Extended FRI Methods and it's applications

Zoltán Krizsán

MSc in Information Engineering

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Faculty of Mechanical Engineering and Informatics

University of Miskolc

Supervisor: PhD Szilveszter Kovács

January 2014

CONTENTS

*

0.1 TÉMAVEZETŐ AJÁNLÁSA

Krizsán Zoltánt 1997-ben ismertem meg, 2000-ban mérnök informatikus oklevelet szerzett kiváló minősítéssel. Doktori tanulmányait 2011/2012 I. és II. félévben végezte. Doktori szigorlatát 2013 május 30.án summa cum laude minősítéssel. Tanszékünkön 2003-tól dolgozik, főleg programozási nyelveket tanít (C++, Java, C#, web alkalmazás fejlesztés, ...). 2010-ben csatlakozott az általam vezetett miskolci egyetemen belül működő Fuzzy rendszerek kutató csoporthoz. Számos közös cikkünk bizonyítja a kutatói munkában való jártasságát. 2009-2011 időszakban a SZTAKI (Számítástechnikai és Automatizálási Kutatóintézet) kognitív informatika (jelenleg: 3D Internet-alapú Kontroll és Kommunikációs Kutatólaboratórium) laborjában, mint külső munkatárs dolgozott. Itt dolgozta ki, majd fejlesztette le a VirCa (VIRtual Collaboration Arena) alapjait, amely egy OpenRTM-aist robot keretrendszer kiterjesztése. Munkájának is köszönhető, hogy a projekt sikeresen zárult 2011-ben. A témából egy könyv fejezet, és 7 konferencia cikk készült. Web információs rendszerek üzemeltetésében már régóta részt vesz. A Gépészmérnöki és Informatikai kar információs oldalát fejlesztette, szerkesztette és üzemeltette 2002-2009 időszakban. A tanszéki információs oldalunkat a 2005-től folyamatosan a mai napig üzemelteti. Hallgatókból álló kis csapatot vezet a mai napig, amely kifejlesztette az FRI Toolbox-ot. Ez a programozói könyvtár implementálja a ma fellelhető Fuzzy interpolációs módszereket, lehetőséget biztosítva a kiterjesztésre. A témához kapcsolódóan 3 TDK dolgozat is született, a konzultálásával. A Magyar Fuzzy Társaság tagja jelenleg is, valamint a társaság információs oldalának üzemeltetője. A társaság által rendezett konferenciákon folyamatosan opponensi, szervezői feladatokat lát el. Jelenleg is munka kapcsolatban áll a BME Mechatronika, Optika és Gépészeti Informatika (MOGI) tanszékével valamint a Kassai Egyetem intelligens technológiák központjával (Center for Intelligent Technologies) robot rendszerek témakörben. Mindezt összegezve Krizsán Zoltánt egy csapat munkába dolgozni képes, kreatív embernek ismertem meg, aki elméleti tudását a gyakorlatban is megvalósítja.

0.2 DECLARATION

The author hereby declares that this thesis has not been submitted, either in the same or different form, to this or any other university for a Ph.D. degree. The author confirms that the work submitted is her own and the appropriate credit has been given where reference has been made to the work of others.

NYILATKOZAT

Alulírott Krizsán Zoltán kijelentem, hogy ezt a doktori értekezést magam készítettem, és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Monday 7^{th} January, 2014

Zoltán Krizsán

A disszertáció bírálatai és a védésről készült jegyzőkönyv megtekinthető a Miskolci Egyetem Gépészmérnöki és Informatikai Karának Dékáni hivatalában, valamint a doktori iskola weboldalán az Ertekezések menüpont alatt:

http://www.hjphd.iit.uni-miskolc.hu.

"The clock is running. Make the most of today. Time waits for no man.
Yesterday is history. Tomorrow is a mystery. Today is a gift. That's why it is called the present."

(Alice Morse Earle)

0.3 ACKNOWLEDGEMENTS

This dissertation contains the work of many years that would not have been realized without the support of others.

First and foremost, I would like to thank my grandmother who brought me up and gave up her own dream because of mine.

I am most thankful to my supervisor, Dr. Szilveszter Kovács who never stop believing in me for his guidance.

I would like to thank all members of Department of Information Technology who helped me for 15 years, and Ádám Csapó for reviewing and correcting my English.

I dedicate this work to my sons Levi and Soma, wishing them a healthy and true life.

1

DEFINITIONS AND NOTATIONS

This section introduces elementary definitions and concepts utilized in later parts of the dissertation. Scalar values are denoted by lowercase letters such as a, b, c, *ratio*, ...; fuzzy sets by capital letters such as A, B, P, Q, ...; and the letter R is reserved to denote fuzzy rules of the form IF $x = A$, THEN $y = B$, or $R : A \longrightarrow B$ in short notation. The letters X, Y, and S are reserved to denote input–output universes and the third dimension of the geometrical representation introduced in the dissertation (see later), respectively. x denotes an observation, and y denotes a conclusion. Antecedent fuzzy sets are denoted by $A(A \in L^X)$; consequent sets by $B(B \in L^Y)$, where L^X and L^Y are fuzzy spaces on X and Y, respectively.

There are several common concepts which FRI methods rely on when calculating the similarity of fuzzy sets. Many FRI methods determine the similarity based on α-cuts in terms of distances in every important α value ($\alpha = [0..1]$). Because this approach presupposes the availability of α-cuts, its applications are restricted to CNF sets (e.g. LESFRI [1]). Another possible fuzzy set similarity calculation is based on the polar coordinate system and polar cuts. This method is also suitable for calculating the similarity of subnormal fuzzy sets. Until now, only a single FRI method used this technique. This is the FRIPOC method, which was introduced by Johanyák in [2]. The FRIPOC method calculates the consequent for every θ angle ($0 \leq \theta \leq 180$) in the domain, such that the reference point of the fuzzy set (e.g. the centre of the core, or the centre of gravity) is also the reference point of the polar coordinate system. Other methods also exist for calculating the similarity of fuzzy sets, such as the "scale and move transformation" [3], in which similarity is approximated by the values of scaling and translation necessary to deform two fuzzy sets with different shapes to fit each other.

It can be stated that at the moment there is no common, universally accepted method for representing the similarity between pairs of arbitrarily-shaped fuzzy sets.

An overview of common guidelines followed by FRI methods, referred to as the "axiomatic approach of the fuzzy interpolation" can be found in [4], [5] and [6]. Two of the guidelines, referred to as the "avoidance of invalid conclusions" and the "preservation of linearity" will be studied more extensively in this dissertation. The "avoidance of invalid conclusions" (or "validity of the mapping" in [6]) criterion means that the FRI method has to generate a valid fuzzy set for its conclusion. The "preservation of linearity" (or "shape invariance of the mapping" in [6]) criterion means that in the case of observation of piecewise linear shaped linguistic terms, the FRI method has to preserve their piecewise linear shapes when generating its conclusion.

1.1 DEFINITION 1 (FUZZINESS)

Several definitions of the concept od fuzziness can be found in the literature. A relatively simple one is introduced by Kóczy, Hirota and Gedeon in [7] in the following form:

$$f_{A,L} = inf\{[A]_1\} - inf\{[A]_{0+}\}, \tag{1}$$

$$f_{A,U} = sup\{[A]_{0+}\} - sup\{[A]_1\}, \tag{2}$$

where $f_{A,L}$ and $f_{A,U}$ are the "lower" and "upper" fuzziness, $[inf\{[A]_1\}, sup\{[A]_1\}]$ is the core, and $[inf\{[A]_{0+}\}, sup\{[A]_{0+}\}]$ is the support of fuzzy set A. In later parts of this dissertation, the concept of fuzziness will be understood based on this definition.

1.2 DEFINITION 2 (DOUBLE FUZZY POINT RULE)

A double fuzzy point rule is a pair of two overlapping fuzzy rules (P,Q) that share the same reference points [8].

NOMENCLATURE

p,q Fuzzy rule pair, page 15

CBIDE Comparison Based Interactive Differential Evaluation, page 33

CORBA Common Object Request Broker Architecture, page 48

EC Evolutionary Computation, page 32

GDFPM Generalized Double Fuzzy Point Methodology, page 16

ICE Internet Communication Engine, page 48

IEC Interactive Evolutionary Computation, page 32

OpenRDK Open Robot Developer Kit, page 53

OpenRTM-aist Open Robot Middleware, page 55

RT-middleware Robot Technology Middleware, page 48

RTC Robot Technology Component, page 48

VirCa Virtual Collaborating Arena, page 34

YARP Yet Another Robot Platform, page 53

LIST OF FIGURES

LIST OF TABLES

2

INTRODUCTION

The theses of this dissertation are gathered around three fields of intelligent computation methods. The first two theses are centered on fuzzy interpolation models for sparse rule bases. In particular, the first thesis connects various Fuzzy Rule Interpolation (FRI) methods into a common framework. Based on this framework, it defines a set of common guidelines for a novel FRI methodology which can handle the fuzziness of the fuzzy represenation used (i.e., a kind of meta-fuzziness). The second thesis is related to the application of Interactive Evolutionary Computation (IEC) based on FRI methods. Finally, the third thesis focuses on the relevance of these topics to robot middleware architectures.

The development of any intelligent system rests on three pillars. The first pillar is the model to be used as an engine for the system. This core component of the artificial intelligent system can be viewed as a controller that obtains input from the environment and responds to it based on its internal knowledge. There are many kinds of decision architectures applied in such models, such as fuzzy logic and neural networks.

The second pillar of intelligent system design is the applied training methodology. The previously described core model can be used for different purposes depending on the methodology used to improve and fine-tune its performance. In this dissertation, the FIVE FRI method will be used as a pre-evaluator to assist the IEC system.

The final, third pillar of intelligent system design can be seen as the glue which connects the first two pillars. Several instances of different models and methodologies often work together to form a complete system. To support this co-dependence, communication middleware systems are often used. Such systems, for example in robot technologies, allow for the simplified creation and joint operations of varous robot parts. Several robot middleware systems with unique capabilities have appeared in the past decade.

2.1 AIMS AND SCOPE

In this dissertation, all three areas of intelligent system development described above are detailed and a set of new approaches for each of the new areas are proposed.

My first goal was to provide a general description of double fuzzy point rules and a common set of guidelines for extending them via fuzzy rule interpolation. In his papers, Sz. Kovács discussed several extended FRI methods in some detail, but a set of common guidelines is still missing which would define the steps of the extension process in precise terms. Related to this task, a definition of the preconditions and limitations is also necessary. The key question, then, is which FRI methods can be extended in theory, and under what conditions.

My second goal was to extend the Interactive Evolutionary Computation (IEC) concept in order to decrease the associated human fatigue. To this end, I introduce a pre-evaluation process which provides an evaluation for each individual depending on current user. If the competent user agrees with the result of pre-evaluation, the individual can be left untouched. More than one example (e.g., 5-10) is shown to the user in parallel, and the user only has to change those evaluations which are off the mark. The goodness value of each individual is computed by the pre-evaluator according to previous evaluations of actual experts. My intention was to define the structure and workflow of this system based on FRI methods with sparse rule bases.

My third goal was to define the general requirements of state-of-the-art robot systems, then to explore and implement some of the features which are currently missing. Interoperability in most robot systems today is based on the CORBA middleware, thus, almost every part of today's robot systems depend on the GIOP protocol. My intention was to restructure a state-of-the-art robot middleware so that its System Editor component would be independent from CORBA, allowing for the the creation of a layered structure which is easy to extend.

2.2 DISSERTATION GUIDE

This dissertation is divided into four main sections. Three of the sections describe various aspects of intelligent system development as outlined earlier, while the final section summarizes the results. The first section that immediately follows this introduction presents a newly proposed double fuzzy point rule extension approach in the case of two-step FRI methods, and defines a set of appropriate guidelines for it. Based on the method, two general extensions of FRI methods are detailed, which can be applied irrespective of the original FRI method used.

A novel pre-evaluator approach for Interactive Evolutionary Computing is proposed in the second section, and the existing FIVE FRI model is used to provide an implementation of it. In the third section, the OpenRTM-aist robot middleware is discussed in detail and two novel improvements are proposed. Finally, a summary of my results are given in the last section.

3

GENERAL METHODOLOGY FOR DOUBLE FUZZY POINT DESCRIPTION

3.1 ABSTRACT

The "Double Fuzzy Point" rule representation opens a new dimension for expressing changes of fuzziness in fuzzy rule-based systems. In the case of standard "Fuzzy Point" rule representations, it is difficult to describe fuzzy functions in which crisp observations are required to have fuzzy conclusions, or in which an increase in the fuzziness of observations leads to reduced fuzziness in conclusions. These problems are mainly due to a lack of information. A fuzzy point rule determines the connection between a pair of fuzzy sets taken from the domain and the range of the rule. Expressing the fuzzy function through a set of fuzzy points and fuzzy interpolation between pairs of those points, each fuzzy point can be considered as a node point with given location and fuzziness. In common, sparse rule-base definitions, these node points are usually disjunctive on the domain, defining only single antecedent-consequent fuzziness connections at the location of the fuzzy points. However, this kind of information is insufficient when the goal is to express changes in the fuzziness of a given location in the domain. One solution to this problem is the double fuzzy point rule representation concept. Double fuzzy points are pairs of fuzzy points which share the same reference locations, but have different fuzziness properties. The existence of two different fuzziness values in a single location within the domain creates new possibilities for introducing fuzzy interpolation methods capable of interpolating not only between locations, but between changes in local fuzziness values as well. The main goal of this chapter is to

discuss how two-step Fuzzy Rule Interpolation methods can be adapted to be able to handle the double fuzzy point concept. To this end, an approach referred to as the Generalized Double Fuzzy Point Methodology (GDFPM) is proposed.

3.2 INTRODUCTION

There are numerous Fuzzy Rule Interpolation (FRI) methods which have appeared in the literature. One of the first methods was published by Kóczy and Hirota (KH method [9, 10]). The KH method can handle convex and normal fuzzy (CNF) sets in single dimensional antecedent universes only, determining the conclusion from the α-cuts of the two rules which immediately surround the observation. The KH method inspired many subsequent approaches, such as the modified α-cut based interpolation (MACI) method [11, 12, 13]. MACI transforms fuzzy sets into vector representation, computes the conclusion based on those representations, and finally transforms the conclusion back to the original space. The first FRI method capable of explicitly dealing with "fuzziness" appeared in the "conservation of relative fuzziness" (CRF) method, which was proposed by Kóczy et al. in [7] for single antecedent dimensions. The CRF uses two closet surrounding rules to the observation. It stipulates that the rate of the left (right) fuzziness of the conclusion and the fuzziness of the rule consequents should be the same as the rate of the right (left) fuzziness of the observation and the fuzziness of the two surrounding rule antecedents. A multidimensional extension of the CRF method, known as IMUL, was proposed in [14] "An improved fuzzy interpolation technique for multidimensional input spaces" (IMUL) is a combination of CRF and the multidimensional MACI methods.

In parallel with these developments, a rather different two-step method was proposed by Bouchon-Meunier et al. [15], [16]. At the first step their "analogy-based interpolation" calculates the reference point of the conclusion. This step is simple interpolation based on the reference point distances of the observation and the rule antecedents. In the second step the FRI method constructs the shape of the conclusion according its similarity (distinguishability) to the rule consequents to be the same as the similarity (distinguishability) of the corresponding rule antecedents and the observation. Another two-step method concept is presented in the "General Methodology" (GM suggested by Baranyi et al. in [17]). GM extends the first step of the original analogy-based interpolation to the generation of interpolated "intermediate rules" in the reference point position of the observation. In the second step, a single rule reasoning method (revision function) is applied to determine the final fuzzy conclusion based on the similarity of the fuzzy observation and an "interpolated" observation. In this way, GM can handle arbitrarily shaped fuzzy sets. An extension of GM appeared in the work of Shen et al. [3, 18], Baranyi [19, 20].

Practical applications of GM appear in the "Least Squares Method" ("LESFRI"), in the "FRI based on Subsethood Values" ("FRISUV") as well as in the "Polar a Cut" interpolation ("FRIPOC") suggested by Johanyák et al. in [1], [21], [22] and [2]. As a single rule reasoning step FRIPOC calculates the similarity of fuzzy sets based on their polar cuts. In the remainder of the chapter, the above "two-step" methods will be studied in detail, with the goal of extending them to be able to adopt the "double fuzzy" point rule representation concept. An improvement, referred to as the Generalized Double Fuzzy Point Methodology (GDFPM), is proposed for the case of SISO Mamdani fuzzy systems (i.e., Mamdani systems with one input and one output dimension).

Many of the above mentioned FRI methods and a sparse fuzzy model identification tool are available as open source code MATLAB Toolbox (Johanyák et al [23], [24] and [25]. Systematic model-based fuzzy control approaches are presented in [26].

3.3 THE DOUBLE FUZZY POINT RULE REPRESENTATION

A number of Fuzzy Rule Interpolation (FRI) methods exist which use a variety of different computational concepts, but most of them handle changes in fuzziness in similar ways. This is because of the "Monotonicity" condition, which was first defined in [6] (condition "I2") for the single dimensional antecedent case, and was extended in [6] ("Property 6.") to multidimensional antecedents in the following manner: If the $f_{A^*I} < f_{A^*II}$ in all dimensions (A^*I is more specific than A^*II) then $f_{B^*I} < f_{B^*II}$ holds as well.

According to the condition, it is not possible to reverse changes of fuzziness in the conclusion. Moreover, a singleton conclusion $f_{B^*I} = 0$ can be gained only if the observation is a singleton as well (i.e., $f_{A^*I} = 0$).

The "double fuzzy point" rule representation was proposed in [8] in order to extend the classical fuzzy point concept so that changes of fuzziness in fuzzy rules could also be expressed. The "double fuzzy point" rule is an extension of the single fuzzy point rule representation to two overlapping fuzzy rules (cf. Definition 1) The fuzzy rule pairs share the same reference points, i.e. in both rules, the corresponding antecedent and consequent fuzzy sets have the same reference points, but apart from this condition, they can have different fuzziness properties. Differences in antecedent fuzziness define the domain, and differences in consequent fuzziness define the range of the fuzziness interpolation [8].

The i^{th} rule of the double fuzzy point rule representation has the following form:

$$A_i^{p,q} \longrightarrow B_i^{p,q}, \tag{3}$$

which combines two overlapping fuzzy rules: $A_i^p \longrightarrow B_i^p$ and $A_i^q \longrightarrow B_i^q$, where in both the overlapping rules the antecedent and the consequent fuzzy sets have the same reference point. Depending on the reference point definition e.g. in triangular linguistic terms it could be the core. This case $core(A_i^p) = core(A_i^q), core(B_i^p) = core(B_i^q)$ (see e.g. on Fig. 1 and on Fig. 2). Therefore the double fuzzy point rule-base $R^{p,q}$ can be considered as two overlapping rule-base R^p and R^q too. According to the double fuzzy point rule representation concept [8], the fuzziness interpolation requires an observation A^* within the fuzziness domain of the double fuzzy point rule:

$$f_{A^*} \in [f_{A^q}, f_{A^p}] \tag{4}$$

and generates conclusion B^* within the fuzziness range of the double fuzzy point rule:

$$f_{B^*} \in [f_{B^q}, f_{B^p}] \tag{5}$$

Depending on the p, q part of the double fuzzy point rule, the direction of the fuzziness change can remaining the same, or reverses. The direction of the fuzziness change remains the same (see e.g. on Fig. 1) if:

$$f_{A^q} \leq f_{A^*} \leq f_{A^p} and f_{B^q} \leq f_{B^*} \leq f_{B^p}, \tag{6}$$

or

$$f_{A^q} \geq f_{A^*} \geq f_{A^p} and f_{B^q} \geq f_{B^*} \geq f_{B^p}, \tag{7}$$

The direction of the fuzziness change reverses (see e.g. on Fig. 2) if:

$$f_{A^q} \leq f_{A^*} \leq f_{A^p} and f_{B^q} \geq f_{B^*} \geq f_{B^p}, \tag{8}$$

or

$$f_{A^q} \geq f_{A^*} \geq f_{A^p} and, f_{B^q} \leq f_{B^*} \leq f_{B^p}. \tag{9}$$

3.4 DOUBLE FUZZY POINT EXTENSION OF TWO-STEP FRI METHODS

In this section, a novel extension of the two-step FRI method concept (following the generalized methodology [17]) is introduced to support double fuzzy point rule representations. Some additional properties of the newly obtained family of two-step double fuzzy point FRI methods are also examined, such as the validity of the conclusion and the preservation of linearity.

The proposed "Generalized Double Fuzzy Point Methodology" (GDFPM) can be applied as a guideline for the double fuzzy point adaptation of any two-step FRI method.

GDFPM can be used to extend any two-step FRI method with the ability to handle the $R^{p,q}$ double fuzzy point rule-base, and with a special additional step enabling the interpolation of fuzziness for the final conclusion. As GDFPM is based on two-step FRI methods, the limitations and preconditions of the original two-step FRI method used will be inherited by GDFPM.

The first step of two-step FRI methods is the generation of a temporal interpolated fuzzy rule at the reference point of the observation. In the case of double

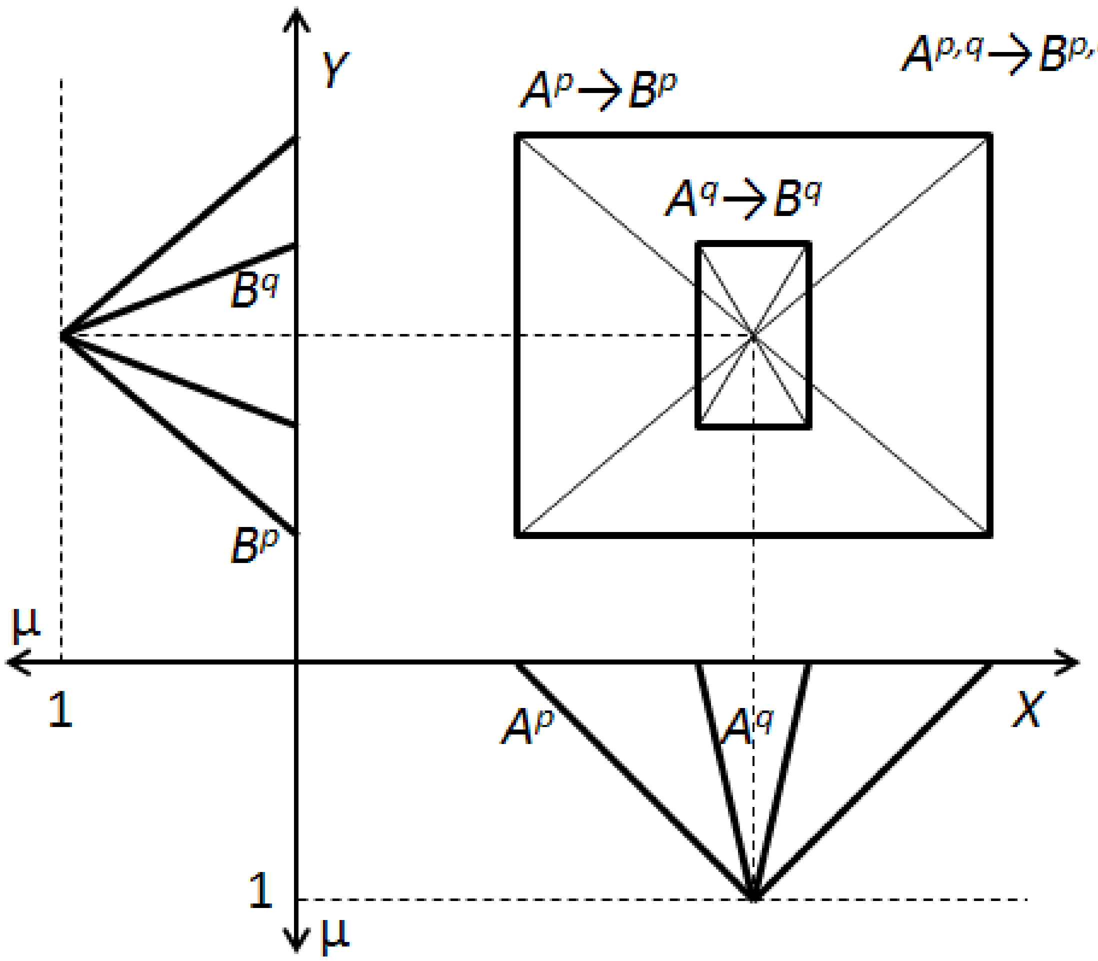


Figure 1: "Double fuzzy point" rule representation when the direction of the fuzziness change is remaining the same

fuzzy point extended FRI methods (GDFPM), a temporal interpolated fuzzy rule pair is generated, one for each of the two fuzzy rule sets R^p and R^q in the position of the observation. If the observation is within the fuzziness domain (4) covered by the antecedent fuzzy sets A_i^p, A_i^q in every input dimension, then the fuzzy conclusion can be obtained through interpolation. In other cases, the fuzzy conclusion can be considered as an extrapolation of fuzziness. In this chapter, extrapolation is not discussed.

