Using fuzzy sets to analyse personal preferences on groupware tools

Gabriela Aranda [1] — *garanda@uncoma.edu.ar*
Alejandra Cechich [1] — *acechich@uncoma.edu.ar*
Aurora Vizcaino [2] — *Aurora.Vizcaino@uclm.es*
José J.Castro-Schez [2] — *JoseJesus.Castro@uclm.es*

[1] Departamento de Ciencias de la Computación – Universidad Nacional del Comahue
Neuquén, Argentina. Fax:(+54) 299-4490313

[2] Departamento de Informática, Universidad de Castilla-La Mancha
Ciudad Real, España. Fax: (+34) 926-295351

Abstract. *When stakeholders are geographically distributed communication presents new challenges for software engineering community. In order to improve communication between stakeholders, we have previously proposed to apply learning style models to characterise the way people interact. In this work we enhance that model, using fuzzy reasoning, to find a set of rules that tell us which groupware tools are more suitable according to stakeholders classification.*

Keywords. *groupware, distributed cooperative work, distributed requirements elicitation, cognitive informatics, fuzzy logic, fuzzy sets*

Workshop. *Ingeniería de Software y Base de Datos (WISBD)*

1. Introduction

Communication between geographically distributed stakeholders usually faces four major problems: inadequate communication, knowledge management, cultural diversity and time difference [3].

There are some areas of research that try to minimise the impact of these problems. One of them is the CSCW (Computer-Supported Cooperative Work), which takes into account human behaviour as well as the technical support for people to work as a group in a more productive way. This technical support is called *groupware*. On the other hand, as another approach to face the problems of a distributed requirements elicitation process, the use of cognitive informatics is increasingly commonplace. Cognitive Informatics [2, 10] is a profound interdisciplinary research area that tackles the common root problems of modern informatics, computation, software engineering, artificial intelligence (AI), neural psychology, and cognitive science. One of the most interesting things found in cognitive informatics is that it embodies many science and engineering disciplines, such as informatics, computing, software engineering, and cognitive sciences, sharing a common root problem: *how the natural intelligence processes information*.

Our main goal is to analyse some aspects found in interpersonal communication by applying both concepts – Cognitive Informatics (particularly learning style models) and groupware, aiming at improving the requirements elicitation process.

In a previous work [8] we have classified stakeholders by analysing the characteristics of *Visual/Verbal* and *Active/Reflective* categories in the F-S model [5] and a set of groupware tools we considered the most commonly used during elicitation. To enhance this work, in this paper we propose an approach – based on fuzzy logic and machine learning algorithms – to find a set of fuzzy rules that tell us about stakeholders preferences. We also illustrate the approach with a preliminary case study.

With this in mind, in section 2 we present some basic concepts about groupware tools and the Felder-Silverman learning style model, including the classification we have previously proposed. Following, in section 3, we present the basic concepts of fuzzy sets and fuzzy reasoning. Then, in section 4 we apply these theories to define our model and we present an example based on a case study. Conclusions are addressed in the final section of this paper.

2. Groupware Preferences based on Learning Style Models

Generally speaking, groupware is software for enabling communication between cooperating people working on a common task, and it may include different communication technologies, from simple plain-text chat to advanced videoconferencing [6]. To avoid ambiguities we refer to every simple communication technology as a groupware tool, and to the systems that combine them as groupware packages.

Groupware tools can be classified as synchronous or asynchronous depending if the users have to work at the same time or not. Both categories are important in group work: In the case of asynchronous collaboration because it allows team members to construct requirements individually and contribute to the collective activity of the group for later discussion. This is very important when groups are distributed across time zones because of the difficulty to schedule real time meetings. But real time collaboration and discussions are also necessary components of group Requirements Elicitation (RE) sessions and synchronous tools give stakeholders the chance of having instant feedback [7].

