Agent-Based User Interface Adaptivity in a Medical Decision Support System

Sue Greenwood, John Nealon and Peter Marshall

Department of Computing, Oxford Brookes University,
Oxford OX33 1HX, UK
{sgreenwood, johnnealon, pmarshall}@brookes.ac.uk

Abstract. Previous work at Oxford Brookes University developed a system to advise on diabetes treatment that enabled data to be displayed according to the choices of each user. Due to the time critical nature of the problem, spending time searching through the data was not feasible. This reduced the usefulness of the system in the clinical setting for which it was designed. Thus a more automated approach was required. A multiagent system has been utilised to drive the adaptivity. A set of simple agents, each concerned with a single aspect of the system, communicate with each other and the suggested summary is a result of the emergent behaviour of the whole system. While emergent behaviour is used in other areas where agents have been applied, notably robotics, it is novel to use this approach in adaptive interfaces. This paper first considers the use of reactive agents to provide a context for the application of emergence in the area of self-adaptive interfaces. The field of adaptive interfaces is also considered to identify approaches that have been used in the past. An emergent multiagent system using a two-layer model is then described. This approach has been applied and tested to the problem of providing self-adaptivity at the interface to allow for decision support to be delivered in real-time for a clinician to employ.

1 Introduction

The work described in this paper is concerned with the provision of accurate dosage advice for diabetic patients. This project has produced a PDA-based system into which patients enter various details about their diet and physical condition and are given accurate dosage advice for the insulin that they require [1]. In collecting data from the user, this system is also a repository of information about the day-to-day condition of the patients.

As part of the project, a desktop computer based system was also developed to allow the data from the PDA to be made available to the clinician. This system provided the required data visualisations. However, the context in which it is used, that of a standard consultation, does not allow enough time for a clinician to access

J. Vázquez-Salceda, J. Díaz de León. (Eds.). e-Health: Application of Computing Science in medicine and Health

the data to find the salient information from a given dataset (a patient record) or browse the visualisations. The current project was undertaken to address this problem.

As the clinician did not have sufficient time to make use of a system that allowed them to analyse the data, it was decided instead to investigate the possibilities of developing a computer system that carried out the evaluation and allowed the medical professional to simply review what the system had produced. The constraints that had to be met were that the system should act in real time to find the interesting patterns in the data and then meet the individual requirements of each clinician. A prescriptive system that forced the user into a particular way of working would have been anathema to the work habits of clinicians, the intended users, so a system that could adapt itself to the work habits of the users was required. To this end, an adaptive interface was developed. This system needed to be capable of learning user preferences and relating them to the patterns in the data. The system was required also to provide a summary of the dataset in the form of a set of recommendations for data views. A multiagent system was developed to drive the adaptivity. A set of simple agents, each concerned with a single aspect of the system, communicate with each other and the suggested summary is a result of the emergent behaviour of the whole system.

2 Diabetes Mellitus

Diabetes mellitus is one of the most common acute diseases and affects a significant proportion of the population. The least severe form of the disease is known as impaired glucose tolerance. This may or may not develop into full-blown diabetes, which has two forms, type I diabetes and type II diabetes. Type II is less severe but more common, and is often treated by a strict diet and/or tablets. In some cases however, insulin replacement therapy may be required. In the case of type I diabetes, patients always requires insulin replacement therapy. Type I sufferers may experience some temporary remission during the early stages of the disease (known as the honeymoon period) but after this, they will require regular insulin replacement therapy for the rest of their lives.

Diabetes is a disease caused by the breakdown of one of the body's feedback mechanisms. Glucose (sugar) is the body's natural energy source. It is obtained from food and used throughout the body although primarily by the muscles and brain. While the liver plays a large part in the regulation and storage of blood glucose, it is insulin as produced by the pancreas that is the most important hormone.

As with all hormones, insulin is a messenger chemical that is secreted as and when needed by the body to create the feedback loop in a process. In this case it is concerned with the maintenance of the blood glucose level. When an excess of glucose is detected in the pancreas, insulin is secreted which promotes the transfer of glucose to cells. As the blood glucose level drops insulin production decreases and so the amounts of glucose present in the blood returns to normal. Another hormone glucagon performs the opposite task encouraging the release of glucose from cells when the level drops too low. This is normally a highly effective system with glucose concentrations kept at between 4 and 6 mmol/l.