The second step of the GDFPM method proposed here is the determination of the conclusion based on the observation (A^*) and the temporal interpolated rule pair ($A_i^p \longrightarrow B_i^p, A_i^q \longrightarrow B_i^q$) generated in the previous step. The concept of double fuzzy rule representation suggests that the property of "fuzziness similarity ratio preservation" should hold between the triplets A_i^p, A^*, A_i^q and B_i^p, B^*, B_i^q. Therefore,

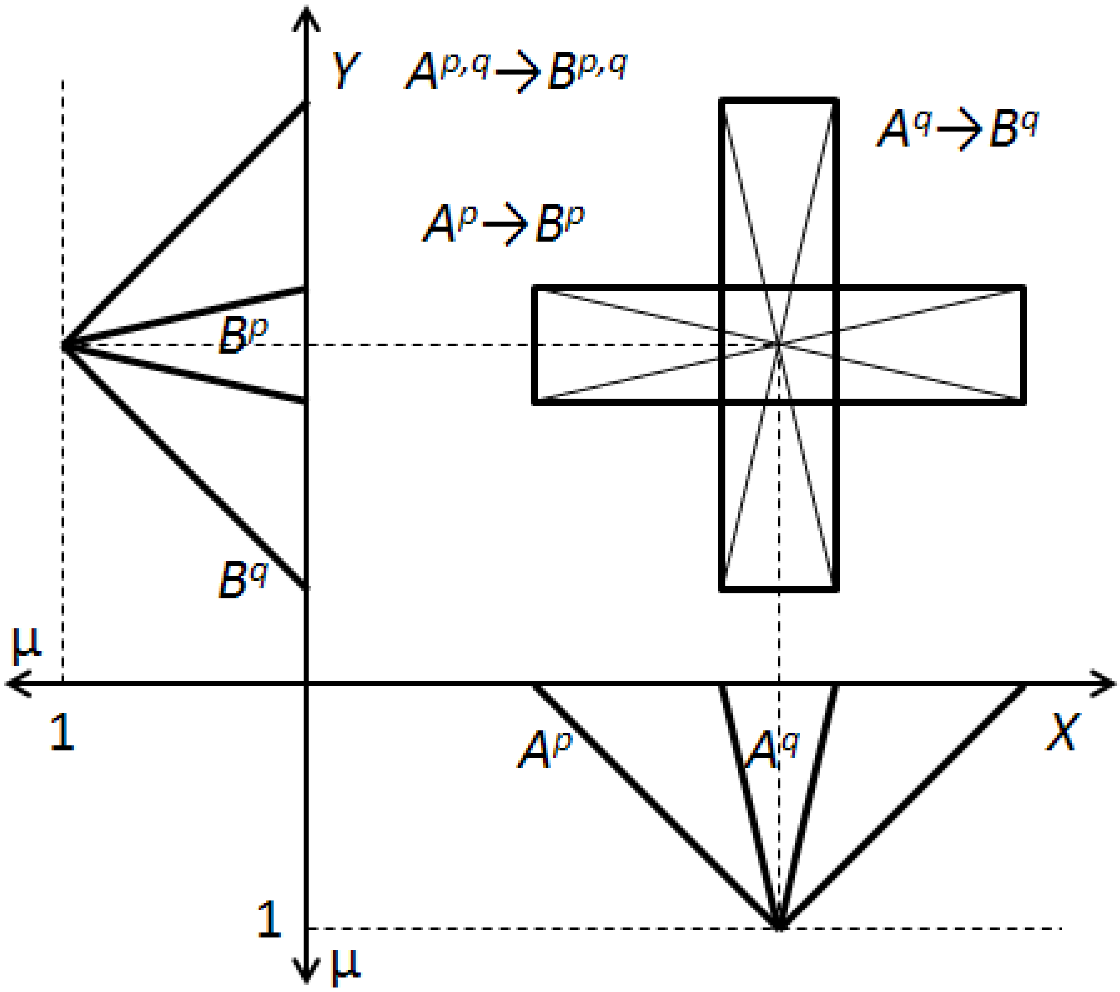


Figure 2: "Double fuzzy point" rule representation when the direction of the fuzziness change is reversing

as a final step of GDFPM, the single rule reasoning step of the original two-step method is replaced with a new "fuzziness similarity ratio preservation reasoning" step. The "fuzziness similarity ratio preservation reasoning" step is an extension of the common single rule reasoning concept. Rather than preserving the fuzzy similarity of the observation and the rule antecedent to the conclusion and the rule consequent, it preserves the fuzziness similarity ratio of the observation and the two antecedents of the double fuzzy rule to the conclusion and the two corresponding consequents. Generally speaking, the fuzziness similarity ratios must be equal on both the antecedent and the consequent sides:

$$fuzzinessRatio(A^{p,q}, x) = fuzzinessRatio(B^{p,q}, y). \tag{10}$$

This "fuzziness similarity ratio" is calculated in the same manner in which fuzzy similarity was calculated in the single rule reasoning step of the original two-step method, but this time it is calculated based on the double fuzzy rule. Therefore, as discussed earlier, the interpretation of the similarity ratio preservation strongly depends on the FRI technique used. In the following, the previously mentioned α-cut based and polar cut based fuzzy similarity calculations will be studied in more detail. When using α-cut based methods (e.g. [23], [21]) the similarity ratio can be determined based on the rate of α-cut distances (d_α) (see Fig. 3):

$$\begin{aligned} fuzzinessRatio_L(A^{p,q},x)_\alpha &= \frac{d_\alpha(K^p_{AL},K_{xL}))}{(d_\alpha(K_xL,K^q_{AL})},\\ fuzzinessRatio_U(A^{p,q},x)_\alpha &= \frac{d_\alpha(K^p_{AU},K_{xU})}{(d_\alpha(K_{xU},K^q_{AU})}, \end{aligned} \tag{11}$$

where L denotes the lower and U denotes the upper α-cut endpoint distances.

When using polar cut based methods (e.g. FRIPOC [2]) the similarity ratio can be determined based on the rate of polar distances (d_θ) (see Fig. 4):

$$fuzzinessRatio(A^{p,q},x)_\theta = \frac{d_\theta(K^p_A,K_x)}{d_\theta(K_x,K^q_A)} \tag{12}$$

Finally the conclusion y can be determined based on requirement of equality between the antecedent and consequent side fuzziness similarity ratio. In case of α-cut based methods (see e.g. on Fig. 5 and Fig. 6)

$\forall\alpha \in (0,1]$:

$$fuzzinessRatio(B^{p,q},y)_\alpha = fuzzinessRatio(A^{p,q},x)_\alpha, \tag{13}$$

$$\begin{aligned} \frac{d_\alpha(K^p_{BL},K_{yL})}{d_\alpha(K_{yL},K^q_{BL})} &= \frac{d_\alpha(K_AL^p,K_{xL})}{d_\alpha(K_{xL},K^q_{AL})},\\ \frac{d_\alpha(K^p_{BU},K_{yU})}{d_\alpha(K_{yU},K^q_{BU})} &= \frac{d_\alpha(K^p_{AU},K_{xU})}{d_\alpha(K_{xU},K^q_{AU})}. \end{aligned} \tag{14}$$

In the case of polar cut based methods (see e.g. on Fig. 7 and Fig. 8) $\forall\theta \in [0°, 180°]$:

$$fuzzinessRatio(B^{p,q},y)_\theta = fuzzinessRatio(A^{p,q},x)_\theta \tag{15}$$

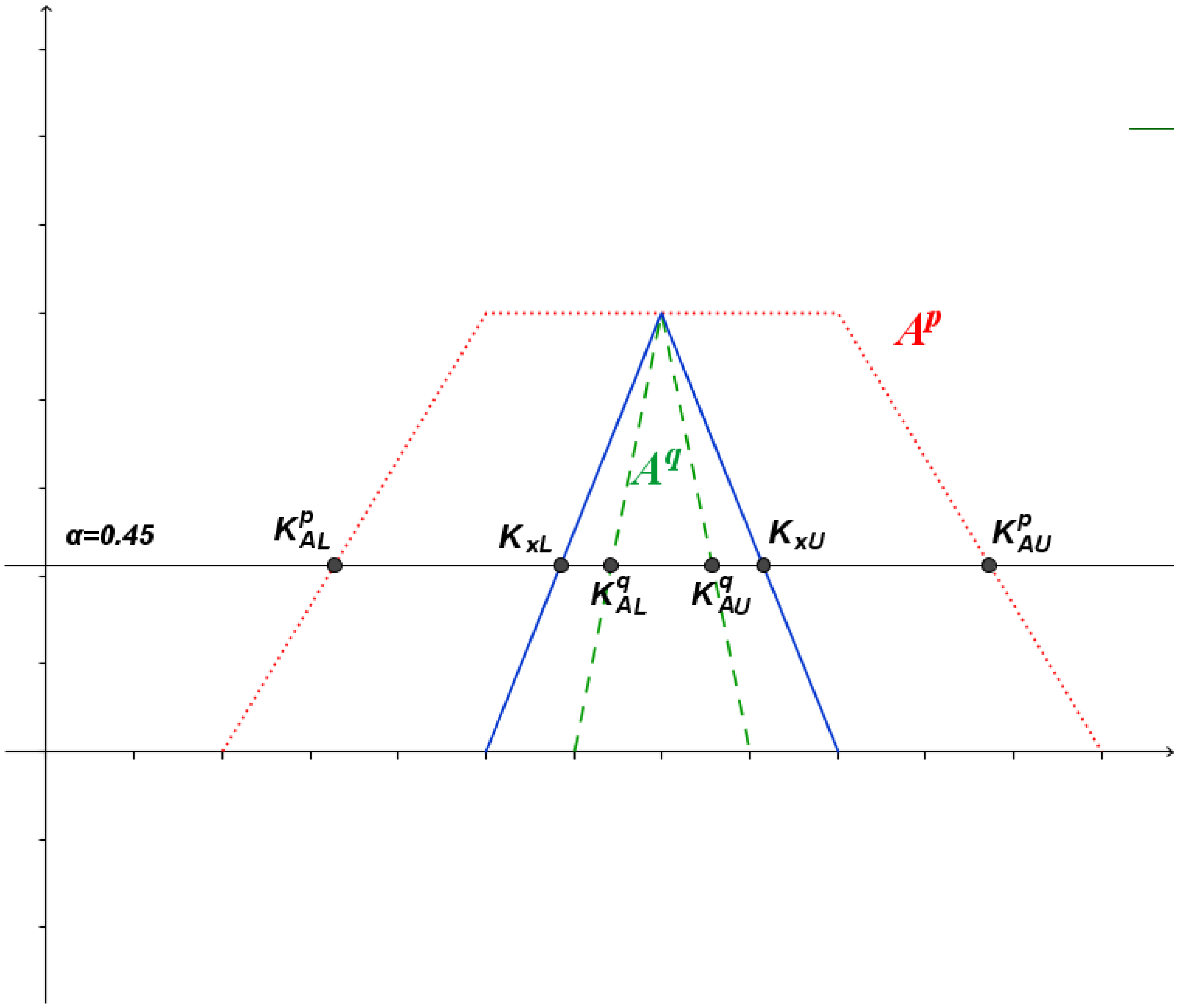


Figure 3: "α-cut based" fuzziness similarity ratio

$$\frac{d_\theta(K_B^p, K_y)}{d_\theta(K_y, K_B^q)} = \frac{d_\theta(K_A^p, K_x)}{d_\theta(K_x, K_A^q)}. \tag{16}$$

3.4.1 *Validity and shape of the Conclusion*

The aim of this section is to briefly check the validity and the shape of the conclusion, i.e. to check whether or not the generated conclusion is a valid fuzzy set in general, and whether or not it preserves the piecewise linear shape of the terms.

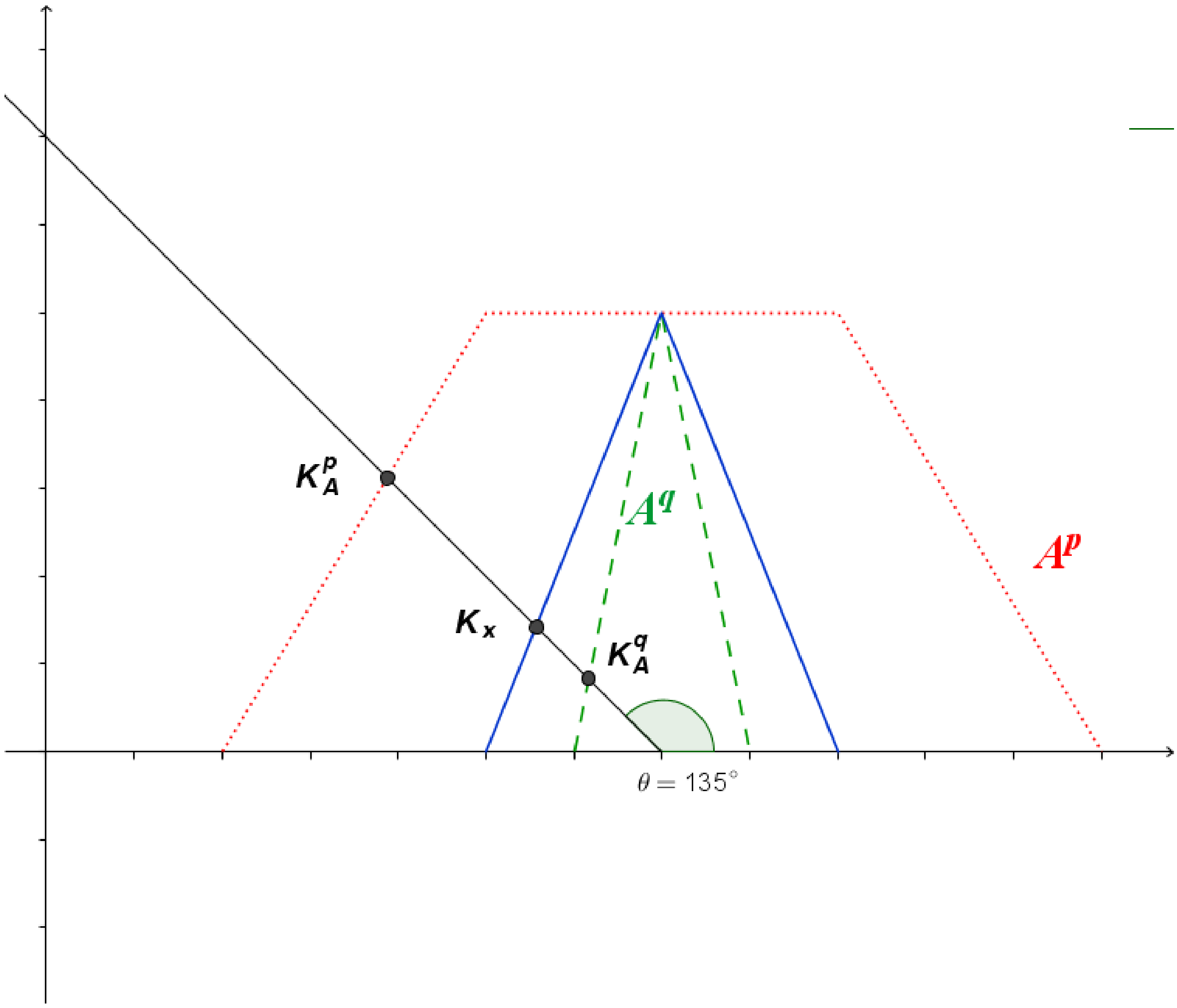


Figure 4: "polar cut based" fuzziness similarity ratio

The validity of a fuzzy set can be defined as the validity of the membership function [6]. A fuzzy set A is valid if:

$$\begin{aligned} \forall \alpha, \alpha_1 < q\alpha_2 \in (0,1] : \\ inf\{[A]_\alpha\} \leq sup\{[A]_\alpha\} and \\ inf\{[A]_{\alpha_1}\} \leq inf\{[A]_{\alpha_2}\} and \\ sup\{[A]_{\alpha_2}\} \leq sup\{[A]_{\alpha_1}\}. \end{aligned} \tag{17}$$

Remark The conclusion of GDFPM is not always valid. See e.g. the example on Fig. 9.

Remark GDFPM does not preserve the piecewise linear shape of the terms. See e.g. the examples on Figs. 10, 11 and 12.

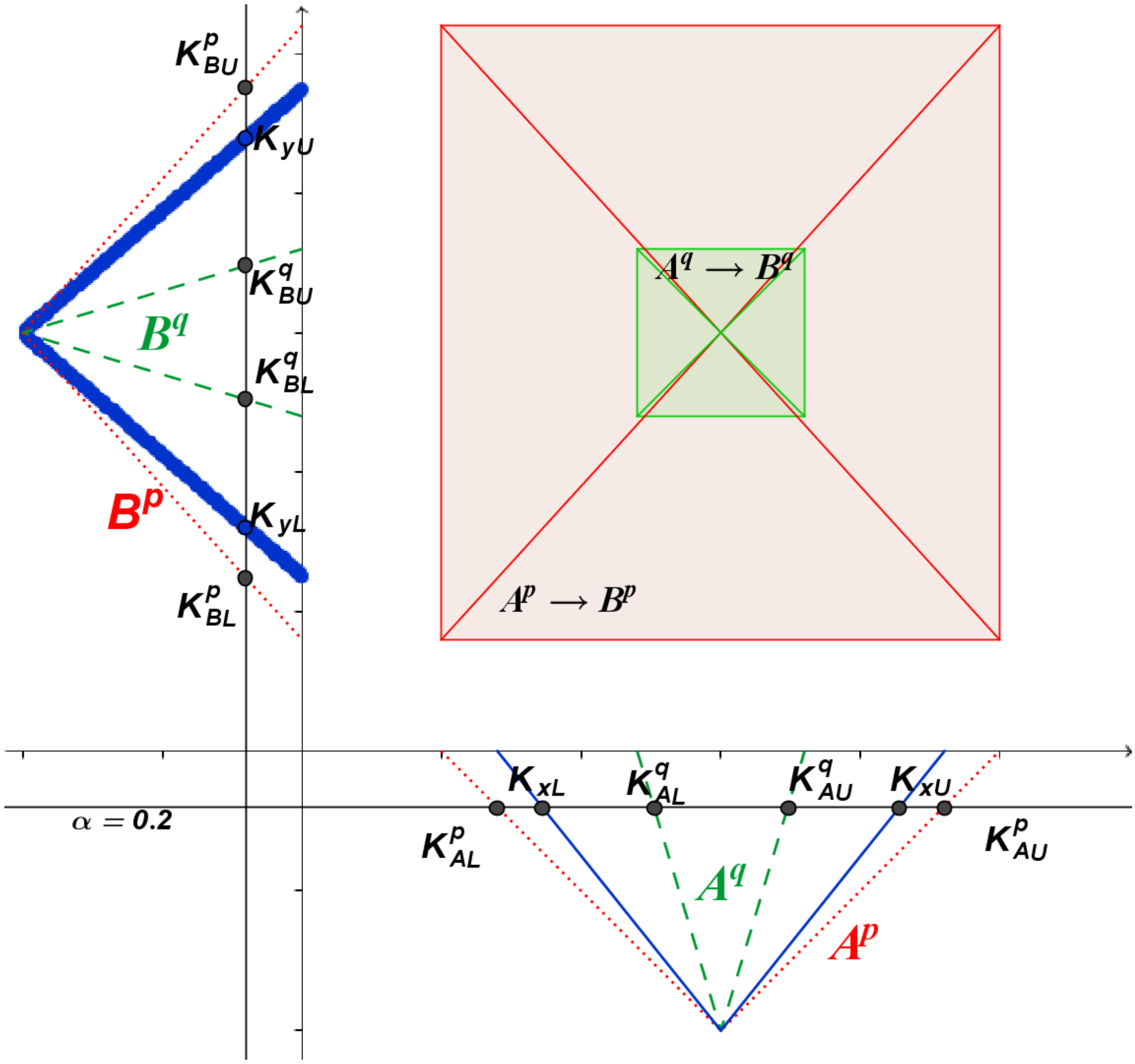


Figure 5: An example of the GDFPM α-cut based fuzziness similarity ratio preservation reasoning step, in which the conclusion is valid, and the direction of change in fuzziness remains the same.

Remark In the α-cut based GDFPM fuzziness similarity ratio preservation reasoning step, if all fuzzy sets involved (i.e., rule antecedents, consequents and the observation) are restricted to normal triangular shaped membership functions, the conclusion will also be a valid triangular shaped fuzzy set. See e.g. the examples on Figs. 5 and 6.

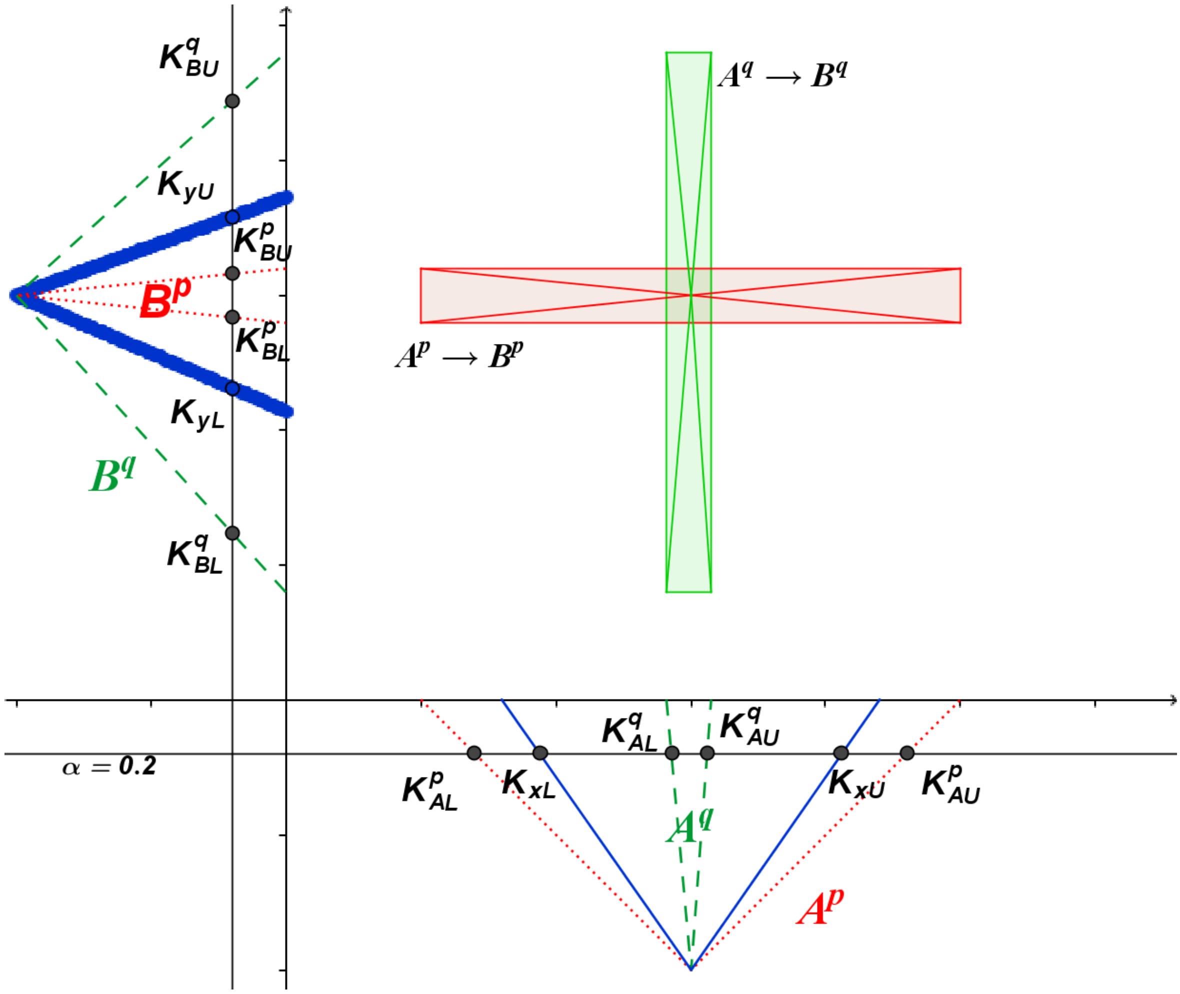


Figure 6: An example of the GDFPM α-cut based fuzziness similarity ratio preservation reasoning step, in which the conclusion is valid, and the direction of change in fuzziness is reverved.

