Clinical user interfaces that learn from experience

Svetla Gadzhanova
PhD Student
Jan Stanek MD
Lecturer in Computer & Information Science
James Warren BSc PhD
Associate Professor in Computer & Information Science/Director
Advanced Computing Research Centre, University of South Australia, Mawson Lakes, Australia

ABSTRACT

Clinical data entry is one key to success in health information systems that is not a matter of technology alone, but of appropriateness and usability of design. We review the technology of adaptive user interfaces and learning agents. In these technologies we see the potential to improve the usability of general practice clinical workstations through machine-learnt adaptation to the user, the patient and the specific situation. Use of intelligent split menus that adapt based on past clinical encounters is one specific adaptive interface method that has shown potential by simulation. We are undertaking research in ‘expert in the loop’ use of data mining for iterative refinement of clinical workstation adaptation with an eye to significantly improving general practice data entry quality.

Keywords: adaptive user interfaces, clinical workstation, data entry, learning agents

Introduction

Extraordinary efforts have been undertaken in developing the technical infrastructure of health information systems (networks, workstations and practice management software), electronic record structures and communication standards (for example, GEHR and HL7) and electronic guideline implementations (such as PRODIGY). However, as the technical components become ubiquitous, information entry – feeding data to the systems for processing – appears to be an increasingly important limiting factor. Designing user interfaces able to cope with the requirement to enter accurate, valid and relevant information, while remaining highly flexible and operating in a non-disruptive way, is still an unsolved challenge. One line of investigation to meet this challenge is research in adaptive or intelligent user interfaces.

General practice offers special challenges for capture of the electronic medical record (EMR). A broad spectrum of patients is seen and physicians use many different practice patterns. Moreover, at the general practice level, there is often less specificity and more uncertainty as compared to the specialist or hospital setting. The number of possible diagnoses is large and the data entry task must be done in brief time, generally without the support of specialised data entry or medical records staff.

To support the general practitioner (GP), an intelligent interface should ideally act as a well-trained co-worker. As an example: in an operating theatre several surgeons are able to perform the same type of operation. The instrumentation nurse (as an interface) should be able to prepare to support a particular surgeon (including his particular style of performing the operation) to match the condition of the particular patient, as well as to change behaviour according to immediate changes during the operation. In terms of user interface this means that the behaviour of the interface should take into account:

- user operating style/habits
- specific features of the patient (such as sex, age, diagnoses and allergies)
- current situation (for example, different behaviour is expected when an asthmatic patient comes to see his GP for a scheduled routine review, as opposed to coming with acute bronchitis).
There are two basic approaches to human–computer interaction:

- **Direct-manipulation (DM) graphical interfaces** – display visual representations of physical or conceptual objects and allow the user to issue commands that change the state of the objects. In a DM interface, the changes to the display state are more or less one-to-one, with the commands explicitly invoked by the user.

- **Indirect management interfaces (agents)** – the user is engaged in a co-operative process in which both human and computer agents initiate communication, monitor events and perform tasks.

'Software agents' are programs that assist users in a range of different ways: they can perform tasks, make suggestions or act on the user’s behalf; they might train or teach the user; they can help different users to collaborate; they may monitor events and procedures.

'Autonomous agents' are agents that take action without user intervention and can operate in parallel with the user. The agent may remain active long after the user issued other commands or has even turned the computer off.

Agents that are employed in the interface and delegated certain computer-based tasks are termed 'interface agents'. These agents actively assist a user in operating an interactive interface (the best examples are intelligent tutoring systems and context-sensitive help systems). Maes suggests several approaches in building interface agents.

- **User-programmed interface agent** – consists of a collection of user-programmed rules for processing information related to a particular task. This is the most straightforward approach, but relies heavily on the user’s programming skills, which is usually an unrealistic expectation.

- **Knowledge-based approach (AI engineered)** – created by traditional knowledge-engineering approaches from the artificial intelligence (AI) field. Extensive domain-specific background knowledge about the application and the user must be encoded. One of the problems here is that a huge amount of work is required from the knowledge engineer and that knowledge is fixed once and for all (it cannot be customised to individual user habits and preferences without re-engineering).

- **Machine learning approach for building interface agents** – the interface agent can ‘program itself’ (that is, it can acquire the knowledge it needs to assist its user). The agent is given a minimum of background knowledge and it learns appropriate ‘behaviour’ from the user and other agents. During the learning process, the interface agent can become gradually more helpful and competent. The agent might be able to give ‘explanations’ for its reasoning and behaviour. Such a ‘learning agent’ acquires its competence from four different sources: observing and imitating the user, learning from direct or indirect user feedback, learning from examples given explicitly by the user, and asking for advice from other agents with similar tasks. Learning agents have been demonstrated for meeting scheduling and email prioritisation.

Horvitz suggests human–computer interaction in terms of a combination of DM and interface agents: a 'mixed-initiative user interface'. This interface enables intelligent agents and users to collaborate efficiently to achieve the user's goals. Because the agent might be uncertain about the user's goals, Horvitz applies a model of the expected utility of an action versus inaction in terms of the probability, $P$, that a user desires the action in light of the evidence in the current context. There is a threshold probability, $P^*$, that is the break-even point of the expected utility of taking autonomous action to assist the user and the expected utility of not taking autonomous action; that is, it is best for the system to take action if the probability of a goal is greater than $P^*$ and to refrain from acting if the probability is less than $P^*$. Horvitz also considers the case when an agent initiates a dialogue about a user’s goal for intermediate values of $P$.

LookOut is a variant of Microsoft Outlook that illustrates the potential for efficient mixed-initiative interaction. LookOut identifies email messages that concern a meeting invitation; the system assists the users with reviewing their calendar and with making the candidate appointments in a way that suits their preferences.