In Type I diabetes, there is a major insulin deficiency and the level of blood glucose increases. The body attempts to remove some of this by excretion in urine. To attempt to compensate, the kidneys must work continuously to remove excess glucose and this produces the characteristic symptoms of increased urination and thirst. Eventually the body can no longer metabolise glucose and so must turn to another process to provide energy. This is achieved by the breaking down of fat cells. This highly inefficient process produces organic acids that provide an important alternative energy source for the brain when present in small amounts. In large concentrations, such as those observed in untreated Type I diabetes, these organic acids or ketone bodies accumulate in the blood stream and urine. They eventually reach a critical concentration and ketoacidosis occurs. This leads to coma and death.

By taking insulin, the diabetic patient can cause their blood glucose levels to reduce but if they take too much, the body's blood glucose level can drop too low again causing coma and death. Thus, keeping the blood glucose level at a safe level is a matter of maintaining a fine balance. Normally the body can modify its insulin and glucagon production as required to allow control at the level of relatively subtle shifts with the feedback systems in the body ensuring that this process can be carried out accurately. When insulin production is impaired and the patient must provide the necessary insulin themselves, it becomes more difficult to control the system as insulin will be taken at a few relatively fixed points in the day and feedback cannot be constantly provided and the first hint that the blood glucose level has moved out of range can be the onset of a hypoglycaemic reaction or 'hypo' when the body attempts to shut down.

Recent major studies have confirmed conclusively that the main aim of diabetes treatment should be to maintain the body's blood glucose level as near to the normal level as possible [2, 3].

In diabetic patients the body cannot produce its own insulin, so the patient is required to regularly take enough insulin to balance the blood glucose levels. This involves the patient in trying to determine how much insulin they will need based not only on the current blood glucose level but also considering various other factors. These include what, and how recently, the patient has eaten; to what intensity and how recently they have taken exercise; how they are generally feeling health wise; the time of day and whether their blood glucose has recently gone outside safe limits.

3 Diabetes Treatment System

The multiagent system described in this paper is part of a research and technical development collaboration with the Diabetes Trials Unit of Oxford University. The objective of the partnership is to allow diabetic patients, clinicians, diabetic nurses and researchers to interact efficiently and effectively with a highly integrated diabetes treatment system. The system architecture includes several interacting elements. The two core elements are the handheld diabetic patient insulin dosage advisor and the diabetes clinic decision support system described in this paper.

The portable advisor, POIRO Mk2, is based on the POIRO system developed by the collaboration [1]. POIRO proved highly effective and user friendly, but was developed originally on a large and expensive hand-held computer, the Epson EHT-10. It was ported to the Apple Newton and subsequently to Palm OS® PDAs, incorporating significant improvements. The PDA system has recently completed a successful clinical study at the Radcliffe Infirmary in Oxford.

At a clinic visit the diabetic patient provides the doctor with details of their blood glucose levels, insulin taken, and 'hypos', recorded in a 'log-book'. However, these records even if completed in full - which is not universally the case - do not provide the clinician with information concerning the factors affecting the patient's metabolism when each glucose reading and corresponding insulin injection was taken. With the advent of the PDA based system, the diabetic patient now has an incentive to enter not only the data which the logbooks were designed to store but also background factors as these are needed to allow the system to make its recommendations. As part of its operation, the PDA-based system stores this information.

Diabetic Patients got on well with the PDA-based system in trials and when in use, the system amassed useful records not only of their glucose levels and insulin dosages but also collected data concerning some of the relevant environmental factors that influenced these. It was felt that if this information could be uploaded to a clinician's system, it could provide a great deal more information about the day-to-day condition of the patient than had previously been possible. To this end, a data visualisation package was created to allow healthcare professionals to view graphs and tables summarising the relevant points from the data.