The groupware tools we have analysed in [8] are:

Asynchronous collaboration:

- *E-mail*
- *Newsgroups and mailing lists*
- *Electronic discussion or Forums*
- *Electronic notice or bulletin board*
- *Asynchronous Shared Whiteboard*

Synchronous collaboration:

- *Synchronous Shared Whiteboard*
- *Chat*
- *Instant Messaging*
- *Videoconferencing*

2.1. A categorization based on learning style models

Learning style models (LSM) classify people according to a set of behavioural characteristics [4] and this classification is used to improve the way people learn a given task. These models have been discussed in the context of analysing relationships between instructors and students. Our intent is to take advantage of this model and discussions adapting their application in order to improve communication during a distributed elicitation process. To do so, it is possible to consider an analogy between stakeholders and roles in the models, as Figure 1 shows, since during the elicitation process everybody must “learn” from others. So that, stakeholders may play the role of student or instructor alternately, depending on the moment or the task they are trying to carry on [8].

After analysing five learning style models in [8] we found out that every item in the other models was included in the Felder-Silverman Model [4, 5], so we might build a complete reference framework choosing the F-S model as a foundation.

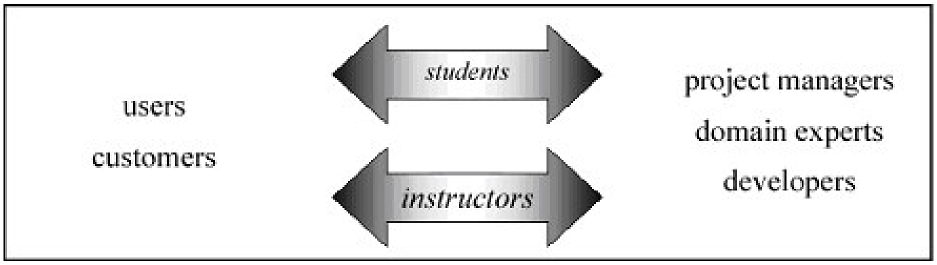


Figure 1: Analogy between stakeholders and roles in learning models

The Felder-Silverman (F-S) Model classifies people into four categories, each of them further decomposed into two subcategories as follows:

– *Sensing* (concrete, practical, oriented toward facts and procedures) or
  Intuitive (conceptual, innovative, oriented toward theories and meanings);
– *Visual* (visual representations of presented material: pictures, diagrams, flow charts) or
  Verbal (written and spoken explanations);
– *Active* (working by trying things out, working with others) or
  Reflective (thinking things through, working alone);
– *Sequential* (linear, orderly, learn in small incremental steps) or
  Global (holistic, systems thinkers, learn in large leaps).

Classification into the different categories is made by the multiple-choice test proposed by Soloman-Felder. As a result, each person gets a rank for each category that suggests his or her preference. A possible result is shown in Figure 2. People may fit into one category or the other depending on the circumstances: people may be "sometimes" active and "sometimes" reflective. The preference for one category may be *strong*, *moderate*, or *mild*. Only when there is a strong preference, people can be catalogued as a member of a certain group.

Preference	**Strong**	**Moderate**	**Mild**		**Moderate**	**Strong**	
Score	**11-9**	**7-5**	**3-1**	**1-3**	**5-7**	**9-11**	
Sensing		✓					Intuitive
Visual			✓				Verbal
Active				✓			Reflective
Sequential		✓					Global

Figure 2: scores in F-S model to characterise people's preferences

Once stakeholders are classified using the previous model, it is possible to take advantage of this categorisation to improve communication during the elicitation process.

In [8] we focused on the categories Visual/Verbal and Active/Reflective to propose a suitable set of groupware tools for each subcategory. Document sharing and electronic notice board were not included in our analysis since communication does depend not only on the way files are interchanged or managed but also on the kind of representation that is used to share information (graphical representations, plain text, combination of them, programs, etc).

Our classification was based on the facts that:

- Visual people need to visualise concepts. They learn best by listing key points, enclosing them in boxes or circles, and drawing lines with arrows between them to show connections. They also like using colour-codes, highlighting words so that everything that is related to one topic has the same colour. Hence, visual tools like shared whiteboard are recommended for them. Messages for visual people should be short in order to be effective: instant messaging and e-mails would be more suitable for them than chat.
- Verbal people would prefer communicating via written and spoken verbal explanations. E-mail and forums would be ideal for them because they would have the chance to share and discuss their ideas. The synchronous tool they would prefer is chat.

