

An Intelligent Interface Agent for an Airline Company Web Portal

Luís Barbosa¹, António J.M. Castro¹, Luís Paulo Reis¹

¹ LIACC-NIAD&R, FEUP, Department of Informatics Engineering, University of Porto,
4200-465 Porto, Portugal
{lbarbosa, ajmc, lpreis}@fe.up.pt

Abstract. In this paper we introduce an implementation of an Intelligent Interface Agent to support the use of a web portal in an airline company. The interface agent architecture and data model is presented. We formalized concepts such as relevance and proximity regarding the data structure. The concepts of personal opinion and general opinion are also introduced and formalized. A statistical analysis was performed to obtain the best value when processing the general opinion. Some results of that analysis are presented and we conclude discussing our work and presenting future improvements.

Keywords: Intelligent Interface Agent, User Interface Agent, User Modeling.

1 Introduction

TAP Portugal - Portal DOV [1] is currently a hub of information and services to all crewmembers. It centralizes information and content drawn from diverse sources enabling users to have access to information and services previously accessed at the TAP installations or by telephone. For instance, now one crewmember can view his schedule of work for the day or next week by simply logging to the Portal at his home or in any place with an Internet connection. Alternatively, if a service or department requires a new Training Manual to be made available for the pilots, it can resort to Portal DOV (Web Portal) to publish it.

The Web Portal grants the user a greater mobility and independence, but it has some limitations. As information and documentation are regularly changed, so the user must be up to date about the significant changes. For instance, a crewmember must take an active attitude to check up his roster for relevant changes, even though they may not occur. The process of returning to the Web Portal only to check up the current state of information can turn out a tedious or monotonous task. This sort of tasks are the competence of interface agents, whose purpose is to facilitate the users work by doing boring or repetitive tasks, freeing the user to use the freed time more efficiently. The Web Portal interface agent allows the user to define tasks about the surveillance of information sources by different levels of priority. An example would

be a flight attendant that defines a high-level alert task to monitor changes concerning the days off in her roster and very-low-level alert task associated with the publication of new internal communications. With the help of the agent she could be more promptly acquainted about changes in the current state of information seen as important to her. The Web Portal relying on interface agents to represent crewmembers could result in two major gains: (i) Even more mobility and independence to the users, as the need to access Web Portal to verify the current information diminishes and (ii) increase in the awareness about crucial information for each professional category by making suggestions to the crewmember. Our Interface Agent has two main goals: (1) allow the user to define properly tasks set to monitor changes in relevant content and ensure the user is properly informed and, (2) make suggestions perceptive to the user about new tasks.

Although there will be an interface agent representing each crewmember and appointed to the vigilance of information deemed important by the owner, it does not require direct interaction with the existent agents. The agent will operate blind to the existence of other agents. The suggestions are based on the general opinion of the users with the same profile in regard to what is important to them. An example would be the case where a flight attendant on vacation gets a suggestion to create a new very-high-level alert task on the Web Portal communications. This occurred because in her absence happened major changes in the access to Web Portal and the next announcement would be extremely important to all **crewmembers** access it properly. This example illustrates one way in how the alert task and suggestions can help the crewmembers in keeping up with what is important to the professional category of crewmembers at a time.

The rest of the paper is organized as follows. In section 2 we present some work of other authors regarding intelligent interface agents. Section 3 describes the interface agent architecture including the data model and the relevance and proximity concepts. The interface agent was designed following the methodologies and guidelines presented in [2]. Section 4 shows how we process the relevance, including the definition of the concepts of Personal and General Opinion. In section 5 we present the General Opinion Processing and the statistical analysis to obtain the best value when processing the general opinion. Section 6 shows the results of the statistical analysis. Finally, we conclude our work in section 7.

2 Related Work

An Intelligent Interface Agent (IIA) cooperates with the user in performing its tasks, working as a *personal user assistant*. The agent is pro-active taking the initiative and not passive. Usually, an IIA is not the interface between the user and the application. Instead, it observes the interaction between the user and the program learns with it and interacts both with the program and the user. *UCEgo* [7] is a Unix consultant that, using natural language and modeling user goals and plans, takes the initiative in offering the user information about certain Unix concepts or commands, correcting, at the same time, any misconceptions. *MOKSAF* [5][6][8] is an example of a multi-agent system for user interfaces. The MOKSAF environment is a system for military

mission planning and designed to explore teamwork within heterogeneous human/agent teams. In [4] we can find an example of an IIA applied to an eye disorder diagnosis system. This IIA is able to reason on user actions and provide assistance on the target application. The personal news agent presented in [12], adapts to the user's preferences and interests based on voice feedback from the user. An interesting use of an IIA is presented in [13]. The *QueryGuesser* agent personalizes users' interaction with databases assisting users of computer applications in which retrieving information from a database is a key task.