One of the key features of a clinician/patient consultation is that it takes place over a very short time, perhaps between seven and fifteen minutes. As this is all the time available to the clinician with a patient to discuss how their health has been over the previous period of three to six months, there is insufficient time to allow for the use of the uploaded data, except for the more experienced clinicians. However, it is increasingly the case that diabetic patients are being treated by general practitioners or specialist nurses, who do not have the same level of experience and knowledge to interpret the data in the way that experienced clinicians are able to. Therefore, support in analysing the data was required.

It was considered necessary to automate the process of finding the relevant patterns in the data, so that the amount of time that was needed to make effective use of the system was reduced significantly. However, different clinicians work in different ways and might be interested in viewing the data in different ways too. They might even wish to view the data in differing ways for a patient different, which further complicates the issue. So while adding pattern finding functionality is a useful first step; because of the way that clinicians work and the very short time frame in which the interaction with each patient takes place, this in itself is not sufficient to make the system worthwhile. If time were not such an important factor, then simply producing a system that was customisable might be sufficient. The fact that a clinician does not have the time to spend customising a system means that any useful system had to be able to tailor itself to the clinician rather than relying on the clinician tailoring the system to their needs.

Our work investigates whether multiagent-based emergent adaptivity at the interface can produce useful and meaningful behaviour in the form of automatically adapting the system to the user. To achieve this, the system must be able to determine the interests of the user and any patterns in the data that are relevant to those interests. These interests could vary as each patient is considered, so a system with the ability initially to learn a clinician's interests and then dynamically change the areas of interest to be investigated with each new set of data is required.

An adaptive approach was chosen to allow for the fact that there *are* patterns in the interactions of the users, that are different for different users. The time constraints put on clinicians , the target users provided one of the main considerations. That is,. The clinician needed to access the factors relevant to them in the data, taking the minimum amount of time from the consultation.

The initial focus of the work was on the needs of clinicians. However,, since the management of diabetes is changing, moving from hospital based treatment to treatment centred in Health Centres it is entirely possible that in the future other health care professionals might come to need access to the data provided. The system developed was thus required to accommodate to the needs of potentially disparate classes of users in addition to the variation within the initially considered user class.

In reviewing the requirements of the system, they were:

- to learn over time to both tailor itself to the user and to make its pattern identification effective.
- to cope with the sometimes short time scale of operation.
- to correlate between user actions and data patterns.
- any system suggestions should be an adjunct to the main operation so that the user could use or ignore them as they saw fit.

The idea of allowing the system to carry out part of the work, to have, as Wooldridge [4] suggests, agents in certain circumstances take the initiative rather than wait for users to say exactly what they require of the system, is very appealing. This is especially the case when one considers the perennial issue of the time constraints that clinicians are under. As the goal in this project was to allow the system to do the initial filtering work for the clinician, the idea of an agent-based approach seemed an appropriate one to consider.

4 An Agent-Based Approach

In his definition of intelligent agents, Muller [5] classifies agents into three main categories: reactive agents; deliberative agents and interacting agents. He then goes on to develop a taxonomy based on these three types and suggests the type of architecture that might be applicable for particular classes of problem. These classifications are worth a closer examination.

Reactive agents Muller defines as those that express reactivity and real time behaviour. Typically, these will have little if any explicit world model and will make decisions at run-time based on simple behaviour-action rules. Classically, this approach has been used in the field of robotics with Brooks subsumption architecture [6] being the ubiquitous example. This is a layered architecture where each individual entity is only concerned with a particular part of the task and it is through the combination of all the activity that the functionality of the whole system emerges. The idea of emergent behaviour is closely linked to this.

While an emergent behaviour approach is most closely associated with robotics research, it has had some application at the user interface. The work of Wavish and Graham [7] shows that reactive agents can produce interesting results in other areas. They have created systems with reactive agents as actors where the behaviour of the system emerges from the interactions of the 'actors'. This suggests that when it is possible to identify each important aspect in a system, agents concerned with each might be able to produce complex behaviour through their interaction.