An 'adaptive user interface' is an interactive software system that improves its ability to interact with a user based on partial experience with that user. Such an interface adapts to the user rather than the user adapting to the system. Norcio suggests two ways that the system can be adaptive:

- **to leave the interface in a form that enables modification by the user (the interface may be modified by a computer specialist, a trained user or any user)**
- **dynamically changing interface that adapts by itself with respect to the particular user and current context**.

Models of the user, task, system and interaction form the basis of this latter type of system-initiated adaptation. The purpose of a ‘user model’ is to capture and include individual characteristics of the user – his/her goals and plans, program-specific...
knowledge and preferences – as relevant to the interaction. It is particularly common to classify users by ‘stereotypes’ – a set of user categories (such as ‘novice’ or ‘expert’) – either explicitly based on user responses to questions or implicitly based on user behaviour. ‘Task modelling’ may be complementary or alternative to user modelling and is concerned with the system leveraging a model of the task/domain (across users) to achieve better task performance.

Use of ‘menus’ is a cornerstone of DM. Since menus make options visible, they speed up performance and learning. An interesting menu variation is the ‘split menu’, which has top and bottom sections. Designers or individual users may place frequently selected items in the top section and infrequently selected items are organised alphabetically in the bottom section. By moving these frequently used items to the top of the menu, users are able to locate and select them more rapidly. As the length of the menu increases, the potential benefits of split menus also increase. Sears and Shneiderman find that split menus can be faster and preferred by users as compared to conventional (alphabetical) or fully frequency-sorted menus.

Towards intelligent clinical workstation interfaces

Our research centres on the integration of learning agents into user-controlled DM interface to support GPs in high-quality and efficient practice with good EMR data entry. The Mars Medical Assistant (MMA) exemplifies use of a combination of user, task and situation models to build an ‘assistant’ aiding a clinician by providing appropriate information and suggestions for the current context, based on stereotypes of users, tasks and situations. As an alternative to an explicit assistance system, ‘intelligent’ split menus show potential for more subtle use of domain knowledge (particularly machine-learnt domain knowledge) in clinical data entry. In echocardiography, Canfield used statistical associations between word categories in specialist narratives to prime the frequency ordering of the split menus. Canfield showed (by simulation) that menu entry with these intelligent split menus required between two and five times less effort than menus arranged alphabetically. Warren extends the idea of intelligent split menus to general practice data entry. He uses Bayesian estimation based on a large database of GP records (113 000 encounters). His work focuses on generating ‘hotlists’ (dynamic top sections of split menus) of likely diagnoses in light of the patient’s RFE (reason for encounter – symptoms, complaints, checkups and so on). Hotlists are shown (by simulation) to be time saving compared to other user interface methods for selection of primary care data codes. There is little question that significant knowledge of practice patterns can be engineered into assistants or machine learnt from large sets of EMRs – the challenge is in integration of this knowledge with effective clinical workstations for use in day-to-day general practice consultations.

We are considering three essential modelling components for an intelligent clinical workstation interface:

- user (physician) model – the preferred approach of a given GP to specific situations
- case model – how a given situation should be handled in general
- patient model – patient history and preferences.

Obviously, large knowledge-engineering projects such as PRODIGY provide case models of evidence-based practice for a considerable range of common situations. Our focus is more on the use of machine learning/data mining to establish normative user, case and patient models as a complementary resource to knowledge engineering.

We aim to balance several persistent and somewhat conflicting demands:

- common practice is not always best practice – any GP can be caught out by the pace of change, or simply make an error; therefore continuous decision support is desirable
- externally imposed guidelines can be inflexible, lack tailoring to local conditions and fail to account for individual patient complexity and preference – therefore support for local adoption strategies is desirable
- tracking each decision along a guideline pathway consumes valuable time – therefore automated association of user actions with guidelines is desirable.

We are investigating an ‘expert in the loop’ approach where EMRs are data mined and patterns are reviewed by GPs of a participating practice, working across major disease areas first (for example, treatment of hypertension, diabetes, depression and so on). Innocuous practice patterns are used to fuel split menus, to facilitate data entry and show system responsiveness (or, to be anthropomorphic, ‘understanding’) of the nature of user preference, the case at hand and patient history. Patterns that are not compliant with best practice have alert dialogues attached to them, generally including the invocation of a guideline and explanation of the specific point of non-compliance.

These split menu and alert features are being embedded in the local practice management software, remaining within a user-controlled DM paradigm of operation but mixing in adaptive behaviour (for instance, in terms of the menu hotlists) and agent-like system initiatives (such as initiating alert dialogues).
We are working with the University of Adelaide Family Practice, Whyalla, towards formulation of an acceptable mixed-initiative system on the above paradigm, based on iterative application of data mining and expert assessment. Our first success factor is simple improvement of the quality of the EMR, including rate of problem coding and apparent compliance of treatment codes with problem codes. However, we believe that a further effect will be improved adherence to localised practice guidelines.

Conclusion

User interfaces are recognised as being an important issue in health information systems. We are undertaking research to bring learning agents into adaptive user interfaces for improved GP data entry. The goal is to maximise available knowledge and experience in the interface while maintaining user control both in the operation of the clinical workstation and in terms of its knowledge content.

REFERENCES

1 Good Electronic Health Record (GEHR): www.gehr.org
2 Health Level 7 (HL7): www.hl7.org
3 PRODIGY: www.prodigy.nhs.uk

ADDRESS FOR CORRESPONDENCE

Jan Stanek
University of South Australia
School of Computer & Information Sciences
Advanced Computing Research Centre
Mawson Lakes Campus
Mawson Lakes SA 5095
Australia
Tel: +61 8 8302 3460
Fax: +61 8 8302 3381
Email: jan.stanek@unisa.edu.au

Accepted July 2002