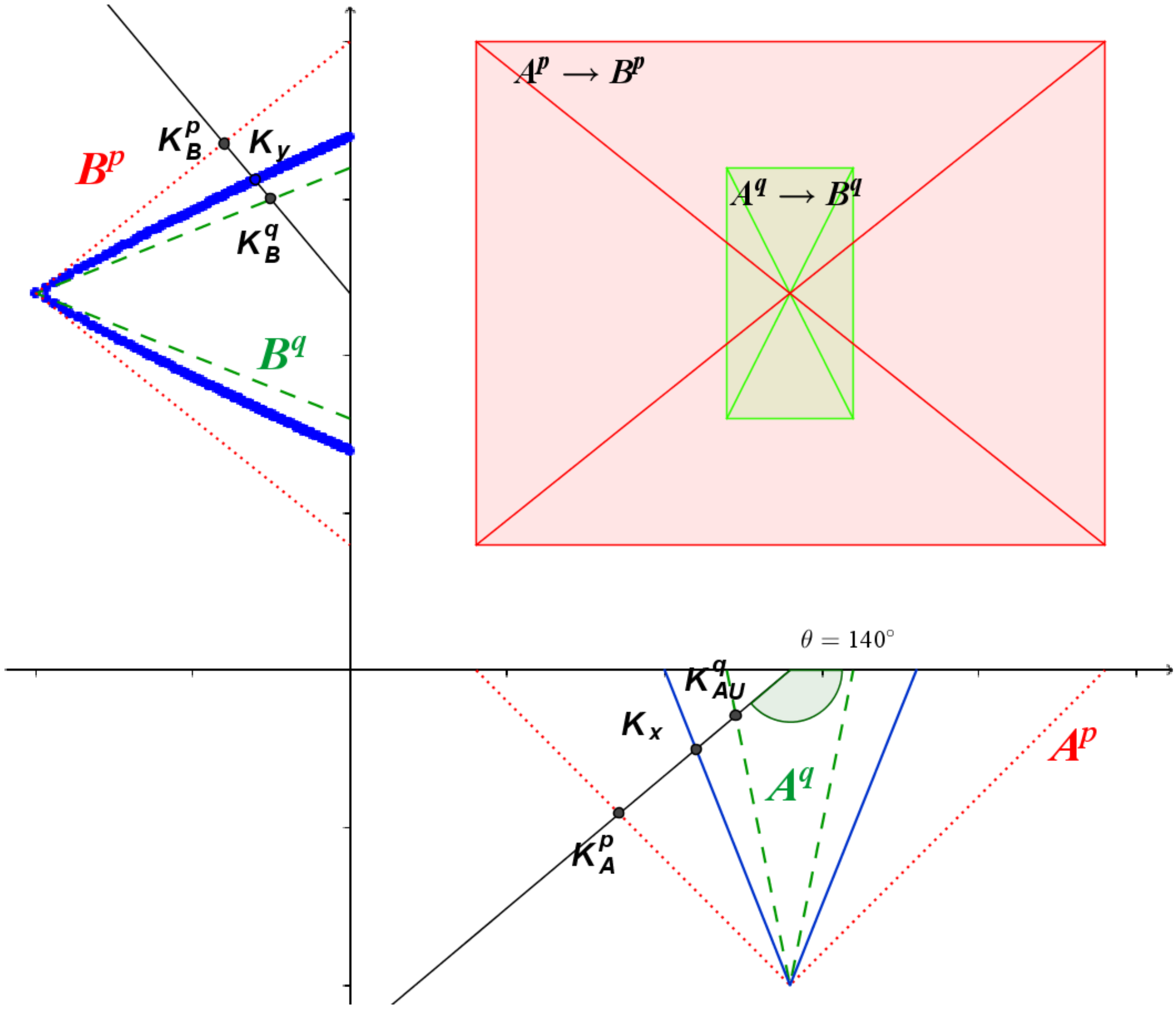


Figure 7: An example of the GDFPM polar cut based fuzziness similarity ratio preservation reasoning step, in which the conclusion is valid, and the direction of change in fuzziness remains the same.

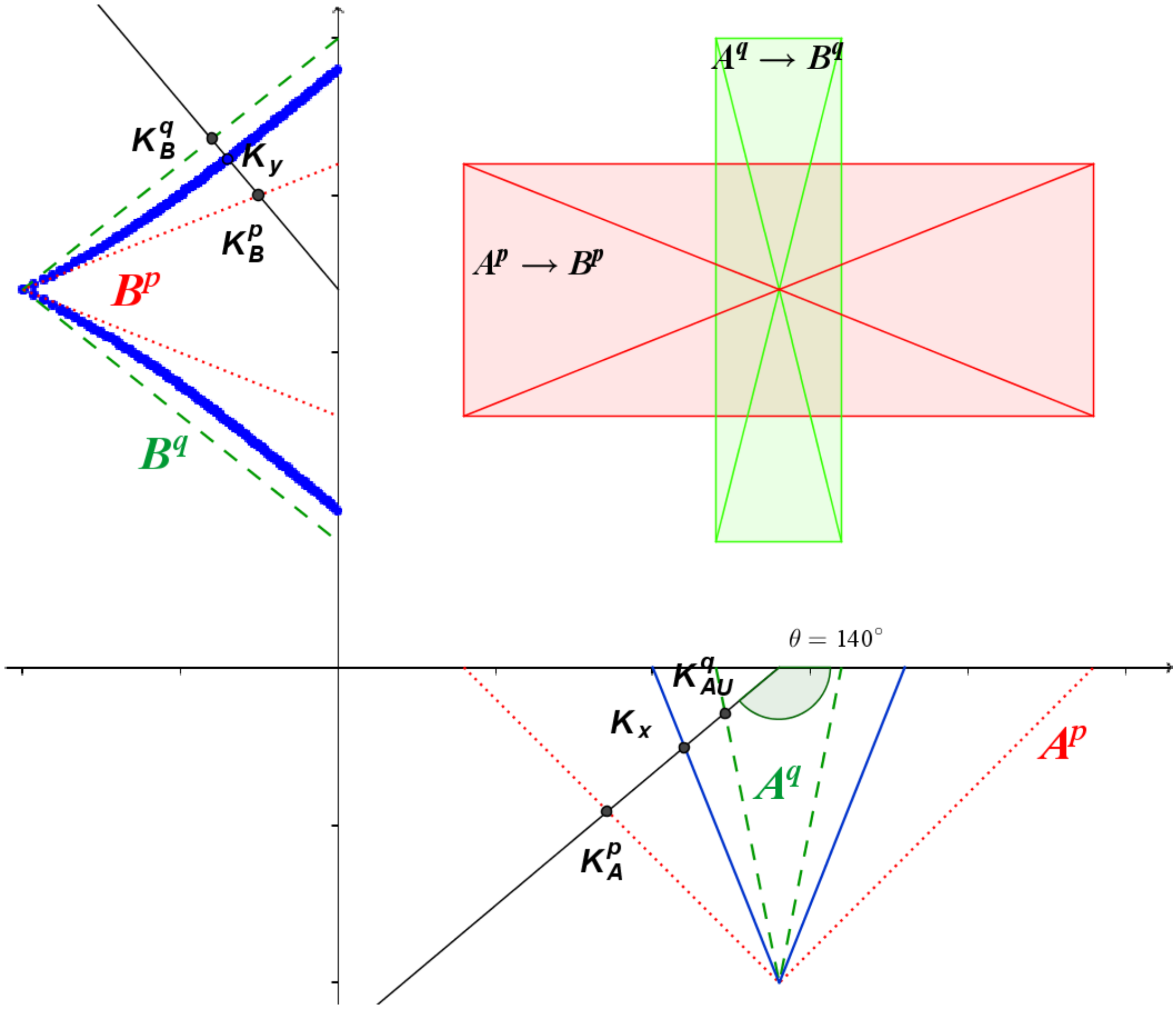


Figure 8: An example of the GDFPM polar cut based fuzziness similarity ratio preservation reasoning step, in which the conclusion is valid, and the direction of change in fuzziness is reversed

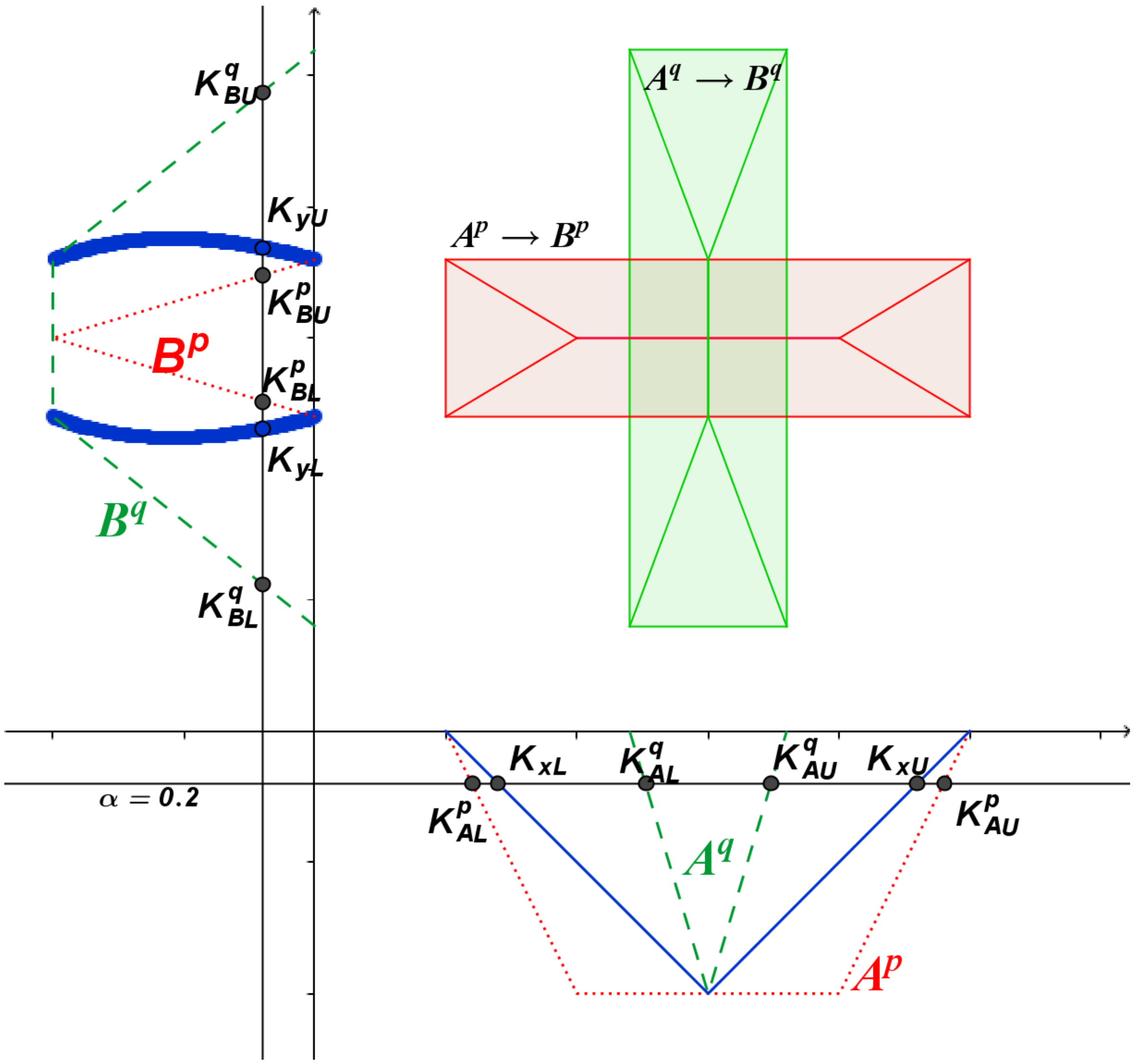


Figure 9: An example of the GDFPM α-cut based fuzziness similarity ratio preservation reasoning step, in which the conclusion is invalid, and the direction of change in fuzziness change is reversed.

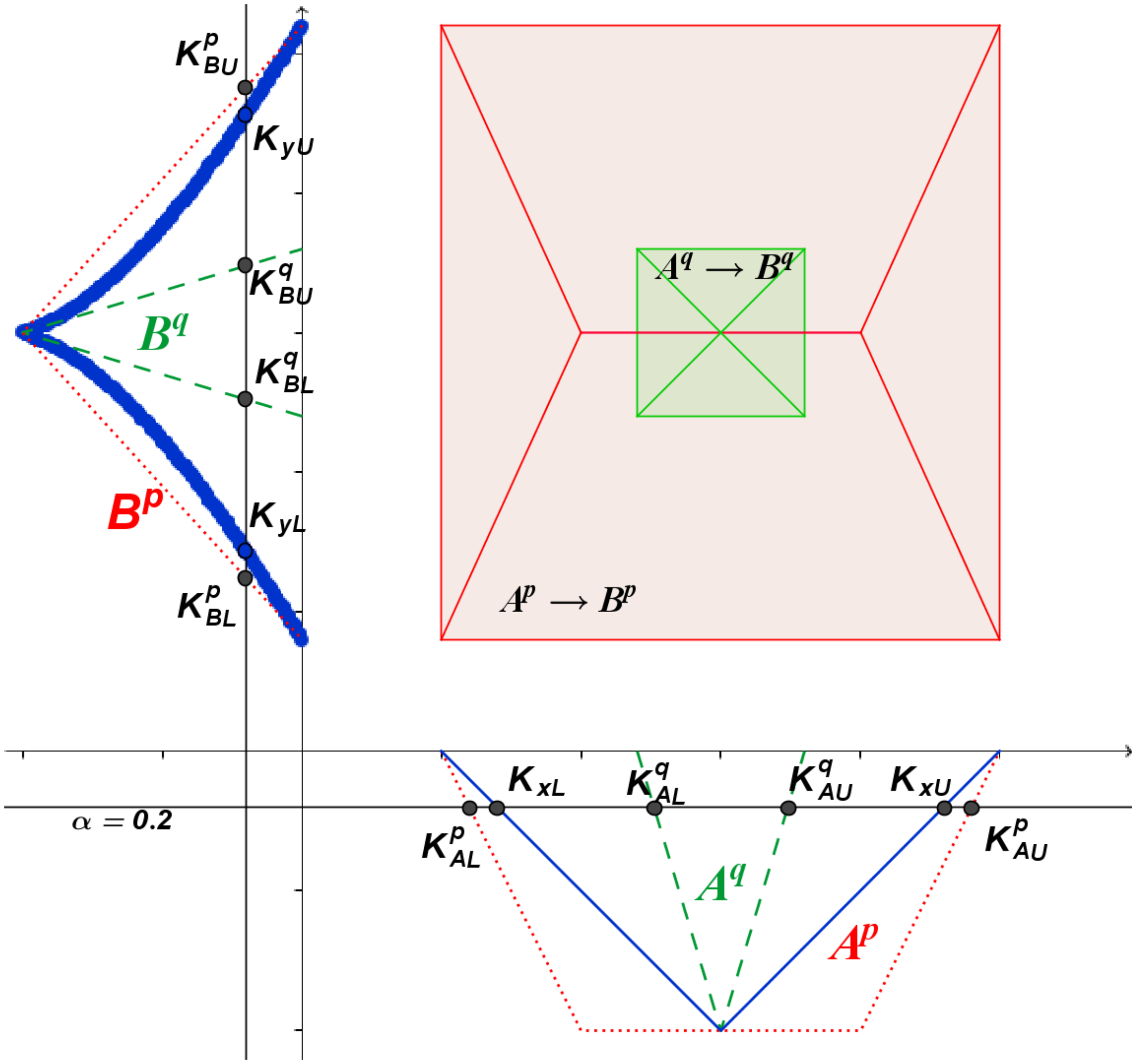


Figure 10: GDFPM α-cut based fuzziness similarity ratio preservation reasoning step, in which the conclusion is valid, but the piecewise linearity is not preserved.

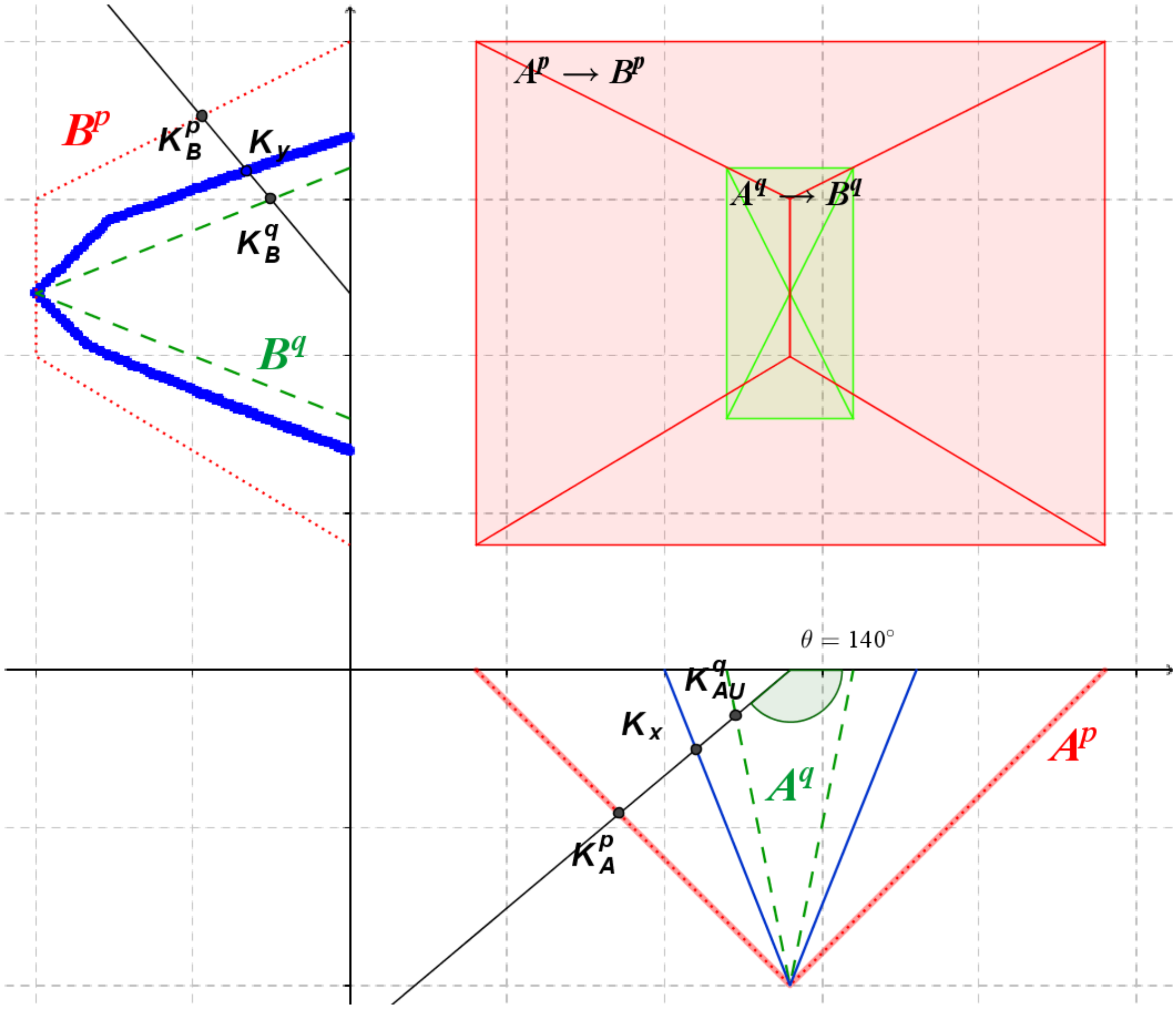


Figure 11: An example of the GDFPM polar cut based fuzziness similarity ratio preservation reasoning step, in which the direction of the change in fuzziness remains the same, the conclusion is valid, and the piecewise linearity is preserved.

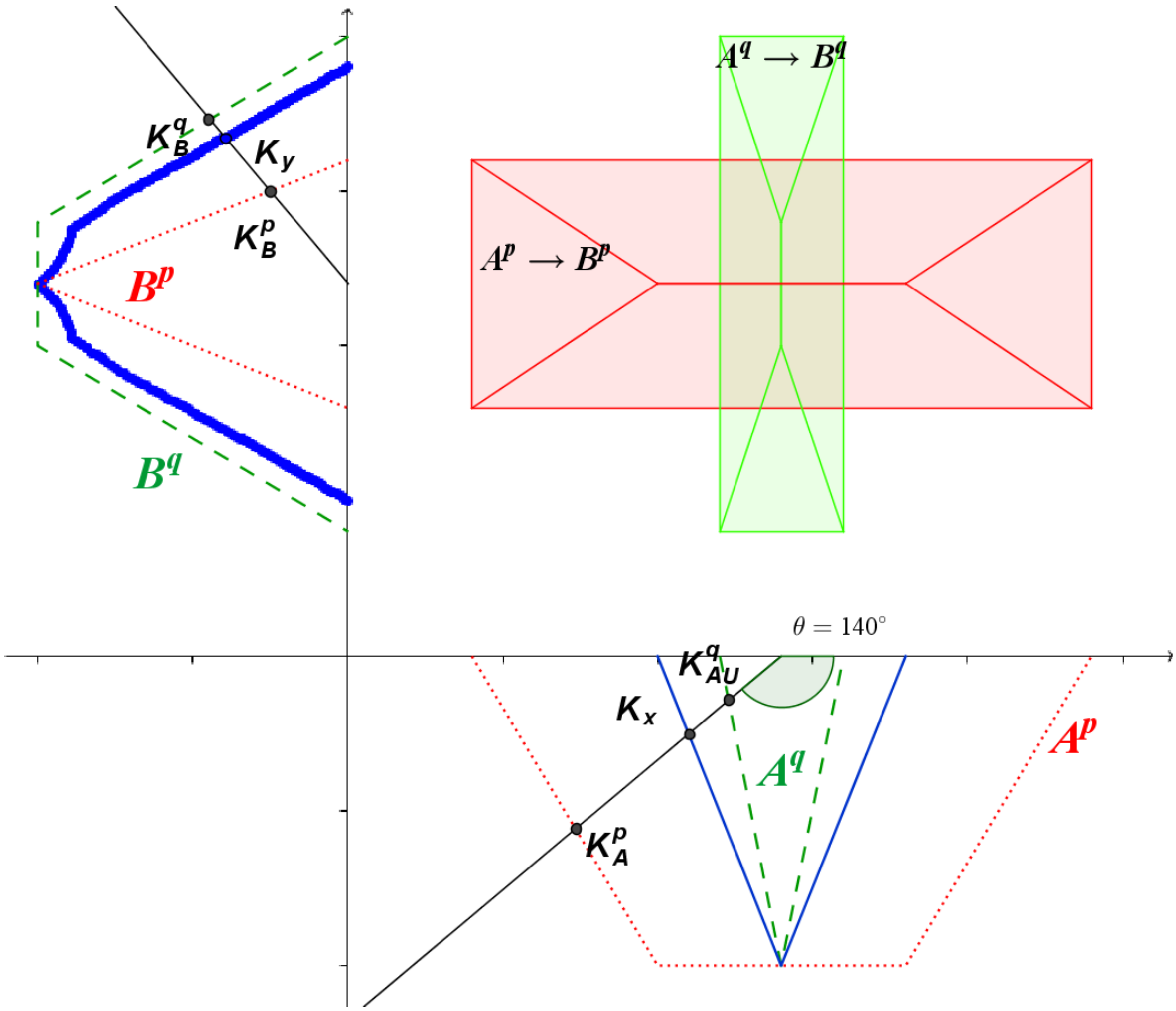


Figure 12: An example of the GDFPM polar cut based fuzziness similarity ratio preservation reasoning step, in which the direction of change in fuzziness is reversed, the conclusion is valid, but the piecewise linearity is nor preserved

3.5 CONCLUSION

In this chapter, a common "Generalized Double Fuzzy Point Methodology" (GDFPM) is introduced, and the questions of validity of conclusion and linearity preservation in the case of the obtained two-step double fuzzy point FRI methods were also briefly examined. The new scientific result summarized as follow.