This idea has been applied to the provision of an adaptive interface. Agent driven adaptive interfaces have been developed where the agents unobtrusively observe the user and make inferences based on the user's actions [8]. In our work, the idea of unobtrusively observing the user as a data source has been used but a community of simple agents has been employed where each is concerned with a particular facet of the interaction and the overall behaviour emerges

The classical approach to developing a system such as this is to develop explicit models of the various entities that the adaptive interface needs. Thus user models, task models and system models are developed. If such models were developed then it might seem logical to use a deliberative agent to control such a system, however, there are problems with this approach. The specification of the current system highlights two important issues. The first is that of speed. A large complex user model is less likely to be able to respond quickly to changes. In a system where the user's interaction with the system involves a series of short consultations, an unwieldy model is not the best choice. A more fundamental issue relates to the actual domain. As discussed above, the data that is now available and that this system is designed to display has not been available before. Thus while clinicians have a good idea about what is important, it is very possible that there are patterns and relations that can only be observed when the data now available is examined. Thus a system that can attempt to derive its own organisation for the data is going to be more useful than one where the relationships have to be explicitly described at design time as is the case with a high level model. For these reasons, it was decided to employ a system of reactive agents

5 The Multiagent System in Action

At the interface, there is a series of agents each concerned with a particular aspect of the functionality. Being simple reactive agents they are able to rapidly respond to changes. As noted above however, such simple reactive agents do not usually express complex behaviour. In this work, the agents are provided with the facility to adapt their reaction thresholds over time and by interacting extensively with other agents, produce through emergence a more complex system. From the point of view of the data, by using a series of reactive agents that are each responsible for a particular facet of the data (statistic derived from the data), such as means, upper and lower quartiles, and allowing them to build up relationships with other data agents, the system is able to self organise itself in such a way that it models the patterns in the data.

When attempting to work at a low level and provide adaptivity by carrying out observation at the level of individual actions, the complexity of the task becomes an important issue. Gervasio et al [9] found that when trying to predict the actions that a user carries out to create a schedule in a crisis planner, that by reducing the complexity of the task to be predicted - by abstracting classes of actions from the set of available actions - the accuracy of prediction increased. Of course, this increases the workload for the user, as they are required to provide the specific details to the action predicted. This raises the issue of just how effort should be divided between user and computer in such a mixed initiative system.