As Felder-Silverman Model strategies suggest, audio-visual media like videoconference, is recommendable for visuals as well as for verbals: for visuals because it is important for them seeing their interlocutor gesture to follow a conversation, and for verbals because they have the chance of listening and expressing them verbally.

- Active people tend to retain and understand information best by doing something active with it (discussing or applying it or explaining it to others) that is why they would prefer synchronous tools.
- Reflective people prefer thinking quietly before doing something. That is why they would prefer to work with asynchronous tools.

Our categorisation of groupware tools is shown in Table 3. We have used the sign "++" to indicate those groupware tools more suitable for a given category. The sign "+" indicates a groupware tool is mild preferred by a stakeholder. Finally, the sign "-" suggests that a particular groupware tool is "not suitable" for communication.

		Visual	Verbal	Active	Reflective
Asynchronous	E-mail	+	++	-	++
	Mailing List, Newsgroup	-	++	-	++
	Async. Shared Whiteboard	++	-	-	++
	Forums	-	++	-	++
Synchronous	Instant Messaging	+	++	++	-
	Sync. Shared Whiteboard	++	-	++	-
	Chat	-	++	++	-
	Video Conference	++	++	++	-

Table 1: characterization of groupware tools based on F-S model

3. Overview of fuzzy sets and fuzzy reasoning techniques

Fuzzy sets were proposed to deal with vague words and expressions we use in everyday conversation [9]. For example, in conventional set theories we need to define the term "tall" exactly, for instance like "the set of people more than 190 cm in height". This definition would mean that a person that is 189,9 cm would not be considered tall at all. This does not seem to be representative of what people consider being tall.

On the contrary, fuzzy sets, instead of saying an element belongs to one set or not, are characterized by membership functions, which are noted as $\mu_j : X \rightarrow \{0,1\}$, where the value for each element *i* for the fuzzy set – $\mu_j(i)$ – is the degree of membership of *i* respect to the set *j*.

For instance, to express how tall a person is we can define three fuzzy sets called *low*, *middle*, and *high*. Considering 170cm to be the limit between low and middle, and 180cm to be the limit between middle and high, a person who is close to those values is not considered to belong to just one of them, but it is considered its degree of membership to every fuzzy set.

As an example, the measures of three people (A,B,C), have been expressed in table 2 and represented in a chart in figure 3, by calculating the value of each membership function for each fuzzy set:

	height	low	middle	high
A	181 cm	0	0.4	0.6
B	171 cm	0.4	0.6	0
C	168 cm	0.7	0.3	0

Table 2: value of membership function in fuzzy sets

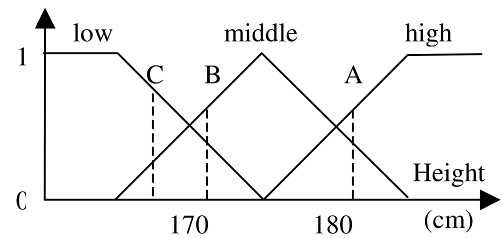


Figure 3: fuzzy sets of height

In this example, the person A is considered *middle* in a degree of 0.4 but *high* in a degree of 0.6, that means that values of membership functions for A are $\mu_{low}(A)=0$; $\mu_{middle}(A)=0.4$ and $\mu_{high}(A)=0.6$.

Low, *middle* and *high* are linguistic variables that can also be used to express people height by saying, for instance: A is higher middle; B is lower middle, C is relatively low. This is highly convenient when talking, since this is the way people usually express themselves.

Once fuzzy sets are known, it is possible to define inference rules for fuzzy reasoning similarly to human reasoning, like "*IF temperature is low THEN turn up heater*".

There are two different approaches to reach the inference rules:

1. by asking to experts (in this case rules are directly written by using the linguistic variables)
2. by analysing examples and inferring rules from them automatically, via machine learning algorithms.

Machine learning tries to generalize the common features of a set of examples, based on induction. For instance, given a set of examples, the algorithm proposed in [1] tries to find a finite set of fuzzy rules able to reproduce the input-output system's behaviour.

