Towards a System to Improve Administrative Processes for Front-Line Academic Staff

Roger Tagg and Madan Kumar Narayana Murthy

School of Computer & Information Science
University of South Australia
Mawson Lakes,5095, Australia

Roger.Tagg@unisa.edu.au Madan.Murthy@unisa.edu.au

Abstract

Current economic pressures are causing severe problems for many enterprises in maintaining service standards with shrinking headcounts. Groupware, Workflow and Agent technologies have been widely advocated as a solution, but there are few reported success stories. The project described in this paper addresses the case of running large undergraduate courses. A preliminary vision of a possible integrated administrative support system is presented, and the future activities necessary to advance such a vision are outlined.

Keywords Administrative Applications, User Interface Design, Software Agents

1 Introduction

Severe financial constraints in recent years have forced many enterprises, including universities, to try to achieve more and more with fewer and fewer resources, including staff. Staff members have become progressively less able to cope with the increased workloads, while consumers have experienced a gradual deterioration in the quality of the actual service for which the enterprise exists (e.g. teaching and supervision, health care etc).

A few years ago it was possible for a single academic to run a course with the help of the university's administration. Nowadays a team approach is necessary, including senior, junior, guest and contract lecturers, moderators and technical and administrative support staff. This team has to share both data and process knowledge.

Regarding data, the student's record is the primary document. But there are also assignment mark sheets, spreadsheets, lecture notes, study guides, documents on the course website, documents for tutors, computer files for practicals (e.g. provided data and source programs).

From the process viewpoint, teams – especially where staff turnover is high - need more help in carrying out the processes than when the course is run by a single lecturer.

The authors of this paper are a senior lecturer and a recent Masterate graduate in a school of computing and information science. It is therefore appropriate to consider how we can take some of our own medicine. Our school is involved in research into groupware, workflow and intelligent agents, which have been proposed as a means of improving the effectiveness of workers in teams and enterprises, for example [1]. We develop theoretical models, and talk about potential benefits, but have yet to apply these in practice.

At the same time, university computer developments such as ERP systems for general student administration, together with web interfaces, have improved both general administration and student accessibility. But they have not helped improve the processes carried out by front-line academics. A recent project at the University of Queensland has developed a system, FlexEl [2], that uses a workflow engine to coordinate a self-paced course; however this does not address the wider aspects of team teaching administration support.

2 Case study: running a large undergraduate course

Our school currently runs 17 core undergraduate courses that the majority of students must take. Third year numbers at the main campus are over 200, with double this in the second year and triple in year one. Teaching is offered at 2 or 3 campuses in South

Proceedings of the 7th Australian Document Computing Symposium,
Sydney, Australia, December 16, 2002

Australia, and there are sometimes external students as well. Courses following the same syllabus and taking equivalent assessments are offered at an increasing number of sites in South East Asia. Some courses additionally run evening streams.

Our current project has been started with the dual intention of improving support for transient staff working in team teaching and providing a test bed for workflow and agent technologies. The authors have started to create models describing the current data, role and process structures.

The *organization model* describes the reporting structure of the various roles involved in team teaching, including course coordinator, moderator, lecturers, tutors etc and also administrative staff.

The *data model* describes the artifacts involved in team teaching, including lectures, tutorials, practicals, self-study modules, assignments, tests and the documents that relate to these artifacts. Data can be related to the whole course offering, or to individual students or project groups.

The *process models* are divided into three main groups: running a course, running a degree program and running a project or thesis. In each process model there is a 4-level hierarchy:

1. run the course, program or thesis over its lifetime
2. run it in any one academic year
3. run one offering (i.e. with a given start date, at a given campus)
4. run one activity within an offering (e.g. assessment, on-campus module).

3 Preliminary design of a prototype system

As the main motivation for this project is to improve the efficiency and effectiveness of staff, the user interface has been an early part of the design. Figure 1 shows a large window that would be offered to the user on startup. This is assumed to occupy all except the outermost edges of the user's screen, where essential desktop icons could remain visible.