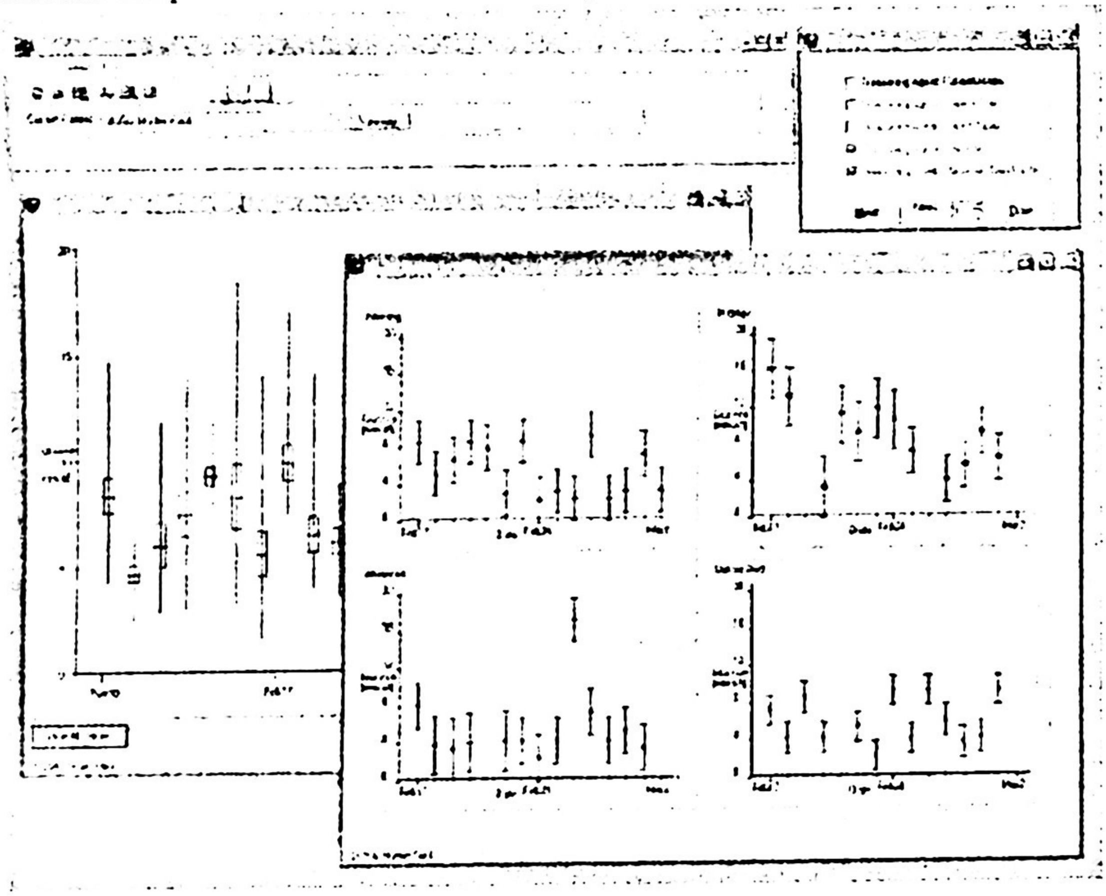

Fig. 1. The Multiagent system in action

The goal in providing adaptivity is to make the user's task easier rather than to actually replace the user. In a mixed initiative system, the user should still maintain control and the system should be simply trying to reduce the complexity of the user's task rather than taking them out of the loop completely. With this in mind, it is perfectly feasible to produce a system that meets the more limited goal of making it easier for the user to obtain the commands they wish to carry out rather than to predict with complete accuracy each command .With this objective, the proposed adaptivity in the system involves a dynamically generated set of options that reflect important aspects in the data available. The system simply provides these options to the user; the user is then free to take or ignore these choices, thus maintaining control. Crucially system generated choices save the user from having to analyze the data to make the choices themselves.

Figure 1 illustrates the interaction from the user's point of view. The interaction follows these stages:

- The user logs in.
- The user selects a patient which causes the patient's dataset ('log-book' record of PDA recorded events) to be loaded, having been either previously or immediately downloaded from the patient's PDA.
- The dataset is analysed (i.e. data agents compute data attributes).
- The user can either select one or more of the system's data visualisations via the system's menu or select a 'summary' – the summary is a prioritised list of the systems visualisations.
- If the user selects the summary, this can be stepped through to display each in turn (the user can select the length of the list from one to the complete list of 16 visualisations).

6 System Architecture

There are two layers to the architecture produced. The interaction layer contains the agents that interact with the data and the user and brings the results of the interactions together to allow decisions to be made. The control layer contains the agents that manage this process. In addition, a blackboard contains a discourse model to record user actions and a domain model to capture relationships between interface actions and dataset attributes. A blackboard is employed to communicate models because the system is using reactive agents that do not have the facility to store large amounts of data.. Figure 2 shows the system architecture and how the two layers of agents operate within the system.

6.1 The Interaction Layer

The main part of the functionality of the agent component is found in the interaction layer. The agents here are responsible for interacting with both the user and the data and combining the information from both to make the decisions about the summary that is to be generated. The task to be completed has two distinct parts. As data arrives, it needs to be analysed for patterns and as the system is used, the interface has to monitor the actions of the user, i.e. choices of visualisations. These are very different tasks. Monitoring the data is a discrete process that is carried out whenever a dataset is loaded whereas monitoring the user is a continuous process. It therefore makes sense when employing a community of agents to employ one set to act in a discrete manner and deal with the dataset while another acts in a continuous manner and deal with the actions of the user. While the data and the interface are monitored separately, each needs to feed into the other. Thus an effective way of combining the information from the data and the interface is needed. To allow this, a third set of agents is required that communicates only with other agents. These bring together the information from the data and interface agents in such a way that the patterns in the data can be used both to drive what is important in the summary and to provide a context for the actions that the user is taking.