Before explaining the algorithm, we need to present some concepts and the notation that will be used:

- $\boldsymbol{\theta = \{e_1, e_2, \ldots, e_m\}}$ represents a set of examples, where *m* is the number of examples; an example e_i have *n* input variables and one output.
- $\boldsymbol{e_i = \{(x_{i1}, x_{i2, \ldots,} x_{in}), y_j\}}$, represents an example, where x_{i1}, $x_{i2, \ldots,}$ x_{in} are the values of the input variables; y_j is the value of the output variable; and $j = 1..m$ is the number of output values.
- **if X is E then Y is $\boldsymbol{y_j}$** represents a rule, where X is the set of input variables, E is a set of subsets of fuzzy labels and Y is the class. It is equivalent to: **if $\boldsymbol{X_0}$ is $\boldsymbol{E_0}$ and $\boldsymbol{X_1}$ is $\boldsymbol{E_1}$ and … and $\boldsymbol{X_n}$ is $\boldsymbol{E_n}$ then Y is $\boldsymbol{y_j}$**, where X is the set $\{X_0, X_1, \ldots, X_n\}$ of input variables and E is a set $\{E_0, E_1, \ldots, E_n\}$ of sets of labels.

The algorithm starts from am empty set of definitive rules and an initial set of examples. The general algorithm is the following:

1. Convert each example in one rule.
2. Remove from the initial set those rules that are the same.
3. Analyse every initial rule so that (whether possible) to extend it and generate a definitive rule.

And the step 3 is divided into the following steps:

3.1. Take an initial rule ***R*** from the set of definitive rules

3.2. Try to *subsume* ***R*** in some definitive rule

> *Definition:* *The rule R_i **subsumes** in the rule R_k if each subset of labels $E_{ij} \subseteq E_{kj}$ and $y_i = y_k$.*

If ***R*** *subsumes*, go to step 3.1

3.3. For each variable in the rule ***R***

3.3.1. For each label not considered yet

3.3.1.1. Prove if it is possible to *amplify* the rule ***R***

> *Definition:* *Given the rule R_i an **amplification** from R_i to $R_{i'}$ is **possible** if there is no rule R_j in the set of initial rules that verify $E_j \subseteq E_{ij}$ and $y_i \neq y_k$.*

3.3.1.2. If amplification of ***R*** is *not* possible, go to step 3.3.1, else amplify the rule ***R***

3.4. If there are rules in the initial set that have not been considered go to step 3.1, else END

About *amplification*, the concept ***possible*** can be redefined using an error percentage, that means, if the quantity of rules that verify $E_j \subseteq E_{ij}$ and $y_i \neq y_k$ respect to the total number of rules, is lower than a given error. In this way it can be reduced the noise in the set of training examples, but there is a disadvantage: the algorithm would lose exactness.

When converting examples into rules it would happen that an example could be classified in two (or more) different rules, which it would be an *ambiguity*. There are two cases that may arise:

- if the example subsumes in one or more rules of the same class (same output), then the example is classified as an element of that class.
- if the example subsumes in two or more rules of different classes (different output), it would be necessary to calculate the *degree of convenience* respect to each rule in which the example can be included and classify the example as an element of the rule with the greatest degree of convenience.

> *Definition:* *The **degree of convenience** of a rule **R** is the lowest value of the functions that are associated with each input variable of **R**, that is $\boldsymbol{\min\{\varphi_i(x_i)\}}$, where φ_i are the membership functions associated with each input variable.*

4. A Case study

In order to define our model, we need: (1) a set of input variables, (2) an output variable, (3) a definition domain for each variable (DDV).