The "Double Fuzzy Point Methodology" (GDFPM) is introduced, which can be applied as a guideline for the adaptation of the double fuzzy point representation in any two-step FRI method.

GDFPM replaces the single-rule reasoning step of the original two-step method with a new "fuzziness similarity ratio preservation reasoning" step. Two solutions are proposed for the "fuzziness similarity ratio preservation reasoning" step: one which is based on the "Least Squares Method" ("LESFRI"), and another which applies the "Polar a-cut" interpolation ("FRIPOC") concept. The guideline formulated based on these results suggests an extension that is based on the Fuzzy Rule Pair which defines the valid domain. The common methodology has twice as much space complexity as the original method, however, time complexity does not change significantly. Compared to the original two-step FRI method, the first step of the proposed GDFPM approach consists of the generation of a temporal interpolated double fuzzy point rule (this is a pair of rules: one for each of the fuzzy rule sets R^p and R^q) in the position of the observation. The second step of the proposed GDFPM approach consists of the determination of the conclusion based on the observation (A^*) and the temporal interpolated double fuzzy point rule ($A_i^{p,q} \longrightarrow B_i^{p,q}$).

Related publications: C1, C4, C7

4

FUZZY MODEL BASED DIFFERENTIAL IEC FOR HUMAN-SYSTEM INTERACTION

4.1 ABSTRACT

Human-system interaction as a form of "inter-cognitive" communication is an important aspect in computer assisted design systems in which direct human evaluation is an unavoidable intermediate step of the iterative design methodology. Such systems are significant application areas of Interactive Evolutionary Computation (IEC) approaches: approaches in which the evaluation of evolutionary fitness values are not machine computable. In interactive differential evolution, a human evaluation step is required for comparison between pairs of object samples. In some cases the objects need not be physically created order to be amenable to comparison, rather, it is sufficient if their virtual models can be viewed and appreciated in a virtual intelligent space. In this chapter, a possible application of (IEC) for a computer assisted interactive design system in virtual 3D system is proposed, using the Virtual Collaboration Arena (VirCA) system. To alleviate the burden of repetitive human interaction in the IEC process, a novel fuzzy rule based pre-evaluation functionality is also proposed, using which the human evaluator is assisted through a pre-evaluated fitness value approximated by an adaptive incremental Fuzzy Rule Interpolation (FRI) model.

4.2 INTRODUCTION

Inter-cognitive communication is a special view of human-system interaction, in which information transfer occurs between two cognitive beings with different cognitive capabilities, i.e. between a human and an artificially cognitive system [27]. Computer aided design is a typical application area in which a special form of inter-cognitive communication is needed, given that the parameters of the objects to be evaluated cannot be effectively displayed in any traditional way. Hence, a merging is needed between human and artificial cognitive capabilities, through a fusion of IEC and inter-cognitive infocommunication approaches.

IEC is based on the Evolutionary Computation (EC) methodology. EC is a general computational concept that is inspired by biology and can be applied based on a number of specific strategies [28] . It uses a combination of genetic algorithms (GA) [29, 30] and evolutionary strategies. The applicability of evolutionary computation is highly dependent on the existence of a machine computable fitness function. In other words, if there is no mathematical formula or explicit function which describes, or at least approximates the fitness value, evolutionary computation will not be applicable. Similar problems arise if the fitness value is computable, but the algorithm available for its computation has an unacceptable time complexity. Indeed, the lack of a machine computable fitness value renders many applications of evolutionary computational methods infeasible.

For such cases, it may be viable to use Interactive Evolutionary Computation (IEC) methods instead. These methods were introduced by Takagi [31, 32, 33]. In the IEC concept, the calculation of the fitness function is replaced by the evaluations of a human user (the "expert"), who can freely provide his/her opinion (acting as a fitness value) whenever requested by the system.

Two types of IEC systems exist. In most cases the expert can examine and compare more than a pair of individuals at any given time. In such cases up to 10-20 samples can be presented to the expert for evaluation. This can be continued through a number of generations (usually 10-20 generations). Applications of the approach have thus far focused on e.g. image processing [34] and signal processing [35].

The other type of IEC system presents only a pair of individuals to the user at any given time. As optimization is slower in such cases, the system is generally run for a larger number of generations (usually around 100-200). Nevertheless, this latter approach can be useful in applications where the goal is to evaluate music (see e.g. [36]) or animation (see e.g. [37]).

Evaluating one or two hundred individuals is a difficult task. Human fatigue and boredom is the single, most significant drawback of this computational method. Several enhancements – i.e., variants of IEC – have been introduced to alleviate this problem to at least some degree. One such enhancement consists in predicting the user's future evaluations based on a training set.

Differential Evolution (DE), and its modified variant, known as Interactive Differential Evolution (IDE), form another class of enhancements. DE was introduced by Storn et al. in [38]. A practical guide for its application is presented in [39]. According to this method, the initial vector population is chosen randomly from the entire parameter space. DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector. It uses difference of vectors for recombination, but also applies a machine computable fitness function to evaluate the individuals. IDE is different from this approach in the same way that IEC differs from EC: the evaluations of a human expert are used instead of a machine computable fitness function [40].

Takagi and Pallez made a detailed comparison of various IEC methods and suggested a new pairwise comparison based IDE (CBIDE) approach in [41] (2009). They suggested that only two individuals should be displayed at any given time, but recommended the pairs to be chosen using the IDE method.

One of the main difficulties in using pre-evaluation supported IEC models is the incremental manner in which such models are built: they have to be based on earlier evaluations of the same user. One possible solution to this challenge is to use a fuzzy rule-based approach. There are caveats to this approach, however. Even though it is possible to use fuzzy rule-based approaches in an incremental manner, classical fuzzy reasoning methods, like the Zadeh-Mamdani type compositional rule of inference [42], or the Takagi-Sugeno fuzzy model [43, 44], impose strict restrictions on the form of the antecedents of the fuzzy rules. Rule antecedents are required to cover the entire antecedent space, i.e. the fuzzy rule-base is required to be "complete". In case the rule base is not complete, but "sparse", there is no way to avoid the possibility of having observations without conclusions. In the case of pre-evaluation supported IEC, this would mean that objects with no evaluation could appear at any time. To avoid this difficulty, fuzzy rule interpolation (FRI) techniques can be applied. Rule-bases of FRI models are in many cases intentionally sparse, as FRI methods are capable of yielding acceptable conclusions through the interpolation of existing rules. In our case, this is a useful property: the completeness of the rule-base cannot be guaranteed during the incremental model generation.

In this chapter, I propose a pre-evaluation based extension to IEC methods with the goal of increasing the comfort of the user. The pre-evaluator predicts the user's evaluations based on a trained (fuzzy based) prediction model.

The other goal of the chapter is to suggest a new possibility to make human-machine interaction more tractable through the usage of the Virtual Collaboration Arena (VirCA) system. (VirCA) is a virtual laboratory that provides teams of researchers and industrial engineers with the opportunity to collaborate with each other any combination of virtualized machines. Through the augmented environment of VirCA, it is possible to control virtual and physical devices remotely in a simple way.

The VirCA system was developed by the Cognitive Informatics Research Group at the Computer and Automation Research Institute of the Hungarian Academy of Sciences. Further details about the system can be found in [45]. The main idea behind VirCA is to place any set of possibly distributed physical devices into a single virtual space, and to facilitate the communication among users, and between users and the devices.

In the following section, the pre-evaluation IEC theorem is presented in detail. In section 4.4, the theorem is applied to the VirCA system from a practical perspective. In section 4.5, a new version of the font design system is described as a working application example of the pre-evaluation based IEC system.

In this section, a new fuzzy model based pre-evaluation for IEC is detailed. IEC methods (including the CBIDE method, which is a complex IEC and DE method [46, 38]) optimize the target system based on user evaluations, such that the user acts as an embedded "black-box" evaluator.

The main goal of the suggested pre-evaluator is to enhance evaluation comfort by offering an approximated improved item to the user, which can be accepted through relatively less effort if it is appropriate. In remainder of the section describes the general pre-evaluation IEC approach, followed by the comparison-based pre-evaluation IEC approach.

4.3.1 *Pre-evaluation concept*

The pre-evaluation model proposed in this chapter is an FRI based approach. It adopts the set of evaluated feature parameters as fuzzy rule antecedents, and the user-provided rating values as conclusions. During the extension mode (phase) of the approach, each user opinion(except for contradicting ones) is stored as a multiple-input single-output fuzzy rule in the pre-evaluation FRI model. The antecedent of each fuzzy rule represents the feature parameters of the target system, and the consequent is derived from the rating of an individual. Formally, the rules can be expressed as follows:

$$\begin{aligned} if\ x_1 = FP_1\ and\ x_2 = FP_2\ and\ \ldots\ and\ x_L = FP_L \\ then \quad y = M_1 \end{aligned} \tag{18}$$

where FP is the Feature Parameter, M is the mark (ranking of the output) value, and L is the number of feature parameters. The mark value is an integer from the set $[1, 2, 3, 4, 5]$.

The main part of the pre-evaluation IEC model is the fuzzy subsystem that contains sparse rules. In this case the reason for using a sparse fuzzy system is the incomplete and continuously growing nature of the knowledge: in other words, less information is available about the system than would be required for complete rule bases. However, there are numerous fuzzy rule interpolation (FRI) methods which can yield fuzzy conclusions even in such cases ([47], [48], [7], [49]). In this dissertation, I use the Fuzzy Interpolation in Vague Environment (FIVE) method, which was introduced by Sz. Kovács [50], [51], [52]. The method requires a relatively short amount of time for its reasoning process, and is therefore suitable for practical applications.

FIVE is based on the concept of vague environments. The concept of vague environments is based on the similarity of elements. It uses scaled distances to determine the similarity between fuzzy sets, or fuzzy sets and singletons. For this purpose, FIVE adopts the Shepard interpolation [53], which is a simple formula for multidimensional interpolation. Applying this idea, the linguistic terms or fuzzy sets of the fuzzy partitions can be described through a scaling function and the fuzzy reasoning process itself can be replaced by classical interpolation. Hence, the fuzziness of the fuzzy partition can be translated to scaling functions, and the problem of fuzzy interpolation reduced to classical crisp interpolation.

The initial scaling functions of the antecedent vague environments of FIVE are set to constant. The scaling functions change in the adaptation mode only. In the adaptation mode (rule base adaptation mode) of the suggested pre-evaluation model, the fuzzy rule consequents are optimized to better match the ratings given by the user.

The applied optimization method is the gradient descent consequent optimization for FIVE. Consequent optimization is based on a set of sample (training) data. The goal of the optimization method is to minimize the squared error E of the training data and the fuzzy model:

$$E = \sum_{k=1}^{N} (y_d(x_k) - y(x_k))^2 \tag{19}$$

where $y_d(x_k)$ is the desired output of the k^{th} training data, $y(x_k)$ is the output of the FIVE fuzzy model, and N is the number of training data points.

The applied steepest descent parameter optimization method modifies the rule consequents based on the partial derivatives to the squared error function E 19 in the following manner:

$$g(c_k) = \frac{\partial E(c_k)}{\partial c_k} = \frac{\partial E(c_k)}{\partial y(x)} \cdot \frac{\partial y(x)}{\partial c_k} \tag{20}$$

$$c_{k\_next} = c_k - \tau \cdot g(c_k) \tag{21}$$

where τ is the step size of the iteration and c_{k_next} is the next iteration of the k^{th} conclusion c_k.

According to 19, 20 can be rewritten in the following form:

$$g(c_k) = -2 \cdot (y_d(x_x) - y(x_k)) \cdot \frac{\partial y(x)}{\partial c_k} \tag{22}$$

Applying the Shepard interpolation formula of FIVE to the partial derivatives, the following formulae are obtained:

$$\frac{\partial y(x)}{\partial c_k} = \begin{cases} 1 & if\ x = a_k \text{ for some } k \\ \left(1/d^{\lambda}_{s,k}\right) / \left(\sum_{k=1}^{r} 1/d^{\lambda}_{s,k}\right) & \text{otherwise.} \end{cases} \tag{23}$$

The next iteration of the k^{th} conclusion c_k can be calculated based on 21, 22 and 23. In the suggested pre-evaluation model, the training set is built from the antecedent-consequent pairs of the actual fuzzy model: the newly given evaluated feature parameters and the user provided rating values, taken as input-output pairs.

We suppose that the user can change his or her opinion during the evaluation, so the answers can be contradictory. Hence, filtering earlier answers is necessary every time a new value is added to the rule base. If an earlier rule exists whose antecedent is closer than ϵ to the new antecedent, and consequent is farther away than δ to the new consequent, then the two rules are in contradiction:

$$d(A_{Ri}, A_{Rj}) < \epsilon \text{and} C_{Ri} - C_{Rj} > \delta \tag{24}$$

where d is Euclidean distance, A is the antecedent of the i^{th} and j_{th} rule, C is the consequent of the i_{th} and j_{th} rule, and ϵ and δ are non-negative numbers.

The new rule is then taken to replace the older, contradictory one.

In the beginning, the model has no rules. In every iteration step, the system obtains a set of feature parameters – generated by the IEC system with the help of a genetic algorithm – as input, and then calculates the mark value using the rule base. The pre-evaluator passes this calculated value to the user. If this value is in agreement with the user's opinion, then it is left untouched. The pre-evaluator then uses the final value to perform a coherency test.

Proposition 4.3.1 *The modification of the rule base depends on the current mode, i.e.*

- *In the rule base extension mode*
  - *If the new rule is in contradiction with any existing rules in the rule base, then the system replaces the consequent of the contradictory rule.*
  - *If the new rule is not in contradiction with any existing rule in the rule base, then the new rule is added to the rule base.*
- *In the rule base adaptation mode*
  - *If the new rule is in contradiction with any existing rule in the rule base, then the system replaces the consequent of the existing contradictory rule.*
  - *If the new rule is not in contradiction with any existing rule in the rule base, then the system optimizes the rule base according to the new training value.*

4.3.2 *The role of pre-evaluation in comparison based IEC*

In comparison-based IEC, only two individuals are presented by the system and evaluated by the user as suggested in CBIDE. In order to assist the user, the model chooses one individual from the two possibilities. One possibility is the winner of the previous iteration, while the other is a new individual that is assumed to be the best of the current population. There are two kinds of IDE systems, depending on how the two alternatives are presented to users, i.e. in parallel or in sequence. In the parallel mode, the two individuals are visible side-by-side at the same time. However, this is not always possible. If the objects of evaluation cannot be shown at the same time, as in the case of, e.g., voices or live demonstrations (such as the movement through time of a mobile robot), then the two examples are presented in sequence. This is referred to as the sequence mode.

One key question is how the individuals are shown in the two modes. The proposed individual can be emphasized by its size, a surrounding frame, a stronger shadow, vivid graphics, and other visual cues in parallel mode. In this mode, user selection can be performed through a single click. In the sequence mode, the preferred item can be given extra weight through emphasized volume, or colorful versus grey-scaled appearance.

After the two individuals are shown, the human evaluator has two possibilities. He or she can either choose the pre-selected, emphasized individual, or the alternative one. The selection can be done by voice commands or through the GUI. The winner of the iteration is then used as a candidate individual in the following step.

The pre-evaluation system has to predict the user choice based on earlier selections. The predicted decision can be modeled via multiple-input single-output fuzzy rules, the antecedents of which represent the feature parameters of the target system and the consequent of which represent the rating of the current individual. The rules have the same form as previously introduced in Eq. 18.

As in the general case, initially the model uses an empty rule base. In every iteration step, the system obtains the feature parameters - generated by the genetic algorithm - as input and calculates a goodness value using the rule base. This goodness value is a real number from the interval $[0..1]$ (not an integer from the set $[1,2,3,4,5]$ as previously suggested. In the first step, because of the empty rule base, two individuals are selected at random from the population. After the user chooses one of them, the losing sample is dropped, while the goodness value of the winner is set to 0.5. The system stores a new rule with the current feature parameters as its antecedent and 0.5 as its consequent.

In the j_{th} iteration ($j > 1$), further individuals are generated. The parent of each population is the individual that wins the previous iteration. The user is presented with two individuals: the first one is the winner of the $j-1$-th iteration, and the second one is the best individual of the current population, based on its goodness value. If more than one individual has the same, maximal goodness value, then one of them is selected at random. After a new individual pair is created, the system compares the GV values of the two, and the one with the higher GV value is selected as a candidate to win the selection.

The user can change the selection explicitly or can leave the system's recommendation untouched. If the user does not pick an individual, then the system waits for a given time (timeout) and the pre-evaluated individual is selected as a winner.

If the user chooses an individual or the timeout is exceeded, then the system continues with next iteration. If the selected individual is the one that was not initially predicted by the system, then the rule base is fine-tuned so that the goodness values of both individuals are updated according to:

$$gv = (1 - b) * GV + b * u \tag{25}$$

where gv is the calculated new goodness value of the individual, u is the update step (+1 or -1), GV is the goodness value determined by the fuzzy model, and b is the step value. The value of u is $+1$ in the case of the selected individual, because its evaluation is better than was predicted, and -1 in the case of the losing individual, because its prediction was too optimistic. If the calculated goodness value is not the same as predicted ($gv \neq GV$), then model has to be modified. If the rule base already contains an appropriate rule, then the consequent has to be modified; otherwise a new rule has to be added whose antecedent corresponds to the current feature parameters, and whose consequent matches the current goodness value.

After K steps the rule base of the system has the following form:

$$\begin{matrix} FP_{I\_1} & FP_{I\_2} & \dots & FP_{I\_21} & GV_I \\ FP_{II\_1} & FP_{II\_2} & \dots & FP_{II\_21} & GV_I I \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ FP_{N\_1} & FP_{N\_2} & \dots & FP_{N\_21} & GV_N \end{matrix} \tag{26}$$

where $I, II, \dots, N$ are the iteration steps ($N \leq K$, because the system updates the rule base in the case when the user's choice is different than was predicted), FP is the Feature Parameter, and GV is the goodness value. After K steps the rule base contains up to K rules.

Using the VirCA system as a virtual space in which to display individuals requires a special topology. If the individuals are shown as 3D objects, the user can rotate them and see them from all perspectives. The sound of individuals can should be heard in 3D space, thus the inclusion of audio sources can be important. In most cases, the development of virtual spaces is a difficult task that requires special knowledge and skills. VirCA, however, provides a relatively simple way to develop simulations in modeling applications. Moreover, the cooperation between components and systems are implemented a priori, and do not require any additional programming. They can simply be configured in real time.

The proposed topology of components is shown on Fig. 13. Human-system collaboration generally requires some interaction techniques that should fulfill the following criteria.

Requirement 4.4.1 *The user has to perceive the examples in space, e.g., as two different shapes or two different objects. Depending on the type of the selection task, the user can feel the examples through audition, vision or the tactile/haptic sense. In VirCA, several modalities are covered through various sensor components, and further components can be developed through the VirCA SDK based on existing examples.*

Requirement 4.4.2 *The system has to pre-select one of the two examples and indicate its selection, as well as the marks of the examples in 3D space. The main point here is that the suggested selection should be emphasized in a clear and intuitive way. In the VirCA system, a number of possibilities are available to emphasize a scenario or object. Visualizations can be emphasized through size, color or 3D position.*

Requirement 4.4.3 *The user should be able to change the marks and toggle between selected individuals. The system has to provide users with the possibility to change the state of the system in a way that appeals to their cognitive capabilities. VirCA can handle user selection using various inputs such as keyboard input, voice command, gesture by through cameras or Kinect sensors, etc. A number of components have been developed which can accomplish this selection task. After the user selects the virtual object, VirCA notifies the ICE system.*

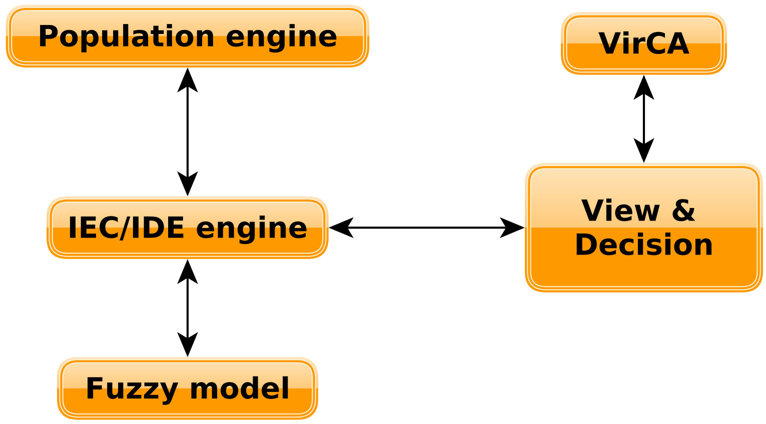


Figure 13: The suggested topology of the IDE system extended by the pre-evaluation module, the VirCA components and connections

The universality of the solution is an important criterion during its design, therefore it is advisable to use a minimal number of components. In the suggested topology, five components and four two-way connections are configured.

In the proposed topology the following components are configured to work together:

Population Engine: at the moment the engine is a Genetic Algorithm implementation. It sends the population in terms of a set of feature parameters to the "IDE engine". The "IDE engine" initializes this module by setting the number and the range of the feature parameters. Later the "IDE engine" requests a new population from this module in the beginning of every iteration. The Population Engine component does not contain any fitness value function. It is independent from the given task.

IEC/IDE Engine: the main component which controls all other components either directly or indirectly. It is initialized by the "View & decision" component and initializes the "Population Engine" and the "Fuzzy Model". In the case of IEC systems, this component is replaced by he IEC Engine.

Fuzzy Model: it represents a pseudo human predicting the user's opinion. The component stores the fuzzy model and determines the conclusion based on previous decisions of the user. This component was implemented using the FRI Toolbox native developer library (more details can be found in [54]).

View & decision component: this is the main interface between the user and the IDE system. It provides the sensory manifestation of individuals to VirCA and consumes various events from VirCA. This component depends on the given task. It registers two models and moves them into the 3D space. It obtains user feedback from VirCA. It depends on the current task, which can be configured through an external configuration file.

VirCA: it is responsible for the user interface and the virtual events. The user can see or feel the individuals and can choose one of them or change the mark values inside the 3D space of this component.

In every iteration the following procedure (see Fig. 14) is executed:

1. A new population is generated by the "Population Engine" and sent to the "ICE/IDE Engine". In the case of CBIDE the parent of the population is the previous winner individual (if one exists).

2. A goodness value is determined for each individual based on the fuzzy model.

3. The decision system registers the appropriate 3D objects into the VirCA 3D space according to the feature parameter sets and their goodness values. In the case of CBIDE, the suggested item with the higher goodness value is emphasized. The best individual is chosen from the given population by the "ICE/IDE Engine" based on goodness values. If more than a single individual has the same goodness value, then a randomly selected individual is selected among the ones with the highest goodness values. The "ICE/IDE Engine" sends the feature parameter sets and the goodness values to the "View & Decision" component.

4. The "View & Decision" component waits for a timeout.

5. After the timeout period is exceeded, or when the user picks one of the individuals in the case of CBIDE, the "View & Decision" component sends the selected item to the "ICE/IDE Engine". In other cases the user might change the goodness value of any individual.