The center of this workspace window would be available to whatever applications the user starts up, either explicitly through the Operating System or implicitly by clicking the buttons round the edge of the window. Some of these buttons support a "drop box" functionality; the actual semantics of dragging and dropping an object onto such a button will be different for each button.

The buttons are arranged in groups, some of which are common to all users, but others of which only apply to those users who have responsibilities in specific courses and degree programs.

A key concept in this system is *context*. This can be established by one of the following means:

- the area of responsibility the user is currently working on (see the highlight on course 123456)
- the function the user is currently carrying out (e.g. the topmost window)
- explicitly by the user dragging and dropping the window he/she is working with into one of the drop boxes.

The W*orkflow design* tool allows the user to tailor the workflow, either for the current case or "from now on". A currently live work item can be dropped into either the *New case* or *Action this case* box. The user can review the progress of all cases within a workflow type, with an optional selection criterion (e.g. all students doing the BInfTech(SwEng) program).

The *Tasks* and *Appointments* areas have drop boxes where e-mails and other documents can be dropped. An agent will scan the dropped document and deduce any tasks or appointments, not just for the current user, but for other members of the teaching team.

The main *E-mail* drop box is an intelligent outbox agent. The agent should be able to format the e-mail for the mailer software being used, and to deduce the email address if possible.

Other general buttons include *Upload* and *Download* for a mobile computing device; *General trash* (an intelligent archive) and *General filing*. This is an intelligent drop box that works out where the document should be filed, either from context (e.g. text within the document itself, current live task) or by prompting the user.

The *University student records* system drop box is an intelligent agent that sends valid transactions to this server system. One common use would be to drop the course marks into it at the end of each teaching period. Students would be matched by keys. Data columns could be matched by name, or the columns highlighted in the source spreadsheet.

In the bottom left corner *My avatar/avataress* allows for possible future use of generated voice prompts and responses.

The remaining groups each represent an area of this user's main responsibility, such as a course. The range of buttons might vary, but for most courses there would be buttons for *Teaching team*, *Planning* and *Teaching resources*; and drop boxes for the *Course website*, *Course filing*, *Classes and Groups* and *Course marks*.

The *Course website* drop box would work out a suitable URL (Universal Resource Locator) for a page where the dropped document would be stored, ensure that valid links were built to this URL, and arrange file transfer to the live web server. Also, a planned course offering could be dropped in the *Course website* drop box to invoke creation of a new subdirectory for the new offering.