The IIA uses different machine learning techniques to build user profiles. For example, in [9] the *PersonalSearcher* agent uses hierarchical clustering to obtain different web topics that a user might be interested. The *NewsAgent* presented in [10] uses case-based reasoning and a topic hierarchy to discover which newspaper news a user prefers. Decision trees are applied in CAP [11] to learn users' scheduling preferences.

3 The Interface Agent Architecture and Data Model

Our intelligent interface agent architecture is composed of just one agent as depicted in Figure 1. The operations inside the box represent the roles performed by the agent. The *AlertPreferencesSet* role allows the crewmember to define the behavior of the agent and the *UserAlert* role allows the interface agent to show the information to the crewmember. The *DataSourceMonitor* and *InformationDifferenceView* roles are performed automatically by the interface agent, taking into consideration the preferences of the user and the information that exists on the web portal. On the environment we have information resources that the interface agent can use. The *User Preferences* and *Information Reference* are resources that the interface agent is able to read and write on it. The *Roster* and *Documents* are resources of the web portal and the interface agent is able to read them remotely through a web service. All these resources are part of the interface agent data model, described on the next sections.

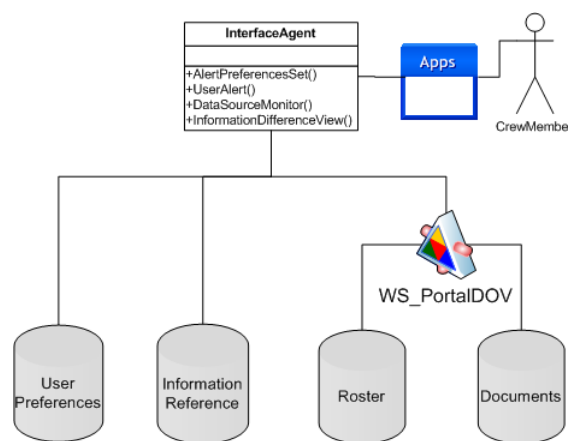


Fig. 1. Interface Agent Architecture

The Interface Agent Data Model is composed of four distinct concepts. First we have the *Data Source* concept, which characterizes the information the agent deals with when searching for new events. The second, *Agent Task*, characterizes the tasks submitted by the user. The third, *CrewMember*, provides the grounds for a practical and proper crewmember depiction. The fourth, *Data Relation*, is oriented to the description of the relations between the previous concepts that compose the model and how they can enable the learning of user behavior. To design the model the technique adopted was the creation of a relational database with the appliance of normalization.

3.1. Data source concept: The data sources are distinct pools of data or information, with a specific purpose and behavior, used by the Web Portal to provide information and the services required by the crewmembers to execute their responsibilities. Every data source has its own entities, or abstract units that help to organize or sort out the information. The entities are themselves composed of a collection of elements. In our work we examined two data sources, the *Roster* and *Documentation*. The *Roster* comprises a set of activities a crewmember will have to execute during a period of time. The crewmember roster is usually planned with a 30-day advance. As activities of crewmember we have for example *Pairing*, *Ground Duty* or *Day Off* to name a few. The *Documentation* encompasses all the available documentation in the Web Portal. A pair of elements defines documentation: *documentation type* and *publisher*. *Documentation type* informs the category to which the document belongs to in the internal organization. It can be in the form of *Internal Communication* and *Schedules* for instance. The *publisher* represents the entity or service responsible by the release of different documents found in the database. In relation to time, the documentation is periodically updated with new documents of different types submitted by the various publishers. The update can take one form of three actions: addition, change and removal of a document.

3.2. Agent task concept: The agent task is the central structure. It represents the user's fundamental requirement, or the need to be updated about events in the data sources. This need is expressed by conjugation of **four** concepts: *who*, *what* and *where*, *when* and *how* and *importance*.