6. All necessary modifications on the rule base are effected. If the user chooses the individual that was not suggested by the system, or reevaluates the good-

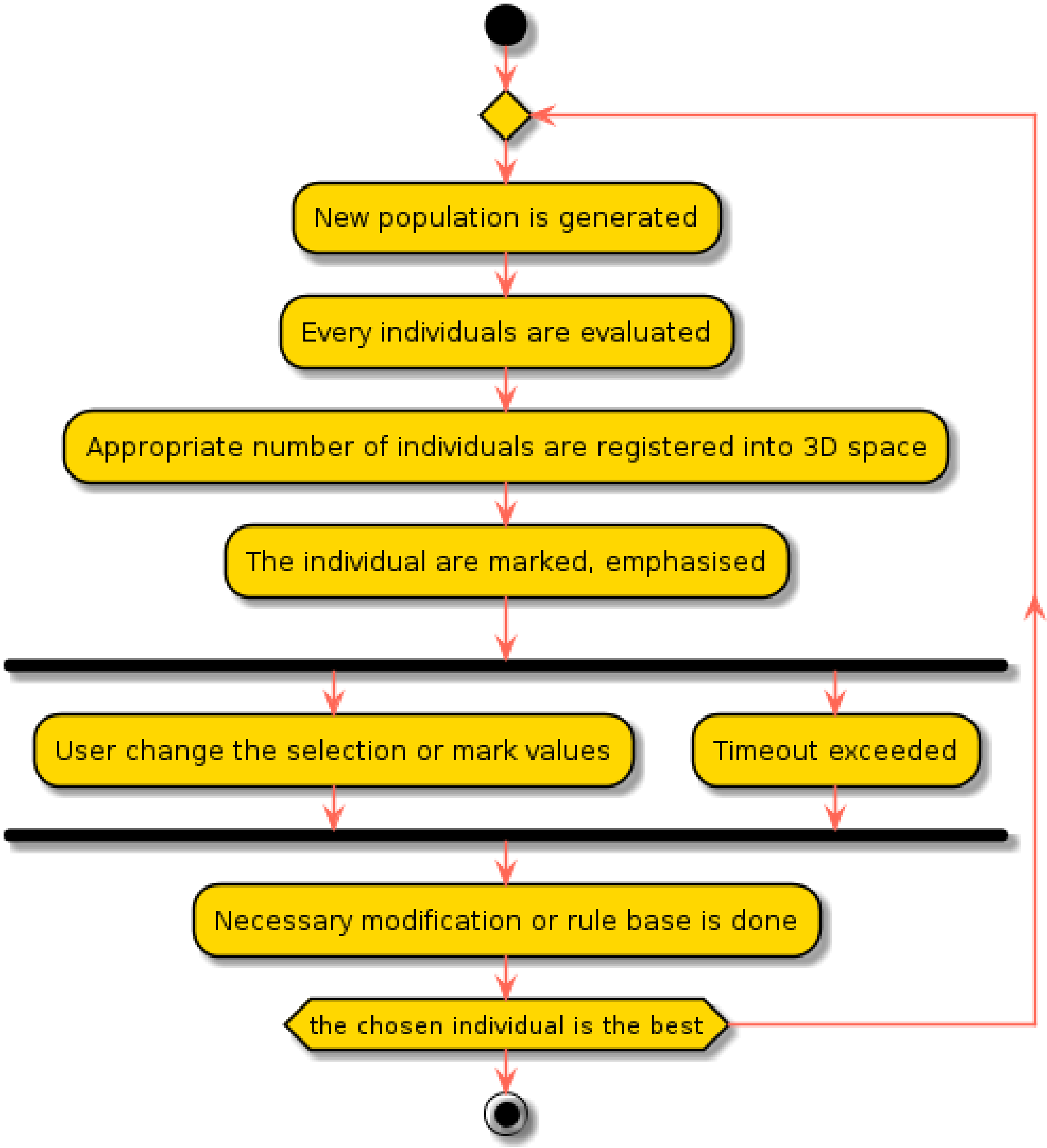


Figure 14: The process flow of the extended IEC/IDE system

ness value of any individual, then the "ICE/IDE Engine" modifies the goodness value of the appropriate individual and sends it to the fuzzy model according to eq. 24, 25 and proposition 4.3.1 . If the fuzzy model receives a rule modification request from the IDE, then it modifies the internal rule base.

7. Finally the “Population Engine” receives the selected individual, which will be the parent of the following population.

Using the services of the OpenRTM-aist robot middleware, which serves as the foundation of network communication in VirCA, the user can select and connect the online components. It should be noted that the prediction component (the Fuzzy model component in this case) is not a mandatory part of the system, as any ICE or IDE system can work without prediction. Hence, the fuzzy prediction model is interchangeable, that is, it can be easily substituted by any other method (for instance, an artificial neural network). The prediction model has been implemented using a pre-defined interface so the user is can connect any component at runtime.

The concept of VirCA allows the user to create his or her own system and evaluate the item-pairs in a single place, inside a main window. Moreover, collaboration among IDE systems is possible because the system supports a distributed topology. Thus, the components of the system can be in different places, in a way that is transparent to the user. Collaboration is also supported by VirCA, which means that distributed, multi-user applications of IEC can also be developed.

4.5 APPLICATION EXAMPLE: FONT DESIGN SYSTEM

The main goal of the font design system is to create a personalized font set based on human subjectivity. An IEC based font design system is introduced by M. Kuzma et al. in [55], which focuses on the design and implementation of a system aiding the user to create a personalized font. It reduces the time required for the design process. The system starts from a randomly selected font set. It uses an elitist genetic algorithm for generating a population of fonts which is shown on the user interface. We have introduced a pre-evaluation method based on a fuzzy model for that IEC system (cf. [56]) and a novel extended version, which was implemented in the VirCA environment, is proposed here.

The new user interface appears in 3D space. The control panel gives control over the Font Evolving System. The user can choose an action to change the population. The possible actions are selecting an item by clicking on it, reset, reset random, and export font. In the 3D environment, two font sets are shown in parallel and the user can select one at given time. The font set that is predicted to be better is slightly bigger than the other. If the user does not evaluate the font samples, the system chooses the predicted one after 2 seconds. The procedure then continues with the next round. There are 21 font parameters (the feature parameters of the target system) which are generated by a genetic algorithm. Fonts are drawn according these parameters. In accordance with eq 18, the font design system can be modeled via multiple-input single-output fuzzy rules, where FP is the Font Parameter and the goodness value (output) is GV. In this case, the observation is represented by 21 font parameters and the conclusion is a real value. Initially, the model has no rules and is empty. During the iteration steps, the pre-evaluation model generates an approximated goodness value for each of the font set parameter configurations appearing in the 3D virtual space. If this value is in agreement with the user's opinion, then the user leaves it unchanged. At the end of each iteration, updated mark values are sent back to the IDE Engine via the "View & Decision" component.

After K steps the rule base of system has the following form:

$$\begin{matrix} FP_{I\_1} & FP_{I\_2} & \dots & FP_{I\_21} & GV_I \\ FP_{II\_1} & FP_{II\_2} & \dots & FP_{II\_21} & GV_I I \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ FP_{N\_1} & FP_{N\_2} & \dots & FP_{N\_21} & GV_N \end{matrix} \tag{27}$$

,where $I, II, \dots, N$ are the iteration steps ($N \leq K$, because the system updates the rule base in the case where the user's choice is different from the predicted winner), FP is the Feature Parameter, GV is the goodness value. After K steps, the rule base contains up to K rules. The usage of the font design system is dramatically simplified by the simple user interface and the small number of examined individuals.

4.6 Conclusion

This chapter introduced the pre-evaluation concept and demonstrated that the FRI FIVE model can be successfully applied as a pre-evaluation model in IEC systems. The new scientific results are summarized as follows.

A pre-evaluation concept is proposed for IEC and CBIDE systems, and the FIVE FRI model is successfully applied as a pre-evaluator component.

The new pre-evaluation concept supports the evaluating user by improving the evaluation comfort. The pre-evaluation mechanism offers an approximated best individual, which can either be accepted or disregarded by the user. If the pre-selected item is accepted by the user then the IEC process requires less user intervention than would be necessary for the evaluation itself. The problem of modeling the opinions and creating a model of the fitness function is a much more difficult task, as IEC applications usually involve high dimensionalities, and user feedback is very rare, covering only an infinitesimal small part of the model space. Additionally, I proposed a topology for pre-evaluated IEC systems in the VirCA environment, allowing the user to perceive pairs of examples in a multi-sensory environment in a way that appeals to human cognitive capabilities.

Related publications: C3, C5, C6, C8 , C14, C15, C16

5

STRUCTURAL IMPROVEMENTS OF THE OPENRTM-AIST ROBOT MIDDLEWARE

5.1 ABSTRACT

RT-middleware (Robot Technology Middleware) is a technology that implements several key concepts needed for the development of complex robot systems, even in geographically distributed environments. Through a useful API, reusable standardized components and communication channels, and some degree of automatization, RT-middleware helps the user to build easily reconfigurable systems. The behavior of the components and the manner of interaction among them are standardized by the Robot Technology Component (RTC) specification. Until now there have been two implementations of the specification. OpenRTM.NET is written in .NET, while the other implementation, called OpenRTM-aist – which is relied upon extensively in this dissertation – is a convenient modular system built on the Common Object Request Broker Architecture (CORBA). The first version of CORBA was released in 1991. Nowadays, CORBA is becoming increasingly outdated. The ultimate goal of this chapter, then, is to develop a new kind of robot middleware in which a modern distributed framework is applied instead of CORBA. As a first step for this substitution of the underlying framework, some practical extensions to OpenRTM-aist can be suggested, through the adaptation of the Internet Communication Engine (ICE) framework for inter-component communication, as well as through the introduction of the web application concept for system editing and control. With these structural modifications, the resulting system is shown to have grown more powerful than was the original system.

This chapter introduces some structural and implementation details of OpenRTM-aist, together with the results of the experiments conducted for performance comparison between the original and extended system.

5.2 Introduction

Within the robotics area, the task of robot systems can change quickly. If the job and environmental circumstances change frequently, reusable and reconfigurable components are needed, in addition to a framework which can handle these changes. The effort needed to develop such components depends on the programing language, the development environment and other frameworks used. Many programming languages can be applied for this process, but developing a brand new framework is a difficult task. Applying an existing framework as a base system, such as the OpenRTM-aist in the case of this dissertation, the associated development tasks can be dramatically simplified.

Frameworks and middleware are gaining popularity through their rich set of features which support the development of complex systems. Joining together any robot framework and robot drivers can form a complex and efficient system with relatively small effort. In many cases, the task of the system designer is reduced to the configuration of an already existing framework.

On the other hand, when an existing framework requires only a set of completely new features, it can simply be extended with them. Improving an existing framework is an easier job than developing a new one because the design and implementation of such a system requires specialized knowledge and skills (design and implementation patterns). In most cases, the missing functionalities can be simply embedded into the existing framework, but there are some cases when this is hard to achieve. An existing framework can be improved simply only in case it is well designed and implemented. In spite of this, building a brand new system from is a much longer process that requires much effort. In the area of robotics, there are many robot parts that share similar features, so the concept of robot middleware as a common framework for complex robot systems is obvious. Probably there is no framework which can fulfill all of the above requirements entirely.

The primary goal is to find a robot framework in this environment that is easy to use, reliable and also easy to extend. If it is impossible to find a perfect one, the secondary goal could be the improvement of an existing one. In the final case, if no acceptable framework exists, a brand new one would have to be implemented from the ground up.

In this chapter, I first compare some robot frameworks to give a starting impression, then I suggest some improvement ideas to enhance one of the existing frameworks: the very promising open source OpenRTM-aist system (it can be downloaded from [57]).

5.3 ROBOT MIDDLEWARE

In this section, I explore the requirements of robot middleware technologies in general and examine some existing robot approaches such as YARP, OpenRDK, and OpenRTM-aist. I give a comparison at the end of the section.

Frameworks and robot middleware are gaining popularity through their rich set of features, helping the development of complex systems.

Definition Robot middleware is a software middleware that extends communication middleware such as CORBA or ICE. It provides tools, libraries, APIs and guidelines to support the creation and operation of both robot components and robot systems. Robot middleware also acts as a glue that establishes a connection among robot parts in transparent way.

First of all, we should define the requirements and the actors of robot middleware. Generally, two actors use such technology: end-users and developers. Every robot middleware has to provide tools and mechanisms to facilitate the work of these actors.

Component developers design and implement components. For this development process, many programming languages and operating systems should ideally be supported. Tools are needed for the generation of skeletons and semi-working components for quicker development. The main activity of the developer actor should be the generation of semi code, the filling out of business logic, the compilation of binary code and the execution of test cases. Developers also *create robot*

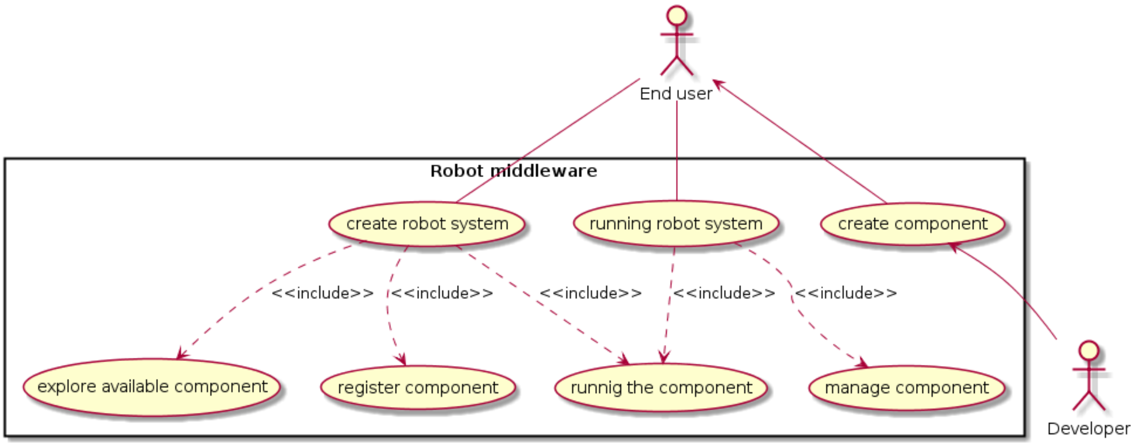


Figure 15: Use case diagram of robot middleware

parts as components using their knowledge about programming languages, robot middleware and the specification of the given robot system. Moreover, the person who develops the component has to validate it as an end-user. In summary, from the developer's point of view, the main requirements for robot middleware are as follows:

Requirement 5.3.1 Support for a variety of programming languages: *in most cases the members of research and development teams who develop various robot parts will be familiar with different languages depending on the trend at any given time.*

Requirement 5.3.2 Support for a variety of operating systems: *nowadays Windows and Linux operating system support is mandatory, as heterogeneous systems may rely on other libraries as well that are specific to these systems.*

Requirement 5.3.3 Skeleton generation and other tools for components: *The framework should be able to generate skeleton code for component development in order to speed up the development process. For this reason, robot middleware technology should include templates and tools (preferably graphical tools) to aid component development. Although developers will have to learn how to use the tools to generate source code, this requires less effort than starting component development from the scratch.*

As opposed to the developer actor, an end-user is someone who has no knowledge about programming languages, as is not interested in developing components. End-users simply want to use components for their own work. In their case only a minimum number of system parameters should be visible, which are sufficient for the creation and operation of the robot system. Nice graphical interfaces and visual aid features are useful to end-users in allowing for the straightforward operation of the robot system. End-users have two main tasks. First, they often*create complex robot systems* using existing robot parts, and second, they often want to run previously created robot systems. The creation of a robot system can include a search for available robot parts during which the user obtains information about the accessibility of available components. This accessibility information can depend the specific robot middleware which is applied for the task, and may include IP numbers, or any internal identifiers or symbolic names. Unfortunately, most robot middleware systems only support online components. For this reason, robot parts have to be available during the robot system creation process, i.e. have to be already running and must be registered.

First of all, end-users want to register their components in order to share functionalities with other end-users. After registration, the component can be found by all other participants. Usually end-users create custom robot systems using available components, so the second task is to find out which of the components are running. Once the robot systems are running, users will want to control them, i.e. activate/deactivate any or all of the components, and run them with various different parameter sets.

In summary, from the end-user point of view, the main requirements for robot middleware are as follows:

Requirement 5.3.4 *The overall change in the applications and general software components running on workstations in a research group should be minimal. In most cases the operation of any system that is relied on by end-users on a daily basis requires a special software environment. The installation of new software components may result in outage periods in terms of existing services.*

Requirement 5.3.5 User-friendly graphical interface *the graphical interface of robot middleware should be as user-friendly as possible.*

Remark Like developers, end-users would like to use Windows or Linux systems, therefore *Windows and Linux operating system support is a strong requirement*. End-users will not replace their familiar workstation environment with a new one just to accomodate the requirements of a new robot middleware.I omit from this list.

In summary, the key consideration for a robotics middleware system to be useful is that it should provide an efficient API and as much automatization as possible. Additionally, middleware systems also have to support as many operating systems and programing languages as possible. If a frameworks is applied for a specific task, some efforts will always be needed to understand the relevant concepts and application details.

One distributed environment for robot cooperation is OpenRDK. The user's robot system can be developed using a set of Agents, through a simple process. A Module is a single thread inside an agent process. Every module has a repository in which a set of internal properties are published. Inter-agent (i.e., inter-process) communication is accomplished by two methods: through property sharing and message sending. RConsole is a graphical tool for remote inspection and management of modules. It can be used as both the main control interface of the robot and for debugging while developing software. It is just an agent that happens to have some module that displays a GUI. The framework can be downloaded from [58].

Another important robot middleware platform is Yet Another Robot Platform (YARP). Communication in YARP generally follows the Observer design pattern (for more details see [59]). Every YARP connection has a specific type of carrier associated with it (e.g., TCP, UDP, MCAST (multi-cast), shared memory, within-process). Ports can be protected by SHA256 based authentication. Each port is assigned a unique name and it is registered into a name server. The YARP name server, which is a special YARP port, maintains a set of records, the keys of which are text strings (the names of the ports). The remainder of each record contains whatever information is needed to make an initial connection to a given port.

A third important robot middleware technology is OpenRTM-aist, which is a convenient modular system also built on the Common Object Request Broker Architecture (CORBA). I chose this technology as a basis for my investigations, therefore it is discussed in more detail in the following section.

Table 1 gives a comparison of the three middleware solutions mentioned here. It can be seen that all three provide similar services through similar concepts. YARP

Table 1: Main features of YARP, OpenRDK and OpenRTM-aist robot middlewares

Name of middleware	YARP	OpenRDK	OpenRTM-aist
Basic logical unit	-	Agent	Component
Basic execution unit	-	Module	Module
Information / functionality sharing	Encrypted data port	property sharing, messages	Data port, service port
Registered item	Port	Properties of Agents	Components
Communication possibilities	TCP, UDP, MCAST, shared memory, in process	TCP	TCP
Applied RMI middleware	CORBA	CORBA	CORBA
Concept	Communication centric	Data centric	Functionality or data centric

simplifies the problem of communication by storing only the ports needed for sending and receiving messages. As is required from any high-quality middleware technology, it provides developers with a flexible and expandable communication mechanism. OpenRDK uses an Agent concept and creates a repository for common data (properties or messages). OpenRTM-aist not only stores information about components, but also manages them. It supports various states during the lifecycle of components, and introduces the Execution concept, which defines patterns for specifying robot jobs more clearly (i.e., through periodically sampled data processing, and stimulus response). OpenRTM-aist also uses TCP communication (which allows for reliable, connection-oriented communication) and enables both functionality-centric and data-centric interoperation. All of these middleware technologies support online components only, therefore unavailable components cannot be used, which is a strong limitation. As a result, every time the user wants to work with the system, all processes of all robot parts have to be running. If not all parts are available, they cannot be simulated or otherwise replaced and the entire system will be unable to execute.

5.4 THE OPENRTM-AIST DETAILED

OpenRTM-aist is a robot middleware that consists of a well-written and convenient modular system built upon a CORBA basis. It is developed by the National Institute of Advanced Industrial Science and Technology - Intelligent Systems Research Institute - Task Intelligence Research Group. OpenRTM-aist components are coequal, which means all participants have the same privileges (e.g., the system editor, various robot components). It has no central logic. The end-user can search for online OpenRTM-aist components only at run-time, using a graphical system editor. The online component search is supported by the CORBA naming service, which is a simple process that acts as a servant object that can be accessed remotely by components as well as the graphical system editor.

The specification of the RT-Middleware (Robot middleware) concept is defined in the "Robotic Technology Component 1.0" and in the "Super Distributed Object 1.1" standard (for the full specification, see [60]). These standards describe the concept and the structure of a modularized robot system, and also its behavior. In OpenRTM-aist, originally introduced by Noriaki ANDO in [61] (more details about it can be found in [62, 63]), the software is modularized into components of RT functional elements (called RT-components). Each RT-Component has an interface (a "port") for communication with other components. The RT system is constructed by connecting the ports of multiple components to each other in order to aggregate RT-Component functions. The advantage of OpenRTM-aist is that it provides an simple way to create and co-operate various robot parts.

In OpenRTM-aist, many programming languages can be used for component development, including C++, Java and Python. Some tools exist to support the development process by automatically generating the skeleton of components. The developer, then, only has to fill out the generated skeleton source, by concentrating only on the business logic to be implemented.

In the case of end-users, only the address of the name server needs to be known for the framework to be configured and used. After the System Editor connects to the name server, it downloads the list of available components, from which the end-user can build his or her own robot system. The System Editor provides a graphical interface for such purposes (the editor is shown on fig. 16).

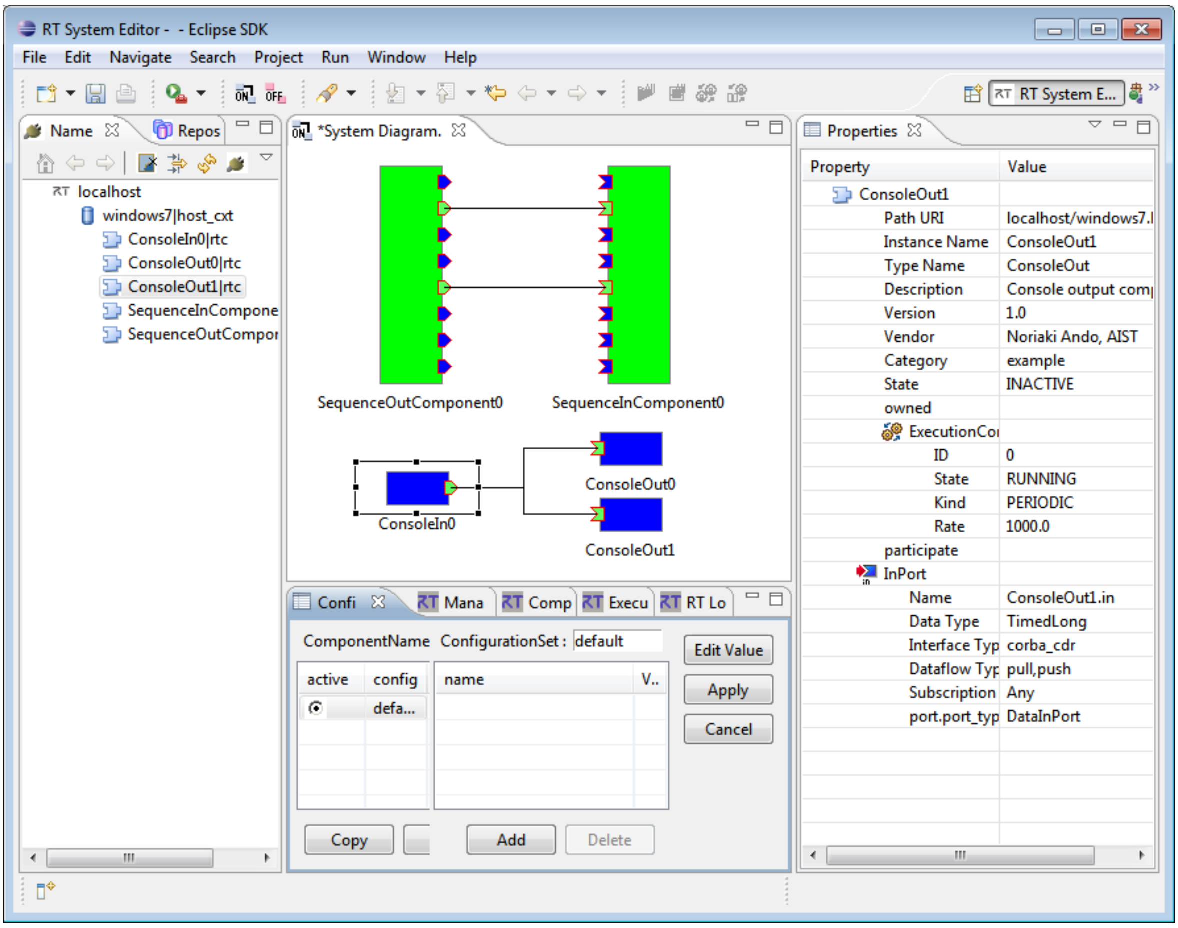


Figure 16: The main interface provided by the OpenRTM-aist system editor

For component creation, the middleware also contains a graphical editor, called RTC Builder. The definition of each RTC component is stored in a programming language independent xml file, and the source codes are automatically generated by the robot middleware framework using this xml file. The developer has only to fill out the body of each skeleton member function and to insert other external classes.