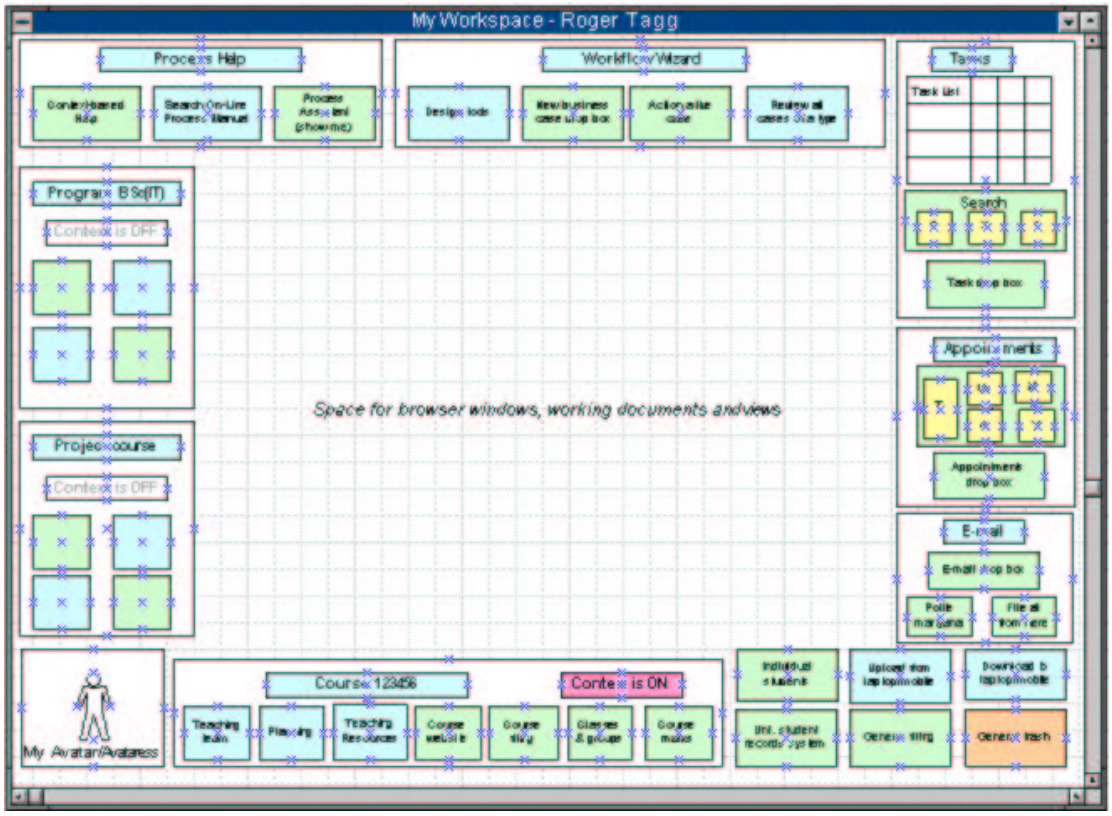

Figure 1: Preliminary Design Layout for the User Interface for a Team Teaching Support System.

The *Course filing* drop box would behave similarly to the general filing one, but would attempt to store documents in their correct place within the folder structure being maintained for the course.

Classes and Groups is used to maintain student allocation to tutorials, practicals and project groups. If the *University student records system* supports class level enrolment, initial allocation can be set up by drag and drop. Formation of project groups - and entering of subsequent changes - can be assisted by dropping student details from *Individual students*

The *Course marks* drop box supports actions relating to the spreadsheets that contain the history and marks of each student on each offering of the course. The main actions would be as follows:

- Drag and drop the *University student records system* onto *Course marks* to download student enrolments, both initially and to refresh the class list
- Generate marking spreadsheets (or other forms) for the markers of assessments, and send these out as attachments to emails
- Drop in the markers' completed spreadsheets when they are emailed back
- Finally, drop the *Course marks* drop box itself onto the *University student records system* at the end of the semester to register the final grades.

4 A planned software architecture

Figure 2 shows the client and server components, together with various other systems with which the user works (either by choice or by decree). Some specific technologies planned for the system are discussed below.

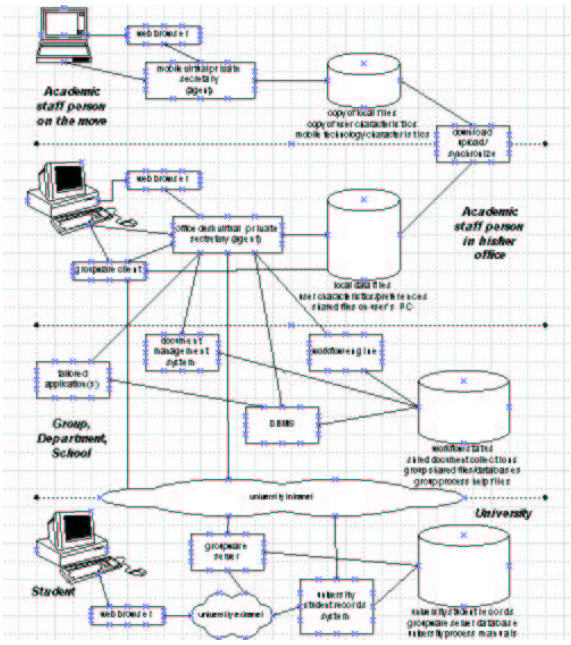

Figure 2: A planned software architecture

We plan to make use of a collection of agents (or components) of various degrees of intelligence. Some will capture and utilize the knowledge of what the users do frequently, and will fit in with their preferred formats and ways of working.