- *Who*: is directly related to crewmember, the owner of the agent, responsible for the creation of tasks to be performed by the agent. Crewmember has its own data structures designed to support the appropriated representation.

- *What and where*: represents the data sources and the selection of inherent elements chosen by the user to be watched over for new events. The data sources are *Roster*, composed of a schedule of activities, and *Documentation*, the collection of types of documents submitted by various publishers.

- *When and how*: represents the task active period. The period of activity is directly related with how or recurrence type. The recurrence type defines the *modus operandi* of the agent in relation to time. There are four types of recurrence, namely: *Never*, *Number of times*, *Until* and *Forever*.

- *Importance*: show how relevant is the task to the crew member.

3.3. Crewmember concept: Represents the application user. It is organized in three views:

- *Application User*: Information required to user access and interaction.

- *Basic Personal Information*: Exhibits personal information about the user.

- *Professional Category*: Represents information about the user in terms of position (or rank) in a flight. It includes information about the crewmember such as: pool (type of aircraft the crewmember flies, for example, narrow body, wide body, A320 fleet, etc.), crew group (flight crew or cabin crew) and rank (the place in the chain of command within a crew group, for example, captain, cabin supervisor, etc.). This view is a fundamental aspect of the extraction of information about crewmember attitude toward the data sources. It is the prime aspect in the crewmember characterization to the judgment of the information present in the web portal, promoting the adequate suggestions.

3.4. Data relation: Data relation is the fundamental path to attain knowledge about the considerations of the crewmember and the constituent professional classes in regard to the data sources when creating the tasks. The *Data Relation* provides the interface agent with the means to extract suggestions to the crewmember about new tasks. The knowledge is captured by the conjugation of three concepts: relevance, proximity and view. These three concepts serve as a function to transmit the weight of a relationship between elements of a data source and a view derived from the professional category. The relationship weight function has the generic form:

$$Weight = F(Elem(Type(k, ds), v)) \quad (1)$$

F – is the function applied to extract the weight of the data source element (either the relevance or the proximity function, both defined below).

$Type(k, ds)$ – function to return an entity k part of the data source ds structure.

$Elem(Type(k, ds))$ – is the function that returns an element from the data source ds . The variable ds can be one of two values: *Roster* or *Documentation*.

v – the view represents the professional category chosen for interpretation of the element data source. At the foundation of the professional category is the concept rank.

3.4.1. Relevance: Relevance, or how significantly the data source element is employed, is the function revealing the importance one element present in a data source has to a professional category. The relevance function (mentioned in Equation 1), is:

$$Relevance = R(Elem(Type(k, ds), i), v) \quad (2)$$

$Type(k, ds)$ – the same as in equation 1.

$Elem(Type(k, ds), i, v)$ – is the function that returns the element i from the entity present k in data source ds . The variable ds can be one of two values: *Roster* or *Documentation*.

v – the same as in equation 1.

3.4.2. Proximity: Proximity, or how strongly each element from the data source is associated to another when defining tasks, informs how frequently each element present in a data source relates to another by the professional category standpoint. To calculate the weight from the relationship the function *Proximity* requires two different elements from the same data source. The proximity function (mentioned in Equation 1) is:

$$Proximity = P(Elem(Type(y, ds), i, v), Elem(Type(z, ds), j), v), if(y = z) \rightarrow i \neq j) \quad (3)$$

$Elem(Type(y, ds), i, v)$ – is the function that returns the element i from the entity y present in data source ds . The variable ds can be one of two values: *Roster* or *Documentation*. When addressing the same entity ($y=z$), variables i and j must be different;

3.4.3. Data relation structure: To capture the relationships between the data source elements and the professional categories, a structure capable of supporting the functions of *relevance* and *proximity* in each data source was defined. The central component in the data relations' structure would be the professional category viewpoint used to evaluate each data source element, namely *pool*, *crew group* or *rank*. The professional category chosen to define the viewpoint between the three was *rank*. The reason to the choice made was expediency, since through the rank we can also establish the relevance and proximity based on the crew group viewpoint with reduced effort. The number of tables needed to express relevance and proximity in each data source by a viewpoint is proportional to the number of elements types present in the same data source. For *relevance* the number of tables is given by n , where n is the number of entities present in the data source. For *proximity* the tables must express the closeness in each entity and between pairs of entities. So the total number of tables follows the progression:

$$T_{tables} = n + \frac{n!}{2(n-2)!} \quad (4)$$

In the worst-case scenario the total number of tables necessary to form Data Relations structure is:

$$nViews * \sum_{i=1}^{nDatasources} 2 * nElemTypes(i) + \frac{nElemTypes(i)!}{2(nElemTypes(i)-2)!} \quad (5)$$

$nViews$ – number of viewpoints to judge the entities.