Robot component operation requires only a brief configuration entry which specifies the name server host and port, the "nickname" (for simplified reference) and path of the robot component.

In extending the features of the OpenRTM-aist, my first goal was to add support for offline components. The starting point is that the complete robot system, parts of which can be running in different places, can be explored and edited using a graphical application called RTC system editor. End-users are allowed to create and activate a new system by allocating and connecting existing online RTC components. The structure of a robotic system and its communication details are shown in the Fig. 17.

Behind the scenes, communication among allocated components can be established via the CORBA middleware, given that all components are implemented as CORBA servant objects. Online components can be controlled by the RTC graphical editor based on the naming service of CORBA. If the connection between the editor and any of the components is broken, from the editor's point of view the component is considered to be offline, and consequently the robot part (the component together with its already defined connections if there are any) is automatically removed from the user's system. This is an undesirable property of the system, as connections may drop once in a while, which should not mean that they cannot be easily re-established.

The second improvement I intended to make was to allow for client authentication together with the applied access policy. In the current system, any of the available components can be used freely by every user without any security restrictions. If the IP of the name service or the parameters of components (IP and port) are known, then anyone can use the resources of the host.

The third improvement I intended to make is to develop of a lightweight graphical system editor. The current system editor's size is about 100 MBytes, which results in a long download time. Updates of the editor must be downloaded and installed by the end-user from occasionally, which can therefore be time-consuming. The building blocks of the system, which are to be handled by the lightweight editor, are RT components (e.g. robots) with well-defined architectures, and the ability of components to be operate independently of each other. Connections (communications) between them is possible only via ports. Specifically, components can

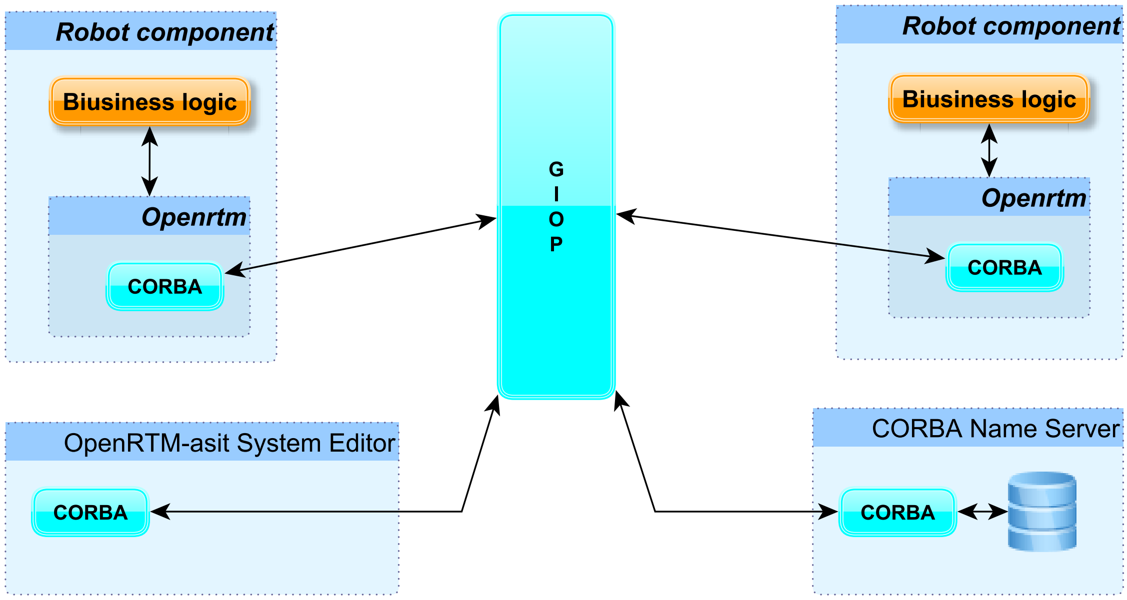


Figure 17: Structure and communication between parts of the original OpenRTM-aist robot middleware

send/receive data to/from each other via data ports. The components can live, work and terminate independently of each other. They can even have further states, depending on their design. Such additional states should be handled by the editor transparently.

5.6 DETAILS OF THE EXTENSION

In the first step of OpenRTM-aist extension, I had two main goals:

1. To provide a new communication channel for components based on ICE communication middleware, which would allow components to share their behaviors.

2. To development of a new platform-independent lightweight graphical system editor.

5.6.1 *The ICE service port extension*

One of the key underlying ideas of the improved system is the use of Internet Communications Engine (ICE) technology (more details can be found at [64]). The OpenRTM-aist system already contains a service port concept, which is the only way to publish the behavior of components among other components, but this port is based on CORBA.

A new kind of service port is therefore introduced, called ICE service port. As with the original CORBA service port, the component can publish its functionality, and it is also easy to design and to implement the business logic. The concept of ICE integration is shown in the following figure.

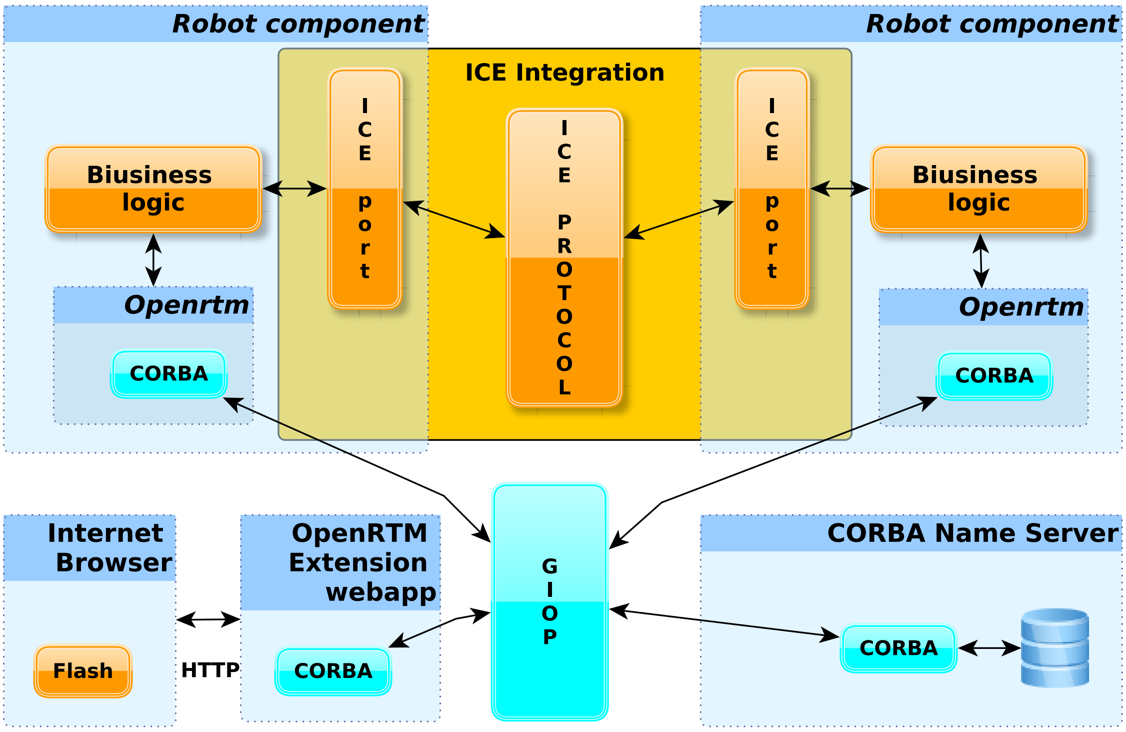


Figure 18: The concept of ICE integration into OpenRTM-aist

Like CORBA, the ICE Framework is also an object-oriented distributed platform that can run on several different operating systems and under many programming languages e.g. C++, .NET, Java, Python Objective-C, Ruby, PHP, and ActionScript, which was introduced by Henning in [65].

CORBA builds on a rather old concept of distributed application development. The first papers about it appeared starting in 1995 [66, 67]. A number of specifications have been proposed [68], and many implementations have been developed. The concept behind CORBA requires a dramatically different way of thinking about programing, and thorough knowledge of a special design pattern explained in [69].

Compared to CORBA, ICE supports .NET and many other programming languages (about 60 different ones). It is a viable candidate to be the new platform that replaces CORBA, allowing robot middleware technologies to preserve their strengths, while overcoming their weaknesses. Michi Henning gave a detailed account of the weaknesses of CORBA and the novelty of the concept behind ICE in [70, 70] ICE dramatically simplifies the complex programming model used by CORBA. The integration of the ICE system into robot middleware technology can therefore lead to a number of benefits.

The first real advantage of IcePort is the simple way in which its programming model can be understood and used (i.e. it provides a simplified programming API). Developers do not need to learn heaps of documentation, which makes the development process effortless and quick. The dynamic invocation and dispatch interfaces of ICE allow the developer to write generic clients and servers that need not have compile-time knowledge about the types used by specific applications. This opens the possibility of creating applications such as object browsers, protocol analyzers, or protocol bridges. During the specification of the IcePort extension, this novel feature of the extended system became apparent, which was not present in the original CORBA based version.

The second important feature of IcePort it makes it possible to create connections between one service consumer and several service providers. This type of communication is not supported in the original system, nevertheless it is often necessary for robot configuration, (e.g., if one robot item can be controlled by multiple input devices). In the case of ICE technology, this type of communication can be easily implemented.

The encryption of communication is a complicated task in the free implementation of CORBA, therefore an easy-to-use and easy-to-configure encryption layer is also desirable. Nowadays it is becoming increasingly important to have secure connections between robot parts, but the CORBA based OpenRTM-aist does not support this. ICE, on the other hand, provides an encrypted data stream between multiple ports, which can be easily introduced on the end-user's side, thus solving the problem of secure communication with minimal effort. The communication stream is usually TCP based, but in some situations the use of UDP datagrams can also have some benefits.

For instance, the UDP protocol can be used for one-way operation, which is a faster solution than to use TCP. UDP can be well-suited, for example, to broadcasting information in a publish-subscribe kind of application or during logging activity. The overhead in negotiating for a TCP socket and handshaking the TCP packets is huge, hence UDP communication is at least two times faster than TCP in uni-directional communication. UDP is supported by the ICE API. The components can be organized into farms, so the system can be structured to be more reliable.

5.6.2 *Details of web application injection*

The main components of a robot system usually have a rich graphical interface. E.g. in the VirCA system (which will be discussed more detailed in the next session), the main component has an interface in three-dimensional virtual space. It is a time consuming process to switch between a separate editor and a main graphical interface. It would be much better if the editor surface could be displayed within the main graphical interface.

The new editor is a lightweight System Editor that is enabled to work within a browser (see in Fig 19).

The business logic is implemented as a web application. The actual robot system can be saved for further editing on the client side in xml format, or on the server side in a database. Our editor can also use favorite components which are stored in a searchable database. As discussed in the introductory section, if the network connection is broken in OpenRTM-aist, then the box of the effected component disappears and all the related connections are deleted as well. In the proposed extension, the editor system simply displays that the state of the component is "not

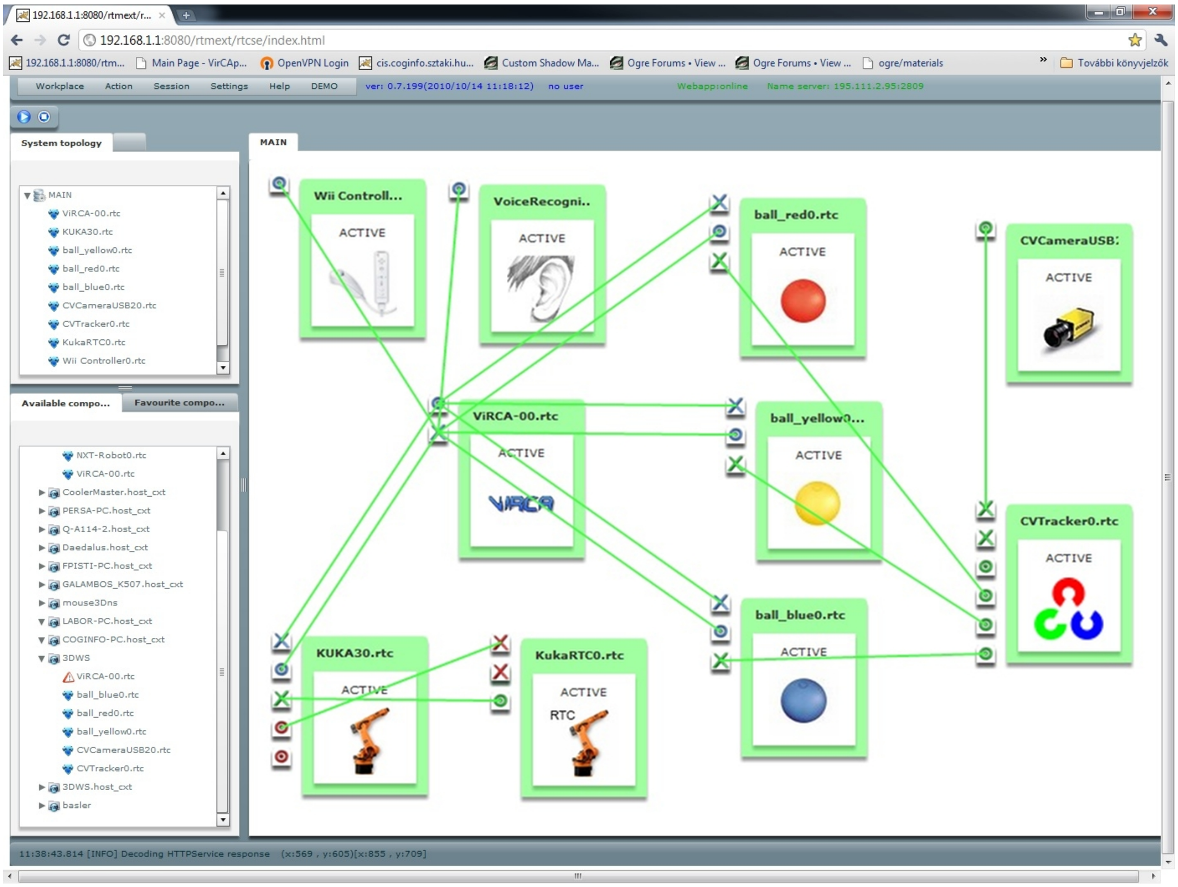


Figure 19: The interface of the new System Editor

available", but the user can still work with the system as a whole. This is one of the key improvements of the extension. During design time, all components and connections of the robot system are grayed out, meaning that the component statuses are unknown. At any given time, the system displays the last known state of each component, which means that once a connection is dropped, the associated components will remain grayed out irrespective of whether it is available or not, until the system is refreshed. When the robot system is once again activated by the end-user, the extension logic tries to activate the appropriate components and to re-establish the connections.

The new editor provides the virtual master component concept in the form of special components. The virtual item allows the division of the system into several smaller parts ('divide et impera'). Using this extension, the editing of parts can be simpler as the unnecessary and distracting details can be hidden from the user. The virtual item can include other virtual or regular components as well. The activation command in this case means the activation of all the included real items. Any ports can be published as new ports of a virtual component.

The original editor is an Eclipse plug-in, therefore in order to run the original system editor, the user must have an Eclipse environment installed (about 100 Mbyte), and a high Eclipse dependency with limited graphical opportunities. In our lightweight system editor, end-users need only a browser to use the system, as the editing system is a web application. At the beginning of each service usage, the browser always checks the URL, thus the last version will be accessed automatically. In this way, version control is also guaranteed.

In the new system, the appearance of the Web layer in the architecture also makes way for the simpler development of further extensions, such as advanced user management, integration into other systems, sup-port legacy system, searching the components, advertisement, etc. To do this, only a minimum investment is needed.

In the case of the current implementation, there is an XML-based communication between the host and the end-user over the HTTP protocol. The applied Web application supports all operating systems with any browser, therefore the development of heterogeneous systems is also supported. For an end-user who does not want to run his or her own robot components, there is no need to use the CORBA protocol or to install any CORBA endpoint, because the lower level of OpenRTM-aist is isolated by the web application tier. The extended robot system has several parts as shown in Fig. 20.

In the new OpenRTM-aist architecture, some new items appear. The first is the *client tier*. At the end-user side, a flash application runs in a browser in order to present an interface to the user. It sends requests to the presentation layer and receives corresponding responses. The basic editor function is the handler here (e.g., system editor area, shape for component, component insertion). To support offline components and minimize communication overheads, the required connections are created just in time as the robot system is starting.

The second tier of the system is the *presentation tier*, which bridges between the "client tier" and the "business logic". After the presentation tier receives the commands and parameters from the client tier, the values provided are checked, and only valid commands are forwarded to the business logic. For this task, the tier can

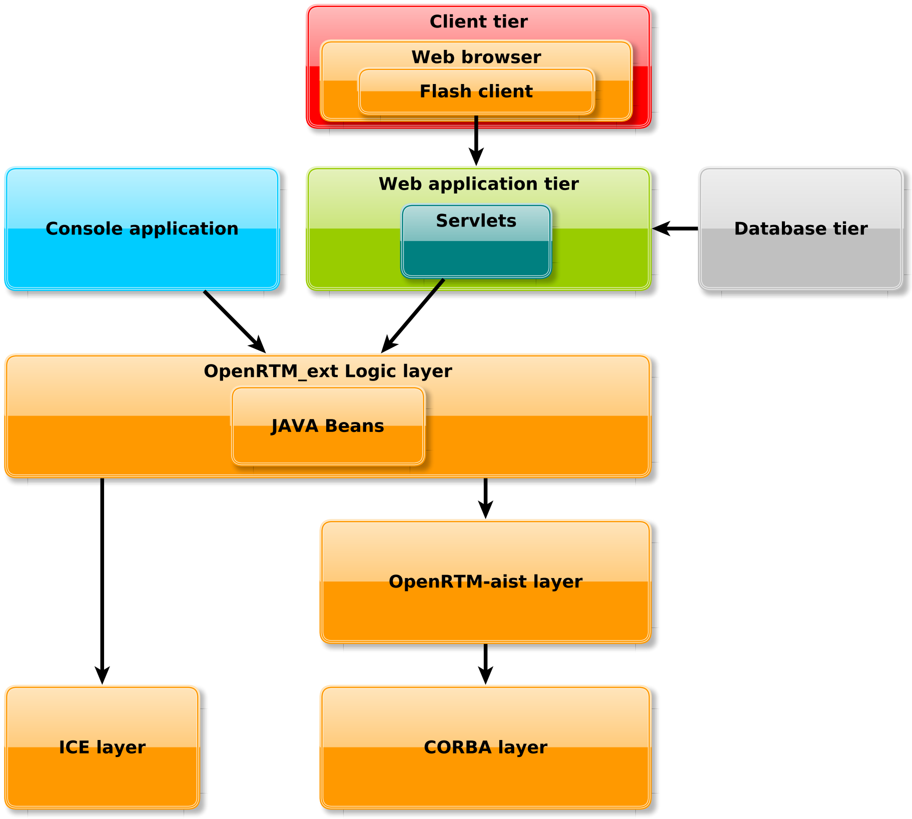


Figure 20: Tiers and layers of the extended OpenRTM-aist architecture.

use a database and/or a common logic layer. This tier is also independent from OpenRTM-aist. Two varieties of this tier exist: console-based and web-based. For automatic execution (and the possibility to perform integration tests), the developed editor system has both a command line interface called OpenRTMExt_Tools (a Java application) and a graphical interface (Java web application). The server logic is implemented as a set of jsp pages and servlets that serve requests. In the case of console applications, commands are sent using simple text (generated by java application), but in the case of graphical applications, communication is XML based. The Web application and the console system editor layers belong to the presentation tier running on server. They translate the requests using server logic. The results are simple text or XML documents sent to the client tier or printed out to the console window.

The *business logic tier*, called OpenrtmExt_logic is used by the presentation tier. This tier coordinates the application collections of Java Beans used in the system. The logic is run in threads for better performance. The master component concept also appears here. The concept is present in the form of a platform-independent developer library (lib) which provides virtual elements which simplify the topology of the user's robot system. The business tier depends on lower level communication implemented by e.g. CORBA and ICE. This layer hides the details of OpenRTM-aist, introduces new features and establishes new complex instructions which are not available in the original system. The request, which is obtained from a client is translated to CORBA method invocation and forwarded to the necessary participant. After the appropriate remote object(s) has performed its work (any value can be sent back), the result is returned to the presentation tier. The implementation of this tier is such that it guarantees the same behavior in the case of the console as well as the graphical interface.

The *database tier* stores the profiles of users and the details of the robot systems. Applying a stored procedure mechanism, some functionalities can be moved to the database tier. The tier guarantees a quick and searchable data background for the system.

5.7 AN APPLICATION EXAMPLE, THE VIRCA SYSTEM

The Virtual Collaboration Arena (VirCA) is a virtual laboratory that provides teams of researchers or industrial engineers the opportunity to collaborate with each other. Using VirCA, it is possible to control virtual and physical devices remotely through an augmented environment. The VirCA system was developed at the Cognitive Informatics Research Group of the MTA SZTAKI research institute. More information about it can be found at [71] and [72]. The main idea behind VirCA is to place physical devices into a generated virtual space, as shown in Fig. 21. The detailed description about the system and its usage can be found in [45, 73, 74].