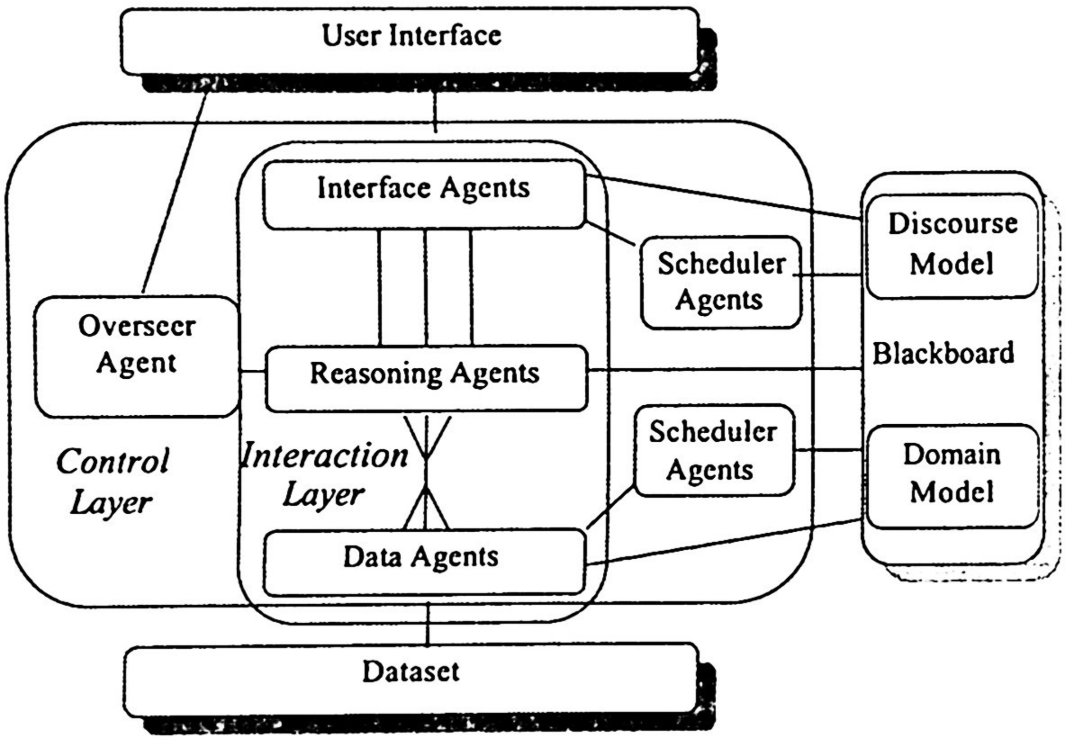

Fig. 2. The Multiagent system architecture

6.1.1 Interface Agents

There is an interface agent associated with each of the relevant interface actions. When the user carries out an action, the relevant interface agent notices this and keeps track of this occurrence. It then looks to see which interface actions preceded it and which follow it. This information is used to update its model about sequential relationships between the action it is associated with and other interface actions. This information can be supplied to the relevant reasoning agent when required. In this way, each interface agent is responsible for building a model of how important its own action is and how it relates to other interface actions.

6.1.2 Data Agents

When a dataset enters the system, some initial calculations are carried out to enable the system to assign attribute values to the dataset. At this time, the data agents will also examine the data. Each data agent is concerned with a particular attribute of the data and looks to see whether the data's value is different from what it normally expects. If the value is beyond a threshold above or below normal, the agent will notify this fact. It will then look to see which other agents have also notified. By taking note of which agents have notified each data agent can build up its own local model of how other data aspects relate to its data attribute. This allows the community of data agents to track correlations and inverse correlations. As with the interface agents, this information can be provided to the reasoning agents when required.

6.1.3 Reasoning Agents

Reasoning agents do not have any direct contact with the users or the data. Instead they combine the data from interface and data agents. Interface agents can determine patterns of behaviour in the actions of the user and data agents can determine patterns in the data. By combining these, the reasoning agents have a fuller picture of the situation. When the data agents are notifying about changes in behaviour, the reasoning agents take note of this. They also take note of when their associated interface action takes place and look for correlations between the use of their interface action and changes in the data. When a summary is requested, each reasoning agent will attempt to make a case for its interface action. It does this by interrogating its associated interface agent about the associated interface action and by interrogating the data agents to decide whether the current data patterns match with the correlations that it has developed. This information allows the reasoning agent to decide whether its interface action's priority in the 'summary' should be amended (up or down). Once each reasoning agent has made an initial case for its interface action, it can check the strength of belief that other reasoning agents have of their actions and use the information from the interface agent to decide whether to update its belief because of the level of belief in other interface actions to which there appears to be a correlation.