$nDatasources$ – number of data sources.

$nElemTypes(i)$ – number of element types the exist in data source i .

For example in our case, with one viewpoint and two data sources, by application of the formula the total number of tables would be $1*(2+2*2+1) = 7$.

4 Relevance Processing

Overview: Relevance Processing consists in revising the relevance of a data source element when a new task is submitted by a crewmember. The revision is directly aligned to the view derived from the crewmember professional category. Whenever a new task is created, a relevance value is given directly by the crewmember. The relevance has the goal of establishing how significant the task is to the user and to define the agent behavior to act accordingly. **Hence this relevance defines the importance of a set of elements of a data source at the moment to the user.**

Before submission, the task is processed to accommodate the user behavior. The user behavior during the interaction may convey implicit value pointing the agent to define more accurately the final relevance. Finally, when the task is accepted, the *Data Relations* structure, which expresses how the elements relate to the users and between themselves, must be updated to include the new relevance and so evolve as the crewmembers attitude about the same elements changes. How the elements relate to each other by a professional category view is the support to elaborate suggestions to the crewmember about new tasks. For the sake of understanding lets adopt an alternative naming convention, i.e. the idea of “*personal opinion*” and “*general opinion*”.

Personal opinion definition: The relevance deriving from the user interaction as the crewmember “*personal opinion*” at the moment in regard to a set of elements is:

$$Op = R(Elem(Type(k, ds), i), ProfView(cm), Crewmember), i \rightarrow 1 \dots total \quad (6)$$

ProfView(cm) – function which extracts the view representing the professional category from the crewmember *cm*.

Elem(Type(k, ds), i, ProfView(cm)) – function that returns the element *i* from the entity present *k* in data source *ds*. The variable *ds* can be one of two values: *Roster* or *Documentation*.

General opinion definition: The value of the relevance found in *Data Relations*, for the same set of elements viewed by a professional category to which the crewmember belongs, is expressed as:

$$Og = R(Elem(Type(k, ds), i), ProfView(cm), Data Relations), i \rightarrow 1 \dots total \quad (7)$$

5 General Opinion Processing and Statistical Analysis

Overview: Because it was opted to keep only one general opinion at a time, the method defined to deliver a new general opinion would be inspired by the exponential moving average. The key issue in the process of generating the new general opinion would be the smoothing factor α applied in the exponential moving average equation (see equation 9). The statistical analyses procedure adopted follows the five steps method mentioned in *Estatística* [3], that is, (i) goal setting and conditions, (ii) definition of data gathering process, (iii) data gathering, (iv) data analysis and (v) ascertainment of conclusions about the population acquired in the previous step.

Goal settings and conditions: The main purpose of this study was to select the most acceptable method by comparing their behavior in a simulation with different smoothing factors to the average personal opinion. Another interesting condition to observe was how the very short memory term would affect the extraction of general opinions. All the methods were subordinated to the structure imposed by the *Data Relations* and so only one record of the general opinion would be stored at a time. This factor, although limiting, could be dealt without difficulty given the following conditions were met:

- Small classification range of the opinions, to limit possible irreconcilable deviations. In this case the range would be in a fairly manageable five qualities scale: *Very low, Low, Normal, High, Very High*;
- General opinions included all the previous past general opinions and personal opinions feedback. This condition is mandatory and is inherent to all the methods devised;
- Personal opinions would not deviate too much when addressing opinions to the same elements. Given the relative small scale, only dealing with extreme opinions would be of concern.

The ability to react to greater and punctual deviations is dependent in two interconnected factors:

- The structure of the data used to capture the general opinions;
- The smoothing value chose for calculation of new general opinions with new personal opinions.