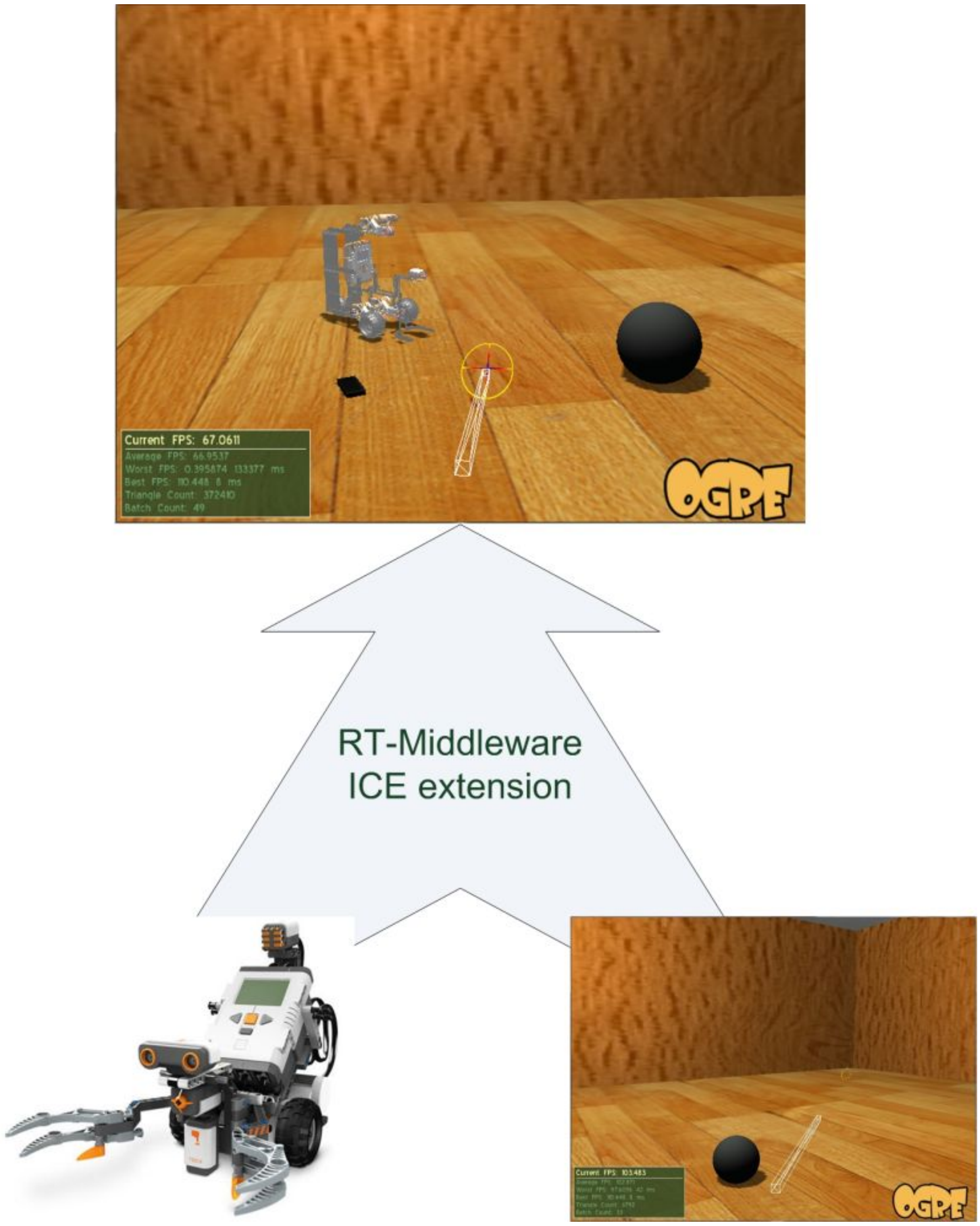


Figure 21: The concept of VirCA system.

Behind the scenes, this system is based on the extended OpenRTM-aist, that is it makes use of extended OpenRTM-aist components. Special components run in all locations, connecting to a simple CORBA name server to register their device lists. Once the components are up and registered on the network, the operators can choose their input and output devices (cognitive informatics equipment and robots) and use them to solve the given task. The main task of VirCA is to visualize physical devices, to manage the communication between them, and to allow users and components to control them, both in the virtual and in the physical world. As a secondary goal, it is also able to connect more than a single virtual space, to form an extended virtual reality environment. By today, VirCA has been established as an extended OpenRTM-aist system. Some cognitive components have also been also developed for the system. The members of the development team are engineers, not programmers, and it is therefore useful that they are not required to program CORBA code.

The VirCA system requires the operation of the following mandatory components: A *3D Space*, that is a 3D visualization component which continuously renders the virtual world. After the user connects (using a graphical interface) any device to this main component, the 3D space is required to display a model of the physical device in the appropriate position of the virtual space (as long as a physical manifestation of the device is specified). The virtual object can also "sense" the presence of other objects (i.e., virtual and physical objects can react to each other). The user can send command to any physical device via the graphical (3D) interface.

The 3D Space component has two interfaces: the register interface and the commander interface. The register interface is an ICE service provider. Physical (cyber) devices can register/unregister themselves into the virtual world, or share their updated positions with it. The commander interface is an ICE service consumer. It sends commands to the physical entity, e.g. for it to move to a specified position, for it to grasp or pick up something, or perform any other pre-defined task, etc. The 3D space component has two kinds of implementations. One can run on a normal personal computer as a usual graphical desktop application. The other is capable of running in a CAVE immersive 3D environment, which is more realistic because the user can stand inside a cube of projected 3D screens, and have the feeling of being a part of the surrounding virtual space.

Another key concept in VirCA is the concept of cyber devices. Several different physical robot devices can exist at the same time in different places. Each type of robot part can have its own component, which can also exist in several instances at the same time. A number of devices are already supported: e.g., Lego NXT, a KUKA industrial robot and a NAO humanoid robot. Cyber device components must have two mandatory interfaces: a commander interface, which is an ICE service provider, and a register interface, which is an ICE service consumer. The commander interface receives commands from 3D main components or from any other component that wants to control the cyber device. The register interface can be used to register into, or un-register from the virtual space. There are some optional components as well. The Camera component provides still or moving images of a pre-determined physical space. The Observer component is capable of receiving such images, identifying objects in them and sending events to the 3D engine, which in turn notifies other components that may be interested in the events. A variety of Controller component forms can exist. The user can for instance send commands to cyber devices via such components using voice or hand-gestures. The input of these components can be provided by camera or microphone or any other cognitive sources. The system has not yet been extended to handle cases where the system uses more cyber devices than can be accommodated by a given commander interface, or if the 3D system consumes results from more than a single commander interface from several cyber devices.

5.8 EXPERIMENT

Having developed a new and improved system with some novel functionalities, a key question I intend to address in this section is the question of system performance. The main goal of the test scenario described here is to compare the performance of CORBA and ICE service ports. I implemented two test components: a test provider and a test consumer. The provider has an IcePort and a CorbaPort. These two ports have the same interface (IDL in case of CORBA and slice in case of ICE) and the implementations are also similar. Similarity in this case means that the same number of algorithmic steps are executed, but of course the instructions are dependent on the specific technology. The consumer has two ports as well, to consume services during the test. After creating the two component instances, the appropriate ports were connected. The test cases were run sequentially step by step under Windows and Linux. The amount of time it took for a client to invoke the test operation in a server and to receive the results of the operation was measured. The latency and throughput of the new ICE service port was tested using the demos of the ICE distribution. In the latency test, the operation had no in-parameters, so no data was sent with the invocation. Similarly, because there were no out-parameters (a void value was returned), no data was returned by the operation. Throughput determines how much data the middleware can move per unit of time. Throughput heavily depends on the type of data that is sent. Byte sequences were selected for transmission, as they can be regarded as the most efficient data types, which do not require further processing. 50 Megabytes of data were sent from consumer to provider in 100 Kbyte-steps.

The test function of the ICE system was used for the comparison of the two service ports. Both ICE and CORBA (omniORB in case of C++, and jacORB in case of Java) were run with no special configuration, thus the default options were used. The time required for 100,000 function calls was measured. In case of the loop-back device test, CORBA was 1.5 times faster than ICE. In real-world applications, endpoints are on different machines, therefore this test result is insignificant.

The results of the test for different hosts are shown on Fig. 22. According to the results (see Fig. 22), the performance of the two technologies is almost the same. Using ICE ports, the features supported by the API are significantly better, and yet the performance is nearly the same as that of CORBA. The two most common programming languages (JAVA and C++) were tested, with the same results.

In spite of the similar performance measures, the usee of *the ICE service port is strongly recommended instead of the CORBA port*, as it has a better programming interface and its capabilities for customization are also much more flexible. The performance of ICE in case of different programming languages and operating systems is also different. The best performance was measured in case of C++ and the Linux operating system. The worst results were produced by the Windows and Java combination. The results are summarized on Fig. 23. According to the opinions of the developers, the minor differences in performance are due to implementation issues of the ICE network layer and the Java virtual machine.

5.9 CONCLUSION

In this chapter, improvement ideas for the existing OpenRTM-aist framework were discussed, and relevant implementation issues were presented and discussed. The

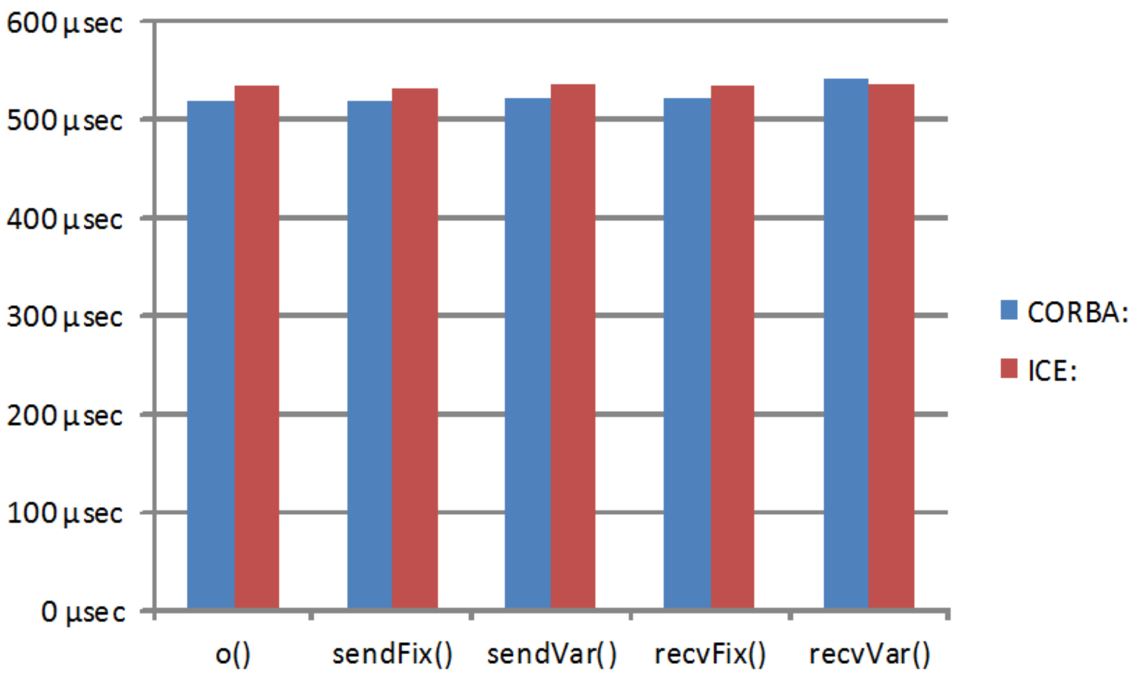


Figure 22: Performance comparison of Iceport and CORBA port performance in case of C++

new scientific contribution introduced in this chapter can be summarized as follows.

I have introduced an extension to the OpenRTM-aist robot middleware framework that allows for CORBA-free operation.

The improvement is accomplished through the use of two existing technologies, i.e., through the integration of the ICE communication framework with OpenRTM-aist, and through the injection of a web application tier into CORBA based robot systems. The introduced web tier has resulted in a new lightweight web browser based system editor. Through the integration of ICE, new functionalities have been introduced to the system, e.g., authentication, authorization, support for offline components, save/load functionalities for robot systems and encrypted communication channels between end-users and the web application. The end-user client of the new editor is a simple web browser which does not require the installation of any special application, such as the CORBA library or Eclipse (both were required for the original system editor). The new editor is built upon the infrastructure of web applications, therefore it supports component sharing in a trusted and controlled way.

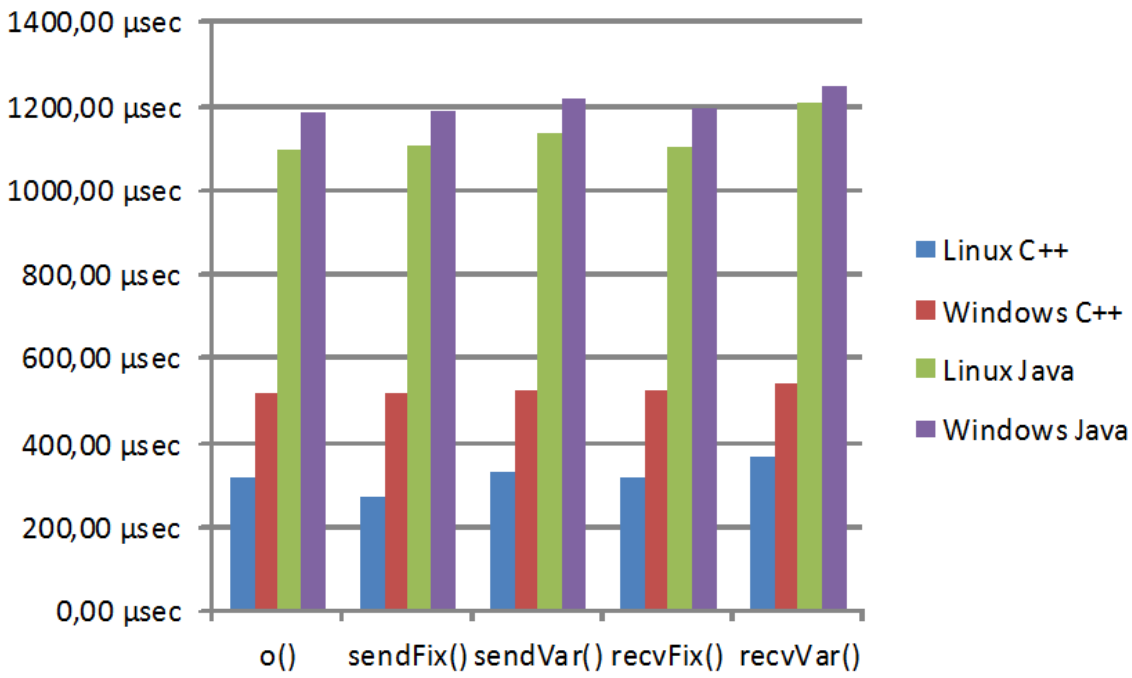


Figure 23: Performance comparison of Iceport in case of different platform

ICE integration was carried out by extending the existing OpenRTM-aist robot middleware without its modification. A new ICE based service port was introduced, which has the same interface as CORBA-based ports in order to simplify the developer's work. Programmers need to invest less effort when using the new, ICE-based port because of the simple programming model used by the ICE communications framework. New features supported by the extended system are the follows: encrypted communication between ICE service ports, one-way communication between service ports, and single-consumer multi-provider topology. The extension was easily added to the unmodified version of the original robot middleware, allowing for original features to be left unchanged for backwards compatibility.

Related publications: C2, C9,C10, C11, C12, C13, C17, C18

6

CONCLUSION

6.1 APPLICABILITY OF RESULTS

The scientific results presented in this dissertation are related to my research at the Department of Information Science of the University of Miskolc. Based on the extension of OpenRTM-aist, the VirCA application was released in 2010 by the 3D Internet-based Control and Communications Research Laboratory of the Institute for Computer Science and Control at the Hungarian Academy of Sciences. The 3D system is under continuous development to this day. The web tier which I injected into OpenRTM-aist, and the new System Editor which I have created based on the web tier is used in VirCA to this day. Programming using ICE ports is also possible through the API of VirCA.

Based on the VirCA system, two large-scale projects, the HUNOROB (HUngarian-NOrwegian research based innovation for development of new, environmental friendly, competitive ROBot technology for selected target groups) project, and the NAP (KCKHA005, OMFB-01137 / 2008) project were successfully completed.

6.2 THE NEW SCIENTIFIC RESULTS OF THE THESIS

THESIS 1 *The "Double Fuzzy Point Methodology" (GDFPM) is introduced, which can be applied as a guideline for the adaptation of the double fuzzy point representation in any two-step FRI method.*

GDFPM replaces the single-rule reasoning step of the original two-step method with a new "fuzziness similarity ratio preservation reasoning" step. Two solutions are proposed *for the "fuzziness similarity ratio preservation reasoning" step: one which is based on the "Least Squares Method" ("LESFRI"), and another which applies the "Polar a-cut" interpolation ("FRIPOC") concept. The guideline formulated based on these results suggests an extension that is based on the Fuzzy Rule Pair which defines the valid domain. The common methodology has twice as much space complexity as the original method, however, time complexity does not change significantly. Compared to the original two-step FRI method, the first step of the proposed GDFPM approach consists of the generation of a temporal interpolated double fuzzy point rule (this is a pair of rules: one for each of the fuzzy rule sets R^p and R^q) in the position of the observation. The second step of the proposed GDFPM approach consists of the determination of the conclusion based on the observation (A^*) and the temporal interpolated double fuzzy point rule ($A_i^{p,q} \longrightarrow B_i^{p,q}$).*

Related publications: C1, C4, C7

THESIS 2 *A pre-evaluation concept is proposed for IEC and CBIDE systems, and the FIVE FRI model is successfully applied as a pre-evaluator component.*

The new pre-evaluation concept supports the evaluating user by improving the evaluation comfort. The pre-evaluation mechanism offers an approximated best individual, which can either be accepted or disregarded by the user. If the pre-selected item is accepted by the user then the IEC process requires less user intervention than would be necessary for the evaluation itself. The problem of modeling the opinions and creating a model of the fitness function is a much more difficult task, as IEC applications usually involve high dimensionalities, and user feedback is very rare, covering only an infinitesimal small part of the model space. Additionally, I proposed a topology for pre-evaluated IEC systems in the VirCA environment, allowing the user to perceive pairs of examples in a multi-sensory environment in a way that appeals to human cognitive capabilities.

Related publications: C3, C5, C6, C8 , C14, C15, C16

THESIS 3 *I have introduced an extension to the OpenRTM-aist robot middleware framework that allows for CORBA-free operation.*

The improvement is accomplished through the use of two existing technologies, i.e., through the integration of the ICE communication framework with OpenRTM-aist, and *through the injection of a web application tier into CORBA based robot systems. The introduced web tier has resulted in a new lightweight web browser based system editor. Through the integration of ICE, new functionalities have been introduced to the system, e.g., authentication, authorization, support for offline components, save/load functionalities for robot systems and encrypted communication channels between end-users and the web application. The end-user client of the new editor is a simple web browser which does not require the installation of any special application, such as the CORBA library or Eclipse (both were required for the original system editor). The new editor is built upon the infrastructure of web applications, therefore it supports component sharing in a trusted and controlled way.*

ICE integration was carried out by extending the existing OpenRTM-aist robot middleware without its modification. A new ICE based service port was introduced, which has the same interface as CORBA-based ports in order to simplify the developer's work. Programmers need to invest less effort when using the new, ICE-based port because of the simple programming model used by the ICE communications framework. New features supported by the extended system are the follows: encrypted communication between ICE service ports, one-way communication between service ports, and single-consumer multi-provider topology. The extension was easily added to the unmodified version of the original robot middleware, allowing for original features to be left unchanged for backwards compatibility.

Related publications: C2, C9,C10, C11, C12, C13, C17, C18

6.3 FUTURE RESEARCH DIRECTIONS

All topics described in this dissertation under continued and active research.

A universal benchmark system is still missing from the area of FRI method. A large number of FRI methods are available which are still poorly documented. The use of any FRI method requires knowledge about FRI concepts, therefore, the widespread applicability of these methods is still limited. It should be possible in the future to use benchmark systems to analyze specific problems in terms of their tractability using available FRI methods. The user should be able to choose the appropriate FRI method based on the results of such a benchmark system.

In the area of robot middleware, new features are still required by end-users and component developers as well. Ontology-based robot component classification would make the comparison of components, as well as the checking of compatibility between components and the finding/discovering of appropriate components for a given task easier. Robot system could be extended with a new robot component broker (as a special resource broker), which could suggest components that are currently available instead of ones that are unavailable or inaccessible. In a fully developed system, complex functionalities could be assembled automatically based on appropriate component groups. All of this could be achieved based on an ontology and the ability to manage offline components in the extension I have proposed and developed.

7

AUTHOR'S PUBLICATIONS

Reviewed Publications in Foreign-Language Published in International Journal

C1 Z. Krizsán, Sz. Kovács. Double Fuzzy Point Extension of the Two-step Fuzzy Rule Interpolation Methods. *Acta Polytechnica Hungarica*, volume 10, number 5 pp. 175-190, 2013. (ISSN 1785-8860) **If: 0.59**

C2 Z. Krizsán, Sz. Kovács. Structural Improvements of the OpenRTM-aist Robot Middleware. *Applied Computational Intelligence in Engineering and Information Technology, Springer-Verlag*, volume 1, pp. 287-300, 2012. (ISSN 978-3-642-28305-5, ISBN 978-3-642-28304-8)

C3 Z. Krizsán, Sz. Kovács. Gradient Based Consequent Optimisation of a FRI Rule Base. *In Production Systems and Information Engineering 2008*, volume 5, pp. 177-188, 2009. (HU ISSN 1785 1270)

C4 Z. Krizsán. Complexity Examination of SURE-P, SURE-LS and REVE Single Rule Reasoning Methods. *In Annals of the Faculty of Engineering Hunedoara*, volume 3, pp. 195-200, 2007.(ISSN 1584–2673)

Reviewed Publications in Hungarian language Published in Local Journal

C5 Z. Krizsán, Sz. Kovács. Közös fejlesztői keretrendszer fuzzy szabály interpolációs módszerekhez. *A gépipari tudományos egyesület műszaki folyóirata (GÉP)*, volume LXIII, number 5, pp 75-78, 2012. (ISSN 0016 8572)

Reviewed Publications in Foreign-language Published in International Conference Proceedings

C6 Z. Krizsán, Sz. Kovács. Fuzzy Model based Differential IEC for Human-System Interaction in the VirCA Environment. *3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2012)*, IEEE, pp. 169-173, 2012

C7 Z. Krizsán, Sz. kovács. Double Fuzzy Dot Extension of the FRIPOC Fuzzy Rule Interpolation Method. *4th IEEE International Symposium on Logistics and Industrial Informatics (LINDI 2012)*, IEEE, pp. 191-196, 2012
Cited by:

a) Z.C Johanyák: New Initial Fuzzy System Generation Features in the SFMI Toolbox, 5th IEEE International Symposium on Logistics and Industrial Informatics (LINDI 2013), Wildau, Germany, September 5-7,pp. 29-34, 2013

C8 Z. Krizsán, Sz. Kovács. Fuzzy Rule Interpolation Developer Toolbox Library. *IEEE 7th International Symposium on Applied Computational Intelligence and Informatics (SACI 2012)*, IEEE, pp. 119-123, 2012

C9 Z. Krizsán, Sz. Kovács. Name Service Redundancy in Robot Technology Middleware. *In 10th IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics (SAMI 2012)*, IEEE, pp. 219-222, 2012

C10 Z. Krizsán, Sz. Kovács. Extending the RT-Middleware Framework with ICE Service port. *In 2nd International Conference on Cognitive Infocommunications (CogInfoCom2011)*, pp. 1-5, 2011

C11 Z. Krizsán, Sz. Kovács. The VirCa System Editor in 3D space. *In 2nd International Conference on Cognitive Infocommunications (CogInfoCom2011)*, pp. 1-3, 2011

C12 Z. Krizsán, Sz. Kovács. Some Structural Improvements of the OpenRTM Robot Middleware. *In 12th IEEE International Symposium on Computational*

Intelligence and Informatics (CINTI 2011), IEEE, pp. 345-350, 2011

Cited by:

a) Ma, Q., Zou, Y., & Zhang, T. Study of service robot architecture based on middleware and abstract environment. In Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on (pp. 1200-1205). IEEE, (2012, December)

C13 Z. Krizsán. ICE Extension of RT-Middleware Framework. *In 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (CINTI 2009)*, IEEE, pp. 513-521, 2009

Cited by:

a) Galambos, P., & Baranyi, P. Virca as virtual intelligent space for rt-middleware. In Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on (pp. 140-145). IEEE. (2011, July)

C14 Z. Krizsán, Sz. Kovács. FRI Model based Pre-evaluation in IEC. *In 5th International Symposium on Applied Computational Intelligence and Informatics (SACI 2009)*, pp. 35-40, 2009

Cited by:

a) Lorencik, D., Vascak, J., & Vircikova, M. Adaptive Fuzzy Cognitive Maps Using Interactive Evolution: A Robust Solution for Navigation of Robots. In Robot Intelligence Technology and Applications 2012 (pp. 703-711). Springer Berlin Heidelberg. 2012

C15 Z. Krizsán. Uniform Complexity of FEAT-p and FEAT-LS Fuzzy Set Interpolation Methods. *In MicroCad 2008*, pp. 83-88, 2008

C16 Z. Krizsán, Sz. Kovács. Gradient Based Parameter Optimisation of FRI "FIVE". *In 9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (CINTI 2008)*, pp. 531-538, 2008