The input variables will be the four categories that correspond to the F-S model as follows:

I = {Sensing-Intuitive, Visual-Verbal, Active-Reflective, Sequential-Global}

The output variable will be the groupware tool a user chooses as his/her favourite one when working on a cooperative task:

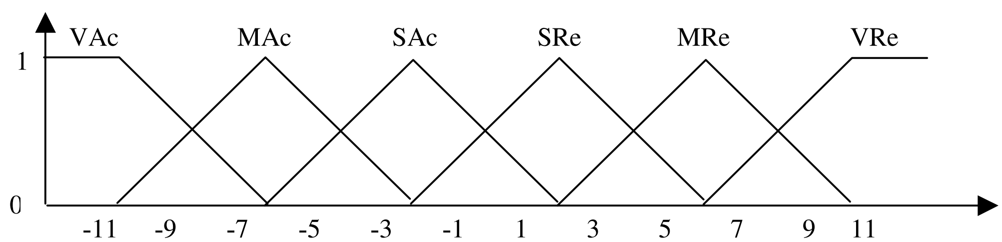
O = {Groupware_Tool}

For each input variable we have defined a domain using the adverbs (and its correspondent abbreviations): Very (*V*), Moderate (*M*) and Slight (*S*). These adverbs correspond to strong, moderate and mild, respectively, in the F-S model, but we have changed their names to avoid confusion respect to the use of the first letter.

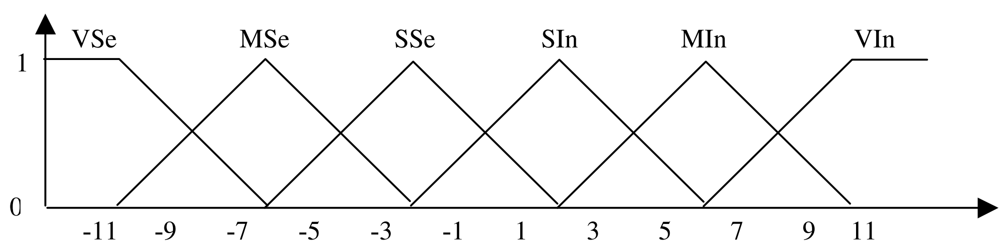
For instance, the definition domain for Sensing-Intuitive input variable would be: Very sensing (Vse), Moderated sensing (Mse), Slightly sensing (Sse), Slightly intuitive (SIn), Moderated intuitive (Min), Very intuitive (VIn).

Following this pattern, the complete set of DDVs for the four input variables would be:

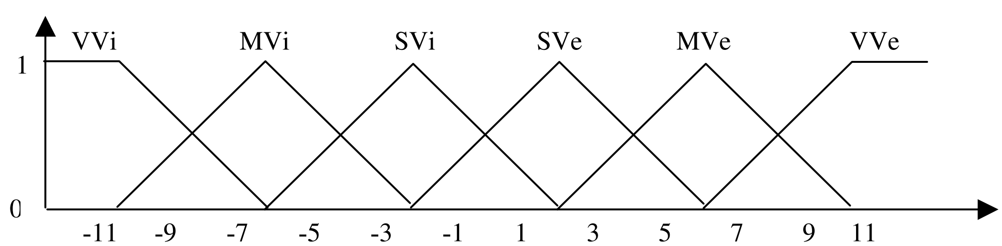
$DDV_{Active\text{-}Reflective} = \{VAc, MAc, SAc, SRe, MRe, VRe\}$



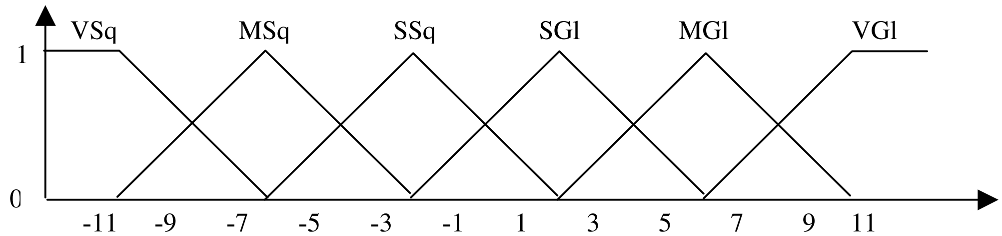
$DDV_{Sensing\text{-}Intuitive} = \{VSe, MSe, SSe, SIn, MIn, VIn\}$



$DDV_{Visual\text{-}Verbal} = \{VVi, MVi, SVi, SVe, MVe, VVe\}$



DDV $_{Sequential\text{-}Global}$ = *{VSq, MSq, SSq, SGl, MGl, VGl}*



For convenience, we show the results of the F-S test with a negative sign for the categories that appear firstly on the presentation of the characteristics (sensing, visual, active and sequential) and with a positive sign for the later (intuitive, verbal, reflective, global).