While the personal opinions are based on five level strata according to their importance, all the intermediate calculations about the general opinion can work with different precisions. When needed a suggestion the general opinion obtained would be converted by rounding to a discrete value in the same five level scale, as indicated in equation 8.

$$Sug_i = round(Og_i) \quad (8)$$

The ability to translate later the data to a proper discrete group enables the future general opinion to be more sensitive and consistent with the evolution of the opinions given by the crewmembers. By converting the general opinion to a suggestion via rounding allows the five groupings to have the same dimension and so ensure that bias toward a discrete value is avoided. The exponential moving average is presented in equation 9.

$$EMA_t = EMA_{t-1} + \alpha(Price - EMA_{t-1}) \quad (9)$$

EMA_t – new exponential moving average value;

EMA_{t-1} – previous exponential moving average value;

$Price$ – current value given;

α – percent-based smoothing factor, $0 \leq \alpha \leq 1$

Equation 10 shows the ensuing formula, which drove the calculation of new general opinions. The smoothing α determines how the general opinion should shift toward the new opinions.

$$Og_i = Og_{i-1} + \alpha(Op_i - Og_{i-1}), \alpha \in]0,1] \quad (10)$$

Og_i – new general opinion value;

Og_{i-1} – previous general value;

Op_{i-1} – submitted personal opinion;

The main purpose was to directly observe the behavior of the method with different values $\alpha=0,5$, $\alpha=0,33$ and $\alpha=0,66$ to select the most close to the average personal opinion in a sequence of personal opinions submission. Because only one

record is stored, the general opinion must be recalculated every iteration for the new personal opinion.

Data gathering process: The data gathering process would have as basis the simulation of *personal opinion* submission for the same entity. Every simulation was established to evaluate the behavior of the methods in regard to two outcomes: how the *general opinion* and the *suggestion* made would respond after a *personal opinion* was submitted. The data gathering process consisted in observing the outcomes *Og* (equation 10) and *Sug* (equation 8) through the application of different factor α to the submission of n *personal opinions*. The number of personal opinions defined to the simulation was 50. The recorded values were compared by the perspective of the mean average (\bar{X}) and mode (Mod). The random sampling was utilized to avoid bias introduction [3] as the typical pattern of submitting personal opinions was not known. Each *personal opinion* had the value scale of submission translated from a quality based to a discrete scale from 1 to 5 (very low to very high). The same discrete scale was used to convert *general opinion* to *suggestion*.

$\bar{x}(\bar{x}(Op) - (Og_i, \alpha)), i \in [1..50]$	$\bar{x}(\bar{x}(Op) - (Sug_i, \alpha)), i \in [1..50]$
$\left\{ \begin{array}{l} \bar{x}(\bar{x}(Op) - (Og_{i,\alpha=0.5})) \\ \bar{x}(\bar{x}(Op) - (Og_{i,\alpha=0.33})) \\ \bar{x}(\bar{x}(Op) - (Og_{i,\alpha=0.66})) \end{array} \right\}$	$\left\{ \begin{array}{l} \bar{x}(\bar{x}(Op) - (Sug_{i,\alpha=0.5})) \\ \bar{x}(\bar{x}(Op) - (Sug_{i,\alpha=0.33})) \\ \bar{x}(\bar{x}(Op) - (Sug_{i,\alpha=0.66})) \end{array} \right\}$

Fig. 2: Group of calculations used in study 1

The entity would have a neutral *general opinion* value in the start. In the base simulation two descriptive statistics measurements were inspected:

- \bar{X} – the average mean provides a rough estimate about the average opinion of the users and the behavior of the methods applied to extract general opinions and suggestions.
- Mod – the mode indicates the value or range of values where the data concentration is bigger. In the specific case tells about the crewmembers most submitted value and general opinion and suggestion value made.

$ \bar{x}(Op) - \bar{x}(Og_{i,\alpha}) , i \in [1..50]$	$ \bar{x}(Op) - \bar{x}(Sug_{i,\alpha}) , i \in [1..50]$
$\left\{ \begin{array}{l} \bar{x}(Op) - \bar{x}(Og_{i,\alpha=0.5}) \\ \bar{x}(Op) - \bar{x}(Og_{i,\alpha=0.33}) \\ \bar{x}(Op) - \bar{x}(Og_{i,\alpha=0.66}) \end{array} \right\}$	$\left\{ \begin{array}{l} \bar{x}(Op) - \bar{x}(Sug_{i,\alpha=0.5}) \\ \bar{x}(Op) - \bar{x}(Sug_{i,\alpha=0.33}) \\ \bar{x}(Op) - \bar{x}(Sug_{i,\alpha=0.66}) \end{array} \right\}$

Fig. 3: Group of calculations used in study 2

The stored table data would be processed in three groups of study involving pairs (Opi , Ogi) and (Opi , $Sugi$) from each α and a process of study. The first study determines the global average difference between the average personal opinion and the general opinion and suggestion of the simulation. Figure 2 shows the group of calculations used in this study.

