. REFERENCES

[1] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function*, Plenum Press, New York, (1981).

[2] L.T. Kóczy and K. Hirota, "Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases", *Information Sciences*, **71**, (1992), pp. 169-201.

[3] P. Pirjanian, "Behavior Coordination Mechanisms - State-of-the-art", *Tech-report IRIS-99-375*, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, http://www-robotics.usc.edu/~paolo/publications/bcm.ps.gz, October, (1999).

[4] Sz. Kovács, "Interpolative Fuzzy Reasoning and Fuzzy Automata in Adaptive System Applications", *Proceedings of the IIZUKA2000*, 6th International Conference on Soft Computing, October 1-4, Iizuka, Fukuoka, Japan, (2000), pp.777-784.

[5] Sz. Kovács, "Similarity Based System Reconfiguration

by Fuzzy Classification and Hierarchical Interpolate Fuzzy Reasoning", *Lecture Notes in Computer Science*, **1625**, Springer, (1999), pp.12-19.

[6] Sz. Kovács and L.T. Kóczy, "Application of the Approximate Fuzzy Reasoning Based on Interpolation in the Vague Environment of the Fuzzy Rulebase in the Fuzzy Logic Controlled Path Tracking Strategy of Differential Steered AGVs", *Lecture Notes in Computer Science*, **1226**, Springer, Germany, (1997), pp.456-467.

[7] Sz. Kovács and L.T. Kóczy, "Path Tracking and Collision Avoidance Strategy of an AGV Implemented on Interpolation-based Fuzzy Logic Controller", *Proceedings of the INES'98 IEEE International Conference on Intelligent Engineering Systems*, Vienna, Austria, (1998), pp.67-72.

[8] Sz. Kovács, N. Kubota, K. Fujii and L.T. Kóczy, "Behaviour based techniques in user adaptive Kansei technology", *Proceedings of the VSMM2000, 6th International Conference on Virtual Systems and Multimedia*, October 3-6, Ogaki, Gifu, Japan, (2000), pp.362-369.