In the case of the output variable, we have chosen a set of groupware tools.

DDV $_{Groupware\_Tool}$ = *{email, chat, videoconference, forum, …}*

Given a set of examples $\boldsymbol{\theta} = \{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_m}\}$, each example would have the form $\mathbf{e_i} = \{(\mathbf{x_{i1}}, \mathbf{x_{i2}}, \mathbf{x_{i3}}, \mathbf{x_{i4}}), \mathbf{y_j}\}$, where :

$\mathbf{x_{i3}} \in$ DDV $_{Active\text{-}Reflective}$
and $\mathbf{x_{i1}} \in$ DDV $_{Sensing\text{-}Intuitive}$
and $\mathbf{x_{i2}} \in$ DDV $_{Visual\text{-}Verbal}$
and $\mathbf{x_{i4}} \in$ DDV $_{Sequential\text{-}Global}$
and $\mathbf{y_j} \in$ DDV $_{Groupware\_Tool}$

For instance **{(SIn, VVi, VAc, VSq), chat}** would be a possible example.

In our case study, we ask people to fill the test designed by Felder and Soloman, in order to obtain values for each input variable. The test is available on the web site of North Carolina State University (http://www.engr.ncsu.edu/learningstyles/ilsweb.html). On the other side, the output variable expresses the personal preference for a tool.

Since the quantity of examples would be small, and in order to get some significant results, we reduced the set of groupware tools just considering email and instant messaging (IM). The election of email and IM is due to our intention of checking differences in the preference of stakeholders for synchronous and asynchronous tools. To do so, we have asked people to choose which one they would prefer to work cooperatively in a group. People we have asked for doing the test use both (email and instant messaging) for daily work, so that they know very well their functionality and the election is not based just on what they know better.

The set of examples we have collected is the following:

((1,-9,-9,-9), IM)	((-7,-3,-9,-7), IM)	((-3,11,7,5), email)
((-7,-1,-3,3), IM)	((-7,-9,-11,-5), IM)	((-5,-3,-3,-5), email)
((-5,-1,-9,-3), IM)	((-7,-11,-11,-5), IM)	((-7,-5,-5,-5), email)
((3,-3,-5,1), IM)	((-3,-7,-7,-1), IM)	((1, -9, -3, 3),email)
		((-5, -1, -3, 1),email)

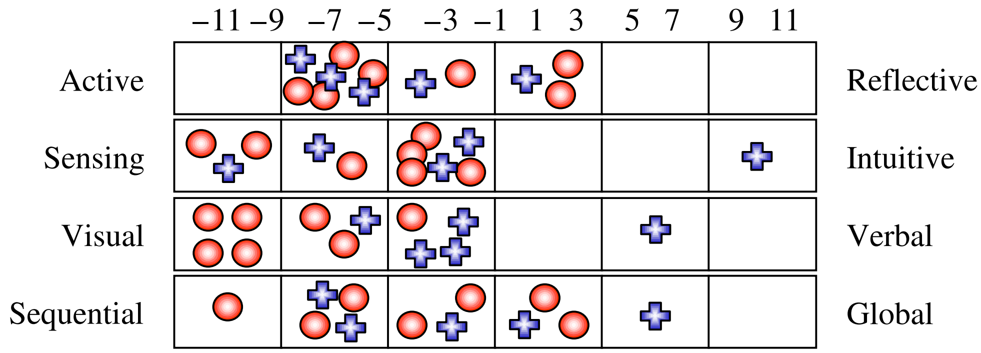


Figure 4: distribution of examples on the F-S categories

Figure 4 shows the distribution of the examples over the categories of the F-S learning style model. This graphic gives an overview about the profile of people being interviewed.