6.2 Control Level

If one allows a community of agents to alter their behaviour then there is a danger that the system's behaviour will migrate away from what is required. To address this, the control layer has final control over any changes made and so can ensure that the actions of the agents stay within sensible bounds. An overseer agent is responsible for this. In a community of agents there is also the issue of coordinating the communication that takes place. Again the control layer takes responsibility for this by providing scheduler agents to mediate the interaction where required.

6.2.1 Overseer Agent

The role of the overseer is to keep the system within sensible bounds. This is something that could have been achieved by coding limits into the interaction layer agents themselves or by providing an overall arbiter. It was decided that a central arbiter was a more sensible choice as the control was being applied to the overall decisions being made rather than the individual parts of the process. In a system that utilizes emergence, trying to effectively constrain the system by considering it at an individual agent level would be a difficult task without affecting the ability of the agents to effectively collaborate. Thus providing the control at the community level was implemented as a more transparent way or providing the necessary functionality. The overseer agent responds to the user's request for a 'summary' by interacting with the reasoning agents to compile a prioritised list of visualisations.

6.2.2 Scheduler Agents

Facilitating the interchange of information in the system are the scheduler agents. While the individual agents can in many cases interact sufficiently without outside interference, it is sometimes the case that an outside entity is required to control part of the data flow. For this reason, scheduler agents are responsible for mediating various parts of the interaction. An example of the use of a scheduler agent is when an interface agent is activated and adds the fact that its action has been chosen into the discourse model. It can then look back over the discourse model to see which actions were carried out previously and use this information to update its model of action sequences. Each agent also needs to know which actions happen after their own action to allow for sequences going forward in time. To provide for this, the interface agents could remain watching the discourse model and take notice of every time a new action was added. This means that a number of agents will be spending time simply accessing the discourse model to see what was happening. However we have seen that it is more efficient to have a single scheduler agent tasked with watching the discourse model and keeping track of which agents' actions have occurred and taking responsibility for sending information to each agent when further actions occur.

7 Discussion

The system has been informally evaluated in a clinical setting in consultations between health care professionals and diabetic patients. Three health care professionals used the system and reported that they found the information provided useful. They also reported that they did not find the system intrusive, one clinician reporting that he had viewed the information in one session with a patient but had not found the information useful in this particular instance, importantly he said that had not felt that it had wasted his time asking for the information. All users reported that the system delivered the information in a timely manner and did not detract in any way from their interaction with their patient. In addition, they stated that if the system were made available for their use they would use it in the majority of the sessions they had with patients.

When creating a system such as this, it is very important that the system is able to make accurate judgments as to what the user wants. As discussed, this has been achieved through observing the user choices in response to the data available. In itself, this might seem a logical start but a feedback mechanism is needed to allow it to appraise effectiveness. Using agents to check how the user utilizes the summaries provided achieves this by analysing choices from the summary in the same way that the choices from the main interface are analysed. This allows the system to determine how accurate the summary is. Just as choices made at the interface provide data for adaptation, advice followed, and indeed advice not followed provide the feedback loop to keep the system in check. The control layer agents keep the actions of the agents within sensible bounds but this in itself does not prevent the agents making incorrect choices. Providing an effective feedback mechanism goes towards addressing this problem.

Currently, the feedback mechanism uses whether or not the choices from the proposed summary were selected to provide the information. If the user selects something that is available from the summary directly from the interface, it does not consider the ramifications of this in terms of acceptance of the summary. An analysis of this type involves a much deeper study of the user interface issues surrounding the use of a system of this type. While worthwhile, it is beyond the scope of the current work.