Having as reference the mean average personal opinion, the second study finds how the opinion and general suggestion mean average deviates in the simulations. Figure 3 shows the group of calculations used.

6 Results

The first study was conceived to see the average difference in every record between the most regular general opinion and suggestion to the average personal opinion. The difference gives a sense of how close the general opinion and suggestion generated by α is to the average personal opinion. Table 1 shows the result obtained in study 1.

Table 1: Results of Study 1

	$\alpha = 0,5$		$\alpha = 0,33$		$\alpha = 0,66$	
	<i>Og</i>	<i>Sug</i>	<i>Og</i>	<i>Sug</i>	<i>Og</i>	<i>Sug</i>
<i>Op</i>	0,59	0,62	0,40	0,57	0,77	0,80

The second study shows the direct difference between the personal opinion average mean to general opinion and suggestion. Greater the value of deviation, greater chance of the general opinion and suggestion not be synchronized with the most prevalent personal opinion. Table 2 shows the result obtained.

Table 2: Results of Study 2

	$\alpha = 0,5$		$\alpha = 0,33$		$\alpha = 0,66$	
	<i>Og</i>	<i>Sug</i>	<i>Og</i>	<i>Sug</i>	<i>Og</i>	<i>Sug</i>
<i>Op</i>	0,01	0,00	0,02	0,04	0,00	0,06

The first study compares the performance of the α by every simulation record while the last evaluates the performance globally by processing the stored results of the 50 simulations in regard to the average value.

The $\alpha = 0,5$ updates the current general opinion by making it lean toward the new opinions with a neutral pace. This value neither highlights the general opinion nor accentuates the contribution of new personal opinions.

The $\alpha = 0,33$ emphasizes the value of the current general opinion, making the progression to new opinion slower. A factor $\alpha = 0,33$ can reduce the impact of random extreme opinions which could distort the estimate of a realistic general opinion.

The $\alpha = 0,66$ facilitates the adoption of new personal opinions. This method is ideal when sudden changes in the personal opinions occur periodically spaced, making the general option harmonize faster. If the behavior of the personal opinions shifts constantly around extreme values makes the general opinion too sensible to the changes. In Table 1 we can see that the $\alpha = 0,33$ has the better performance. In table

2, $\alpha = 0,5$ had the lowest deviation score ($Og = 0,01$; $Sug = 0,00$) making the values returned by his application usually more close to the personal opinion tendency.

To a lower smoothing factor the general opinion and the suggestion tends to be more close to the average personal opinion, while they seem to suffer from worse effectiveness in keeping pace with the most recent personal opinion. By the objective of the analyses both $\alpha = 0,5$ and $\alpha = 0,33$ are generally better than the alternative, having the lower deviation and the smaller difference value from the average personal opinion. This performance was already expected at the beginning of the simulation. The combination of these two factors means that lower α gets values more close to the most common personal opinion at a very satisfactory rate. The smoothing value should be updated in the future to increasingly adapt to the behavior of the population:

- If the population tends to have a very regular set of opinions punctuated with sporadic extreme opinions α should decrease.
- If the population shifts to new set of opinions periodically α should increase;

In our work we have set the value of α to **0,45**.

7 Conclusions and Future Work

By implementing this Web Portal Interface Agent we hope to improve the dissemination of crucial information to each professional category. All the crewmembers help to define what is important to them at a time and indirectly share that knowledge or opinion between them in the form of general opinion. Because the gathering and processing of the tasks is continuous this makes current suggestions always sensitive to the needs of the users. Two immediate tests can provide more accurate results about the actual performance of the agent, being the first one a more extended use of the interface agent by the crewmembers and evaluation of the crewmember's feedback about the agent behavior and utility. With time, a more precise perception about the general behavior of the crewmembers can become more evident and so help to locate a smoothing factor more efficient. The smoothing factor would be gradually adjusted according to the variations.