Non-reviewed Publications in Foreign-language Published in International Conference Proceedings

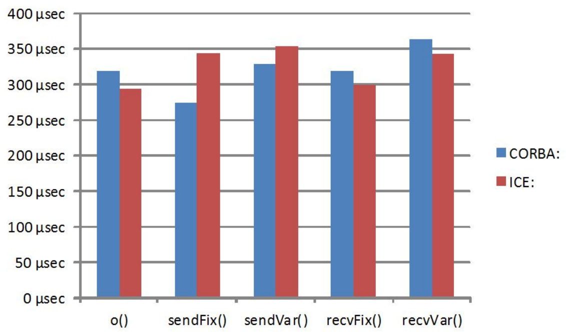
C17 Z. Krizsán, Sz. Kovács. Structural Improvements of the OpenRTM Robot Middleware. *In Doktoranduszok Fóruma 2011*, pp. 72-78, 2011

C18 Z. Krizsán, P. Galambos. VirCa System Editor. *In HUNOROB Closing conference 2011*, pp. 1-4, 2011

A

OPENRTM-AIST EXTENSION APPENDIX

A.1 PERFORMANCE OF THE ICEPORT VERSUS THE CORBA PORT UNDER C++ AND LINUX



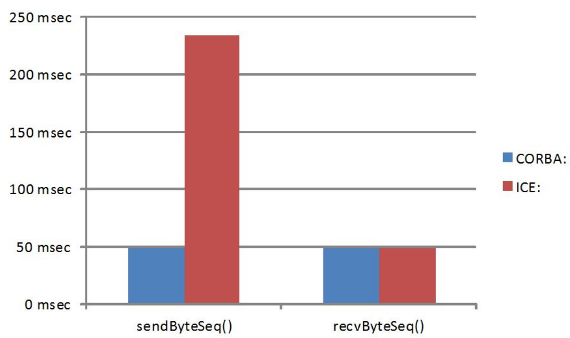
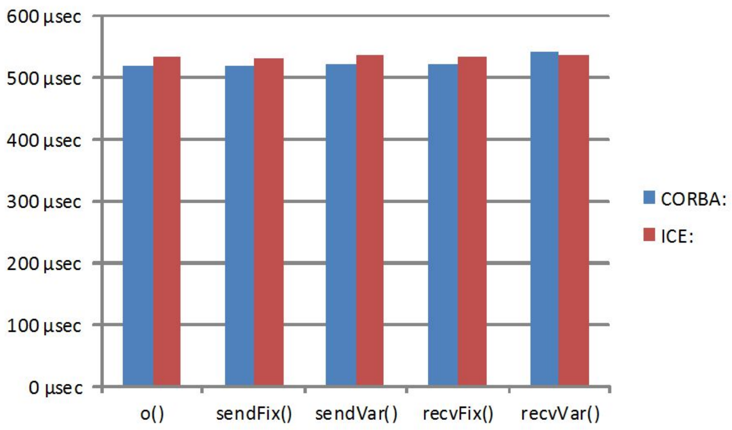


Figure 24: Measurement results on the use of the IcePort versus the CORBA port under C++ and Linux

A.2 Performance of the IcePort versus the CORBA port under C++ and Windows



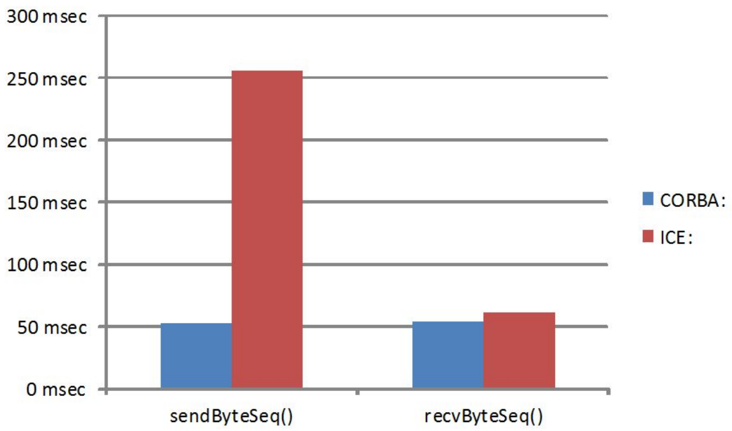


Figure 25: Measurement results on the use of the IcePort versus the CORBA port under C++ and Windows

A.3 PERFORMANCE OF THE ICEPORT VERSUS THE CORBA PORT UNDER JAVA AND LINUX

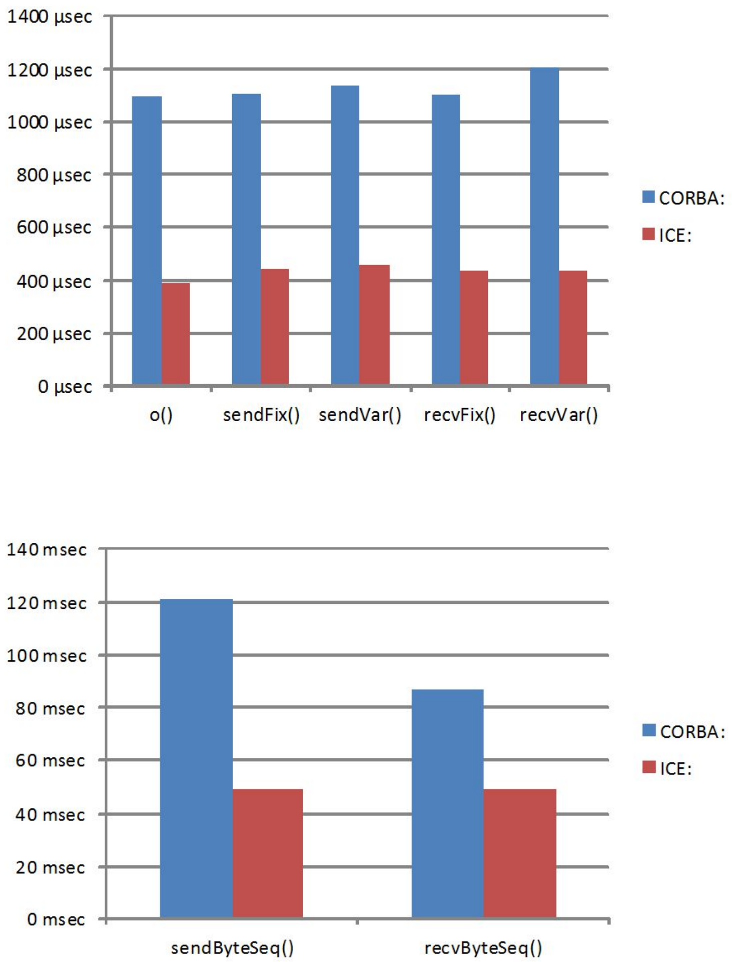
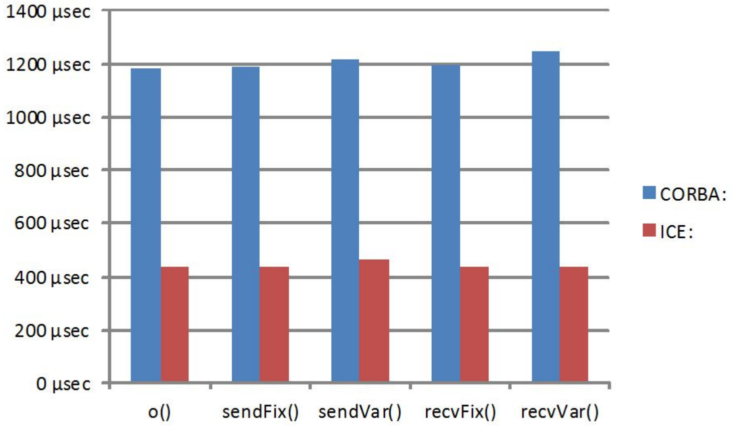


Figure 26: Measurement results on the use of the IcePort versus the CORBA port under Java and Linux

A.4 Performance of the IcePort versus the CORBA port under Java and Windows



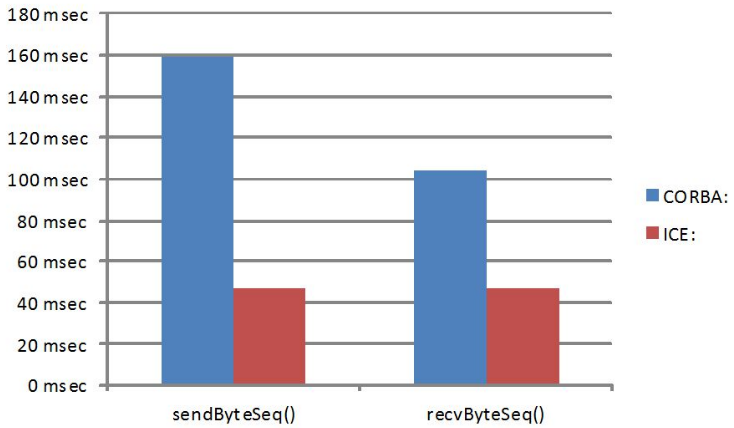


Figure 27: Measurement results on the use of the IcePort versus the CORBA port under Java and Windows

A.5 Performance of the IcePort versus the CORBA port under Java and Linux

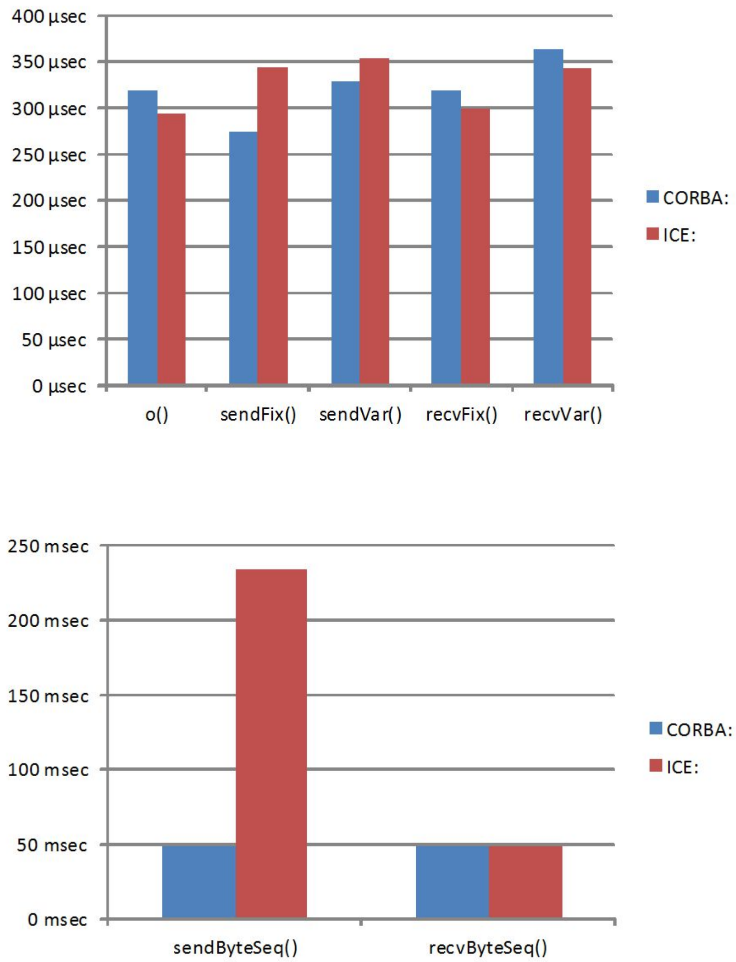


Figure 28: Measurement results on the use of the IcePort versus the CORBA port under Java and Linux

BIBLIOGRAPHY

[1] Zsolt Csaba Johanyák and Szilveszter Kovács. Fuzzy rule interpolation by the least squares method. In *7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI 2006)*, pages 495–506, 2006.

[2] Zsolt Csaba Johanyák and Szilveszter Kovács. Fuzzy rule interpolation based on polar cuts. In *Computational Intelligence, Theory and Applications*, pages 499–511. Springer, 2006.

[3] Zhiheng Huang and Qiang Shen. Fuzzy interpolative reasoning via scale and move transformations. *Fuzzy Systems, IEEE Transactions on*, 14(2):340–359, 2006.

[4] Irina Perfilieva, Didier Dubois, Henri Prade, Francesc Esteva, Lluis Godo, and Petra Hod'áková. Interpolation of fuzzy data: Analytical approach and overview. *Fuzzy Sets and Systems*, 192:134–158, 2012.

[5] S Jenei. Interpolation and extrapolation of fuzzy quantities revisited–an axiomatic approach. *Soft Computing*, 5(3):179–193, 2001.

[6] Domonkos Tikk, Z Csaba Johanyák, Szilveszter Kovács, and Kok Wai Wong. Fuzzy rule interpolation and extrapolation techniques: Criteria and evaluation guidelines. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 15(3):254–263, 2011.

[7] L. T. Kóczy, K. Hirota, and T. D. Gedeon. Fuzzy rule interpolation by the conservation of relative fuzziness. Technical Report 97/2, Hirota Lab, Dept. of Comp. Intelligent and Sys. Sci., Tokyo Institute of Technology, Yokohama, Japan, 1997.

[8] Sz. Kovács. Extending the concept of fuzzy rule interpolation with the interpolation of the fuzziness. *Proceedings of the WCCI 2012 IEEE World Congress on Computational Intelligence*, pages 1106–1113, June 2012.

[9] L. T. Kóczy and K. Hirota. Rule interpolation by α-level sets in fuzzy approximate reasoning. *BUSEFAL*, 46(Automne):115–123, 1991.

[10] Laszlo T Koczy and Kaoru Hirota. Size reduction by interpolation in fuzzy rule bases. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 27(1):14–25, 1997.

[11] Peter Baranyi, Domonkos Tikk, Yeung Yam, Laszlo T Koczy, and Laszlo Nadai. A new method for avoiding abnormal conclusion for α-cut based rule interpolation. In *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE'99. 1999 IEEE International*, volume 1, pages 383–388. IEEE, 1999.

[12] Domonkos Tikk, Tamás D Gedeon, and Péter Baranyi. A fuzzy interpolation algorithm closed over cnf sets. *International Journal of Fuzzy Systems*, 4(1):634–638, 2002.

[13] Yeung Yam. Eliminating the abnormality problem of α-cut based fuzzy interpolation. In *Proceedings of the 8th International Fuzzy Systems Association World Congress*, pages 762–766, 1999.

[14] Kok Wai Wong, Domonkos Tikk, Tamás D Gedeon, and László T Kóczy. Fuzzy rule interpolation for multidimensional input spaces with applications: A case study. *Fuzzy Systems, IEEE Transactions on*, 13(6):809–819, 2005.

[15] B Bouchon-Meunier, J Delechamp, C Marsala, N Mellouli, M Rifqi, and L Zerrouki. Analogy and fuzzy interpolation in case of sparse rules. In *Proc. of the EUROFUSE-SIC joint conf*, pages 132–136, 1999.

[16] B Bouchon-Meunier, C Marsala, and M Rifqi. Interpolative reasoning based on graduality. In *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on*, volume 1, pages 483–487. IEEE, 2000.

[17] Péter Baranyi, László T Kóczy, and Tamás D Gedeon. A generalized concept for fuzzy rule interpolation. *Fuzzy Systems, IEEE Transactions on*, 12(6):820–837, 2004.

[18] Z Shen, L Ding, and M Mukaidono. Methods of revision principle. *Proc. 5th IFSA World Congr*, pages 246–249, 1993.

[19] P Baranyi and LT Kóczy. A general and specialised solid cutting method for fuzzy rule interpolation. *J. BUSEFAL, URA-CNRS*, 66:13–22, 1996.

[20] Péter Baranyi, Sándor Mizik, László T Kóczy, Tamás D Gedeon, and István Nagy. Fuzzy rule base interpolation based on semantic revision. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 2, pages 1306–1311. IEEE, 1998.

[21] ZC Johanyák. Performance improvement of the fuzzy rule interpolation method lesfri. In *Computational Intelligence and Informatics (CINTI), 2011 IEEE 12th International Symposium on*, pages 271–276. IEEE, 2011.

[22] ZC Johanyák. Fuzzy rule interpolation based on subsethood values. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 2387–2393. IEEE, 2010.

[23] Zs Cs Johanyák, Domonkos Tikk, Szilveszter Kovács, and Kok Wai Wong. Fuzzy rule interpolation matlab toolbox-fri toolbox. In *Fuzzy Systems, 2006 IEEE International Conference on*, pages 351–357. IEEE, 2006.

[24] ZC Johanyák. Sparse fuzzy model identification matlab toolox-rulemaker toolbox. In *Computational Cybernetics, 2008. ICCC 2008. IEEE International Conference on*, pages 69–74. IEEE, 2008.

[25] Zsolt Johanyák. Fuzzy rule interpolation matlab toolbox website. `http://fri.gamf.hu`.

[26] R-E Precup, Stefan Preitl, and Péter Korondi. Fuzzy controllers with maximum sensitivity for servosystems. *Industrial Electronics, IEEE Transactions on*, 54(3):1298–1310, 2007.

[27] P. Baranyi and A. Csapo. Definition and synergies of cognitive infocommunications. *Acta Polytechnica Hungarica*, 9:67–83, 2012.

[28] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies–a comprehensive introduction. *Natural computing*, 1(1):3–52, 2002.

[29] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[30] John R Koza. *Genetic Programming: vol. 1, On the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[31] H. Takagi et al. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, 2001.

[32] Hideyuki Takagi. Interactive evolutionary computation. *JOURNAL-JAPAN SOCIETY FOR SIMULATION TECHNOLOGY*, 23(3):191–196, 2004.

[33] Hideyuki Takagi and Hitoshi Iba. Interactive evolutionary computation. *New Generation Computing*, 23(2):113–114, 2005.

[34] Rudolf Jaksa and Hideyuki Takagi. Tuning of image parameters by interactive evolutionary computation. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 1, pages 492–497. IEEE, 2003.

[35] Hideyuki Takagi and Norimasa Hayashida. Interactive ec-based signal processing. In *4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL2002), Singapore*, pages 375–379, 2002.

[36] Nao Tokui and Hitoshi Iba. Music composition with interactive evolutionary computation. In *Proceedings of the 3rd international conference on generative art*, volume 17, pages 215–226, 2000.

[37] Jeffrey Ventrella. Disney meets darwin-the evolution of funny animated figures. In *Computer Animation'95., Proceedings.*, pages 35–43. IEEE, 1995.

[38] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[39] Kenneth V Price, Rainer M Storn, and Jouni A Lampinen. Differential evolution a practical approach to global optimization. 2005.

[40] A.E. Kanlıkılıçer. Interactive differential evolution for facial composite generation. *Genetic and Evolutionary Computation (GECCO), Seatle, USA*, 2006.

[41] H. Takagi and D. Pallez. Paired comparison-based interactive differential evolution. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 475–480. IEEE, 2009.

[42] Ebrahim H Mamdani and Sedrak Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1):1–13, 1975.

[43] Michio Sugeno. An introductory survey of fuzzy control. *Information sciences*, 36(1):59–83, 1985.

[44] Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. *Systems, Man and Cybernetics, IEEE Transactions on*, (1):116–132, 1985.

[45] P. Galambos and P. Baranyi. VirCA as virtual intelligent space for RT-Middleware. In *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 140–145, Budapest, Hungary, July 2011. IEEE.

[46] K.P. Wong and Z.Y. Dong. Differential evolution, an alternative approach to evolutionary algorithm. In *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on*, pages 73–83. IEEE, 2005.

[47] L. T. Kóczy and K. Hirota. Size reduction by interpolation in fuzzy rule bases. *IEEE Trans. on System, Man and Cybernetics*, 27:14–25, 1997.

[48] Gy. Vass, L. Kalmár, and L. T. Kóczy. Extension of the fuzzy rule interpolation method. In *Proc. of the Int. Conf. on Fuzzy Sets Theory and its Applications (FSTA'92)*, pages 1–6, Liptovský Jan, Slovakia, 1992.

[49] P. Baranyi and L. T. Kóczy. A general and specialized solid cutting method for fuzzy rule interpolation. *BUSEFAL*, 66:13–22, 1996. http://www.univ-savoie.fr/Portail/Groupes/LISTIC/busefal/Papers/66.zip/66_03.pdf.

[50] Sz. Kovács. New aspects of interpolative reasoning. In *Proceedings of the 6th. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain*, pages 477–482, 1996.

[51] Sz. Kovács and L.T. Kóczy. Approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rule base as a practical alternative of the classical CRI. In *Proceedings of the 7th International Fuzzy Systems Association World Congress, Prague, Czech Republic*, pages 144–149, 1997.

[52] Szilveszter Kovács and Laszló Kóczy. The use of the concept of vague environment in approximate fuzzy reasoning. *Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences*, 12:169–181, 1997.

[53] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968.

[54] Z. Krizsán and Sz. Kovács. Fuzzy rule interpolation developer toolbox library. In *7th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2012)*, pages 119–123, Timisoara, Romania, May 2012. IEEE.

[55] Miron Kuzma, Rudolf Jaksa, and Peter Sinčák. Computational intelligence in font design. In *9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (CINTI 2008)*, pages 193–203, Budapest, Hungary, 2008. IEEE.

[56] Z. Krizsán and Sz. Kovács. FRI model based pre-evaluation in IEC. In *5th International Symposium on Applied Computational Intelligence and Informatics (SACI 2009)*, pages 35–40, Miskolc, Hungary, 2009. IEEE.

[57] Noriaki ANDO. Openrtm-aist robot middleware website. http://openrtm.org.

[58] Openrdk robot middleware website. http://openrdk.sourceforge.net/.

[59] Yarp robot middleware website. http://eris.liralab.it/yarpdoc/index.html.

[60] Robot technology component specification website. http://www.omg.org/robotics/.

[61] Noriaki Ando, Takashi Suehiro, Kosei Kitagaki, Tetsuo Kotoku, and Woo-Keun Yoon. Rt-middleware: distributed component middleware for rt (robot technology). In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3933–3938. IEEE, 2005.

[62] Noriaki Ando, Takashi Suehiro, and Tetsuo Kotoku. A software platform for component based rt-system development: Openrtm-aist. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 87–98. Springer, 2008.

[63] Noriaki Ando, Shinji Kurihara, Geoffrey Biggs, Takeshi Sakamoto, Hiroyuki Nakamoto, and Tetsuo Kotoku. Software deployment infrastructure for component based rt-systems. *Journal of Robotics and Mechatronics*, 23(3):350–359, 2011.

[64] Marc Laukien. Internet communications engine (ice) website. http://zeroc.com.

[65] Michi Henning. A new approach to object-oriented middleware. *IEEE Internet Computing*, 8(1):66–75, January 2004.

[66] Thomas J Mowbray and Ron Zahavi. *The essential CORBA: systems integration using distributed objects*. John Wiley & Sons, Inc., 1995.

[67] Steve Vinoski. Corba: Integrating diverse applications within distributed heterogeneous environments. *Communications Magazine, IEEE*, 35(2):46–55, 1997.

[68] OMG group. Omg specifications related to corba. http://www.omg.org/spec/index.htm.

[69] Thomas J. Mowbray and Raphael C. Malveau. *CORBA design patterns*. John Wiley & Sons, Inc., New York, NY, USA, 1997.

[70] Michi Henning. The rise and fall of corba. *Queue*, 4(5):28–34, June 2006.

[71] Péter Baranyi. Virtual collaboration arena (virca) website. http://virca.hu.

[72] Péter Baranyi. Cognitive informatics research group website, 2013. http://www.coginfo.sztaki.hu.

[73] Péter Galambos, Christian Weidig, Péter Baranyi, Jan C Aurich, Bernd Hamann, and Oliver Kreylos. Virca net: A case study for collaboration in shared virtual space. In *Cognitive Infocommunications (CogInfoCom), 2012 IEEE 3rd International Conference on*, pages 273–277. IEEE, 2012.

[74] Christian Weidig, Adam Csapo, Jan C Aurich, Bernd Hamann, and Oliver Kreylos. Virca net and coginfocom: Novel challenges in future internet based augmented/virtual collaboration. In *Cognitive Infocommunications (CogInfoCom), 2012 IEEE 3rd International Conference on*, pages 267–272. IEEE, 2012.