A second issue when considering accuracy of decisions made is the quality of the information upon which the decision was made. Many adaptive systems do not have a strong model of exactly what the user is trying to achieve. A good example of this is the work of Korvemaker and Greiner [10] where they were trying to predict Unix commands. They demonstrated that a system could, in the case of Unix command prediction, attempt to predict the pattern of commands that is to be repeated. However, this does not mean that one can necessarily have any understanding of what the user is trying to achieve. Without this knowledge, the task of prediction is, as shown, very difficult. In the case of web page prediction, the various systems can attempt to match keywords in the available pages to pick their recommendations. This could be seen as starting to move towards trying to understand what the user is trying to achieve and perhaps make it easier to then predict what they require from the system. The agents have some idea about what each page is concerned with and user choices allow them to determine what type of page is of interest. Of course the use of keywords is not perfect. Unfortunately HTML based pages do not allow for much else. With more widespread use of the various XML related technologies [10], we could perhaps be moving towards a situation where much richer knowledge about what the user is attempting to retrieve is available. This could be used to enable agents to produce better-informed choices about what is required.

In our system, a conscious effort is made to try to make the most of the available information from both data streams. The actions of the user are considered alongside patterns in the data. This allows the system to not only observe patterns in what the user does but relate these to the data being considered thus placing these actions within a context.

By using a community of simple agents that communicated with each other, it was possible to consider the actions at the interface and the patterns in the data separately while still having a mechanism in place in the form of the reasoning agents, which allowed these two analyses to be combined to provide the final decisions.

8 Conclusion

This work demonstrates that the use of emergent behaviour in a community of agents provides a means of driving a self adaptive system. To achieve this using a conventional approach would have required the construction of a far more complex system with the various high level models that such an approach entails. We have shown that a group of agents working at finding patterns can combine together through their interactions to produce a working system. With the relationships between the various patterns in the data and user actions implicitly modelled, one can, at least in some cases, avoid the need for complex high level models.

Hence the agent-based system adapts to the clinician's usage, rather than to his or her implicit directions, in order to provide the clinician with high quality information in a form that is pertinent to their enquiries yet unobtrusive in use.

9 Acknowledgement

The authors wish to acknowledge the collaboration of the Diabetes Research Laboratories and Diabetes Trials Unit of Oxford University.

References

1. Holman R.R., Smale, A.D., Pemberton, E., Riefflin, A., Nealon J.L. Randomised controlled pilot trail of a hand-held patient-oriented insulin regimen optimizer. In: Journal of Medical Informatics, 21:4, (1996), 317-326
2. Diabetes Control & Complications Trial Research Group: The Effect of Intensive Treatment of Diabetes on the Development and Progression of Long-Term Complications in Insulin-Dependent Diabetes Mellitus. New England Journal of Medicine, 329(14), (1993)
3. UK Prospective Diabetes Study (UKPDS) Group. Intensive blood glucose control with sulphonylureas or insulin compared with conventional treatment and risk of complications in patients with type 2 diabetes. Lancet 352, (1998), 837-853
4. Wooldridge, M. An Introduction to MultiAgent Systems. John Wiley, (2002), 258-259
5. Muller, J.P. Architectures and applications of intelligent agents: a survey. In: Knowledge Engineering Review, Vol. 13(4) (1998), 353-380
6. Brooks, R.A. Intelligence without Representation, Art. Intell. 47 (1991), 139-159
7. Nwana, H., Ndumu, D. A perspective on software agents research. In: Knowledge Engineering Review Vol 14(2), (1989), 125-142
8. Goecks, J., Shavlik, J. Learning User's interests by Unobtrusively Observing their Normal Behaviour - IUI 2000. In: Proc. Int. Conf. On Intelligent User Interfaces, (2000), 129-133
9. Gervasio, M.T., Iba, W., Langley, P. Learning to Predict User Operations for Adaptive Scheduling Proc. of the 15th Nat. Conf. on AI (1998), 721-726
10. Korvemaker, B., Greiner, R. Predicting Unix Command Lines: Adjusting to User Patterns. In: Proc. 17th Nat. Conf on A I (2000), 230-235
11. Berger, G., Ruddock, N. Is XML the Missing Link in Raising Browsers to a Higher Intelligence, XML99 (1999)