Since the number of examples is small, we assume that the sample is not representative of a real population, however we believe that it can be useful to show the way rules can be inferred from a set of examples, following the steps of the algorithm previously proposed as follows:

Step 1: *Convert each example into a rule*

Because of the nature of our model, there is no possibility of ambiguity at the moment for converting examples into rules. This is because the conversion is direct. For instance, in the case of the input variable *Active-Reflective*, conversion is done by:

-9, -11 $\Rightarrow$ VAc;	-5, -7 $\Rightarrow$ MAc	-1, -3 $\Rightarrow$ SAc
1, 3 $\Rightarrow$ SRe	5, 7 $\Rightarrow$ MRe	9, 11 $\Rightarrow$ VRe

Doing so, the initial set of rules is the following:

Instant Messaging (IM)	**Email**
(SRe, VSe, VVi, VSq) (SRe, SSe, MVi, SGl) (MAc, SSe, SVi, SGl) (MAc, SSe, VVi, SSq) **(MAc, VSe, VVi, MSq)** (MAc, SSe, VVi, MSq) **(MAc, VSe, VVi, MSq)** (SAc, MSe, MVi, SSq)	(SAc, VIn, MVe, MGl) (MAc, SSe, SVi, MSq) (MAc, MSe, MVi, MSq) (SRe, VSe, SVi, SGl) (MAc, SSe, SVi, SSq)

Step 2*: Remove those rules that are the same*

In this example there are only two cases that generate the same rule, which are shown in bold letter in the previous table. Removing those rules, the resulting set is:

Instant Messaging (IM)	Email
(SRe, VSe, VVi, VSq) (SRe, SSe, MVi, SGl) (MAc, SSe, SVi, SGl) (MAc, SSe, VVi, SSq) (MAc, VSe, VVi, MSq) (MAc, SSe, VVi, MSq) (SAc, MSe, MVi, SSq)	(SAc, VIn, MVe, MGl) (MAc, SSe, SVi, MSq) (MAc, MSe, MVi, MSq) (SRe, VSe, SVi, SGl) (MAc, SSe, SVi, SSq)

Step 3: *Analyse every initial rule so that (whether possible) to extend it and generate one definitive rule*

Let's take the tuple **((SRe, VSe, VVi, VSq), IM)** as an example.

a. Firstly, we have to prove if it subsumes into some definitive rule. Since the set of definitive rules is empty we go directly to (b).

b. We convert the initial rule into a new definitive rule and we start the amplification process trying to add the value VAc to the set of labels of the first term.
The rule proposed is (({VAc, SRe}, VSe, VVi, VSq), IM), and the amplification is possible, since there is no rule (VAc, VSe, VVi, VSq), email).
Also the amplification (({VAc, MAc, SRe}, VSe, VVi, VSq), IM) is possible because there is no rule (MAc, VSe, VVi, VSq), email).
And it is possible to include the values SAc, MRe, and VRe, too. Finally we obtain the rule (({VAc, MAc, SAc, SRe, MRe, VRe}, VSe, VVi, VSq), IM).

c. The same process can be applied to the second term and we obtain the rule:
(({VAc, MAc, SAc, SRe, MRe, VRe}, {VSe, MSe, SSe, SIn, MIn, VIn}, VVi, VSq), IM).

d. When we try to do the same with the third term, adding the label MVi:
(({VAc, MAc, SAc, SRe, MRe, VRe}, {VSe, MSe, SSe, SIn, MIn, VIn}, {VVi, MVi}, VSq), IM), the amplification is not possible because of the initial rule ((MAc, MSe, MVi, MSq), email).

e. About the fourth term, the amplification is possible to all the domain values, and the definitive rule is (({VAc, MAc, SAc, SRe, MRe, VRe}, {VSe, MSe, SSe, SIn, MIn, VIn}, VVi, {VSq, MSq, SSq, SGl, MGl, VGl}), IM)

The definitive rule obtained can also be expressed as: *if X_3 is VVi then y is IM*

f. Following, the algorithm should be repeated with the set of initial rules:

Instant Messaging (IM)	Email
(SRe, SSe, MVi, SGl) (MAc, SSe, SVi, SGl) (MAc, SSe, VVi, SSq) (MAc, VSe, VVi, MSq) (MAc, SSe, VVi, MSq) (SAc, MSe, MVi, SSq)	(SAc, VIn, MVe, MGl) (MAc, SSe, SVi, MSq) (MAc, MSe, MVi, MSq) (SRe, VSe, SVi, SGl) (MAc, SSe, SVi, SSq)

and the set of definitive rules:

R_{o}: *if* X_3 *is VVi then y is IM*

g. When we analyse the rules ((MAc, SSe, VVi, SSq), IM), ((MAc, VSe, VVi, MSq), IM) and ((MAc, SSe, VVi, MSq)), IM), we find that they subsume in the definitive rule R_o, so the algorithm continues with the set of initial rules:

Instant Messaging (IM)	**Email**
(SRe, SSe, MVi, SGl) (MAc, SSe, SVi, SGl) (SAc, MSe, MVi, SSq)	(SAc, VIn, MVe, MGl) (MAc, SSe, SVi, MSq) (MAc, MSe, MVi, MSq) (SRe, VSe, SVi, SGl) (MAc, SSe, SVi, SSq)

and the set of definitive rules:

R_{o}: *if* X_3 *is VVi then y is IM*

By applying repeatedly the algorithm we would obtain a set of definitive rules that tell us about the preferences of people according to their personal characteristics. For instance, in our example, the rule R_o can be read as:

> *"If a user has a strong preference for the Visual subcategory, the tool that he or she would prefer is Instant Messaging"*

5. Conclusions

The set of groupware tools used during a distributed development process is usually chosen based on predetermined business politics, or personal preferences of managers, or people in charge of the project. However, this can be inappropriate for someone in the group and it is even possible that someone would not be completely comfortable with the situation. To avoid this situation we propose to use Learning Style Models to classify stakeholders and use this information to choose appropriate groupware tools according to their characteristics.

To do so, we have introduced a model based on fuzzy logic that, given a set of examples, let us obtain a set of rules that tell us which tools people prefer taking into account the classification on Felder-Silverman categories.

We are aware that the quantity of examples we present in this work is not enough to use its results directly because it is not representative enough. However, results are useful to show how the algorithm proposed might be applied to solve this problem.

6. References

1. Castro, J. L., Castro-Schez, J. J., Zurita, J. M.: Learning Maximal Structure Rules in Fuzzy Logic for Knowledge Acquisition in Expert Systems. Fuzzy Sets and Systems, vol. 101, no. 3, (1999), pp 331-342.

2. Chiew, V., Wang, Y.: From Cognitive Psychology to Cognitive Informatics. In Proceedings of Second IEEE International Conference on Cognitive Informatics (ICCI'03), London, UK, August 2003, pp 114-120.
3. Damian, D., Zowghi, D.: The impact of stakeholders geographical distribution on managing requirements in a multi-site organization. In Proceedings of IEEE Joint International Conference on Requirements Engineering (RE' 02), Essen, Germany, September 2002, pp 319-328.
4. Felder, R.: Matters of Styles. ASEE Prism, vol. 6, no. 4, (1996), pp 18-23.
5. Felder, R., Silverman, L.: Learning and Teaching Styles in Engineering Education. Engineering Education, vol. 78, no. 7, (1988, preface 2002), pp 674-681.
6. Gralla, P.: How Intranets Work. Emeryville, California, Ziff-Davis Press, I.S.B.N. 1-56276-441-1, (1996)
7. Herlea, D., Greenberg, S.: Using a Groupware Space for Distributed Requirements Engineering. In Proceedings of 7th IEEE Int' l Workshop on Coordinating Distributed Software Development Projects, Stanford, California, USA, 1998, pp 57-62.
8. Martin, A., Martinez, C., Martinez, N., Aranda, G., Cechich, A.: Classifying Groupware Tools to Improve Communication in Geographically Distributed Elicitation. In Proceedings of IX Congreso Argentino de Ciencias de la Computación, CACIC 2003, La Plata, Argentina, October 2003, pp 942-953.
9. Tanaka, K., Niimura, T.: An introduction to fuzzy logic for practical applications. Springer-Verlag, I.S.B.N. 0-387-94807-4, (1996)
10. Wang, Y.: On Cognitive Informatics. In Proceedings of First IEEE International Conference on Cognitive Informatics (ICCI' 02), Calgary, Alberta, Canada, August 2002, pp 34-42.