Others will service the drop boxes and buttons. In some cases these agents will provide intelligent front-ends for typical groupware functions such as task lists and calendars, which may reside in a package such as Microsoft Exchange or Lotus Notes. We have christened the aggregation of all our client-side agents as the "virtual private secretary".

On the server side we plan to include a workflow server and a shared process help system.

With regard to the workflow component, totally predefined processes are neither practicable nor acceptable in this application. Therefore the total process support system has to be a mixture of services, ranging from ad hoc requests for process help, through workflow control applied at a user's request, to predefined workflow with ad hoc exceptions and routing changes. A flexible workflow software approach is therefore a requirement.

We plan to adopt, where possible, techniques that reduce keyboard data entry by teaching team staff. Examples are bar coding, character recognition, telephone enrolment by students – and in the future, voice input. Wherever we can, we want to use data that already exists somewhere on the university's network. We have already mentioned the re-use of course mark spreadsheets instead of re-inputting marks into the student record system. Another opportunity arises when entering information about persons. Many people can be identified by data held in the staff or student records systems, and by such things as personal web page URLs.

Because of the need to link the proposed system with other existing or planned IT systems, there will also be a need for intelligent agents to resolve unclear matching with data from other systems, databases and files. These agents would make use of database schemas, XML(Extensible Markup Language) DTDs (Data Type Definitions) and ontologies.

It is envisaged that the proposed system will be made up of a number of autonomous agents, rather than a monolithic client or server. Components (and possibly third-party web services) will be capable of being plugged out and in when justified.

5 Current and future work, and conclusions

We have started modelling the existing processes – most of which are not documented at all - and are building up a process inventory. As the team teaching approach is still being implemented, these processes are not settled. We will next be seeking feedback from our colleagues on the correctness of our models and usage scenarios.

We have also started building a simple system to record receipt of assignments, based on the use of bar code readers to scan assignment cover pages that contain the student or project group's identity, as well as the course and assignment number, in a bar code font. This system is due to operate live from the beginning of the year 2003.

We are about to start an evaluation of the Chameleon flexible workflow engine from DSTC (Distributed Systems Technology Centre, based in Brisbane, Queensland) [3]. Chameleon has been developed from the workflow engine that was developed for FlexEl [2]. The corresponding modelling tool for Chameleon is FlowMake. We are already using the Adonis [4] business process management tool to chart some of our processes, and are building a translator from Adonis to FlowMake.

In 2003. we have scheduled a project to investigate the use of software agents to support our proposed system. At this stage we will carry out a full literature search and evaluate what existing agents we can utilize. We will then prototype parts of the proposed system and seek further feedback.

In conclusion, groupware, workflow and intelligent agents are important research areas in document management and computer science. But theory has got a long way ahead of practice. This project aims to utilize these technologies in a pragmatic way on our own working environment. It is significant that our Vice Chancellor has recently nominated efficiency and workloads as two of the most important issues to be tackled in 2003!

References

[1] Chaffey, D., Groupware, Workflow and Internets, Digital Press, 1998

[2] Distributed Systems Technology Centre, Annual Report, 2000. Available from http://www.dstc.edu.au/Publications/Annual_Report/AR00.pdf

[3] Distributed Systems Technology Centre, Praxis Project Website, http://www.dstc.edu.au/praxis/

[4] BOC Ltd. Website. http://www.boc-eu.com/english/index.shtml