Alternatively, a different approach to the calculation of suggestion can be adopted. The current data structure supports the implementation of different methods to find a general opinion by different ways because it stores the data concerning all the past tasks. The past data is an enormous source of information for the application of different statistical methods. An interesting test would be to evaluate the difference in performance between the exponential moving average and suitable method of comparison like mean average or preferably the weighted arithmetic mean. Since the data structure allows with little effort the implementation of different methods for the calculation of relevance we have greater flexibility and a greater chance of adaptability of the agent to current and future patterns of crewmember behavior.

One interesting path would be to expand the number of present category views applied to extract suggestions and so get increasingly more detailed users profiles. While the number of viewpoints to evaluate opinions may grant more insight about the nature of each user and consequently more adequate answers, it comes at a cost.

The cost would derive directly of the rate of growth of the data structure presented in section 3. One future course of action could be to broaden the number of data sources monitored. Currently the Interface Agent is expressly conceived for crewmember in regard to the interaction with the two data sources, *documentation* and *roster*. Yet there is more information sources present in **Web Portal** with the possibility of exploration. In our opinion the interface agent is a benefit working in both ways. By the perspective of crewmember, promotes the timely acquisition of relevant information granting the user with more independence and autonomy. To the **Web Portal**, the major improvement comes with clarification of the notion of crucial content throughout the professional categories. This allows a more efficient diffusion of information without the direct interference of the airline company personal.

Acknowledgments. The second author is supported by FCT (Fundação para a Ciência e Tecnologia) under research grant SFRH/BD/44109/2008. The authors are grateful to TAP Portugal for allowing the use of real data from the company.

References

1. TAP Portugal – Portal DOV [online] <http://portaldov.tap.pt>, consulted on April 2009.
2. Castro, A. and Oliveira, E.: The rationale behind the development of an airline operations control centre using Gaia-based methodology. *IJAOSE*, Vol. 2, No. 3, pp.350–377 (2008).
3. Guimarães, Rui Campos and Cabral, José Sarsfield.: *ESTATÍSTICA*, McGraw-Hill de Portugal. (1997)
4. Jing, Yanguo, Brown, Keith and Taylor, Nick.: Intelligent Interface Agents for a System to Diagnose Eye Disorders. Proceedings 1st IJCAAM, Bologna, Italy, pp. 794-795 (2002).
5. Lenox, T., Hahn, S., Lewis, M., Payne, T., and Sycara, K.: Task Characteristics and Intelligent Aiding, Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics, Nashville, TN, 1123-1127, (2000).
6. Lenox, T., Payne, T., Hahn, S., Lewis, M., and Sycara, K.: Agent-based Aiding for Individual and Team Planning Tasks, Proceedings of IEA 2000/HFES, (2000).
7. Chin, D.: Intelligent interfaces as agents, in *Intelligent user interfaces*. Sullivan, J. and Tyler S. (editors), ACM Press, New York, USA (1991).
8. MOKSAF web site [online]. <http://www.cs.cmu.edu/~softagents/moksaf/index.html>. Consulted in April 2009.
9. Godoy, D., and Amandi, A.: PersonalSearcher: An Intelligent Agent for Searching Web Pages. *Advances in Artificial Intelligence-LNAI 1952*, pp. 43-52, (2000).
10. Cordero, D., Roldán, P., Schiaffino, S., and Amandi, A.: Intelligent Agents Generating Personal Newspapers. In Proceedings ICEIS 99, pp. 195–202 (1999).
11. Mitchell, T., Caruana, R., Dermott, J.M., and Zabowski, D.: Experience with a Learning Personal Assistant. *Communications of the ACM* 37(7), 80–91, (1994).
12. Billsus, D. and Pazzani, M.J.: A personal news agent that talks, learns and explains. In O. Etzioni, J.P.Muller, and J.M. Bradshaw (Eds.), *Proceedings of the 3rd Agents'99*. ACM Press: Seattle, WA, USA (pp. 268–275), (1999).
13. Schiaffino, S., and Amandi, A.: An Interface Agent Approach to Personalize Users' Interaction with Databases. *Journal of Intelligent Information Systems*, Vol. 25, No. 3, pp. 251-273 (2005).