Resource-Centric Worklist Visualisation

Ross Brown[1] and Hye-young Paik[2]

[1] Faculty of Information Technology,
Queensland University of Technology, Brisbane, Australia
[2] School of Computer Science and Engineering,
University of New South Wales, Sydney, Australia
r.brown@qut.edu.au, hpaik@cse.unsw.edu.au

Abstract. Business process management, and in particular workflow management, are a major area of ICT research. At present no coherent approach has been developed to address the problem of workflow visualisation to aid workers in the process of task prioritisation. In this paper we describe the development of a new, coherent approach to worklist visualisation, via analysis and development of a resource-centric view of the worklist information. We then derive appropriate visualisations for worklists and the relevant resources to aid worker in decision making. A worklist visualisation system has been implemented as an extension to an open-source workflow system, YAWL (Yet Another Workflow Language).

1 Introduction

Visualisation techniques offer powerful tools for understanding data and processes within complex systems. However, visualisation in the area of Business Process Management (BPM), and in particular workflow management systems lags behind the state of the art in other areas such as medicine, engineering and mining [1]. A recent Gartner's report suggests that many business organisations consider BPM to be a fundamental driver of business innovation. This is demonstrated by the large amount of money and expert resources invested in business process modelling, analysis and roll-out of the models. Workflow Management Systems (WfMS) play a vital role in BPM in that the business process models are implemented and executed through a WfMS, which routes and dispatches the tasks defined in a model to the individual workers[1]. The result of "routing" tasks is presented to the workers as a *worklist*. A worklist can be understood as a "to-do" list of tasks that the workers need to carry out in order to complete the process defined by the model.

The success of business process models depends on communicating them to the model consumers effectively. However, modern workflow systems have largely overlooked the needs of the workers of understanding their given tasks in the manner that would help manage them efficiently. For example, it is quite common that the workers would have questions such as "how urgent is this

---

[1] The workers are the "consumers" of the model who will carry out the tasks. In this paper, we use the terms "workers" and "model consumers" interchangeably.

task?", "who else can do the task?", "where do you have to go to carry out the task? (eg., where is this meeting room B809)", "do I have enough resources? (eg., are there enough chairs for 20 people in the meeting room B809)", etc.

A typical representation of a worklist includes a list of tasks with short textual descriptions, and/or attachments (eg., email, document forms, etc). It, however, does not include any support (context) information about the tasks that may assist the worker in planning the tasks. At any point in time, a given worker may be involved in many workflows and may thus be presented with a large to-do list. The worker needs to have available tools to help them decide which would be the "best" task to undertake next.

We believe that visualisation techniques can be applied to many areas of BPM due to their previous use in application domains that support decision making processes. In decision support systems, information is typically provided to enable the user to be adequately informed to the direction to be taken for a particular scenario. This applies to all levels of business systems, and to BPM as a whole.

For the purpose of this paper, we limit the scope of the work to the area of workflow management, in particular, managing worklists. We apply a visualisation technique to provide workers with information about the context of a task, in order to improve their understanding of the process models and the communication of such models between the model designers and the consumers. The visual information is designed to help workers make decisions in managing worklists (eg., accepting, postponing, delegating, or rejecting tasks).

In this paper, we propose a generic visualisation framework that is used to provide support (context) information about the tasks in a worklist. Our contributions are three folds:

- An analysis of the decision making process in managing workflow tasks, especially in relation to the resources available to the worker
- A novel and generic visualisation technique for worklists
- The implementation of the framework as a proof of concept

The rest of the paper is organised as follows: Section 2 investigates the state of the art in worklist visualisation. Section 3 details the development of a resource centric approach to the management of worklists. Section 4 explains the mapping of important worklist resources to appropriate visualisation techniques to aid the process of task selection by workers. Section 5 describes the general approach to using these visualisations in workflow systems. Section 6 details the implementation of a visualisation system incorporated into a workflow system. The paper then concludes with a discussion of future work.

2 Related Work

Computerised data visualisation is a broad field that has its beginning in pictorial representations used in pre information technology timesr [2]. Today it has developed to the point of being one of the main areas of application for

computer graphics, and is a powerful tool for the analysis and presentation of data [1]. Many areas in science and mathematics have benefited from the exploitation of modern computing power in data exploration. Business experts are now using advanced computerised visualisations to analyse multi-dimensional business data and processes. We believe that there is a very good opportunity to apply these leading edge techniques to the visualisation of business processes models and their execution.

The present state of play for the visualisation of BPM systems uses process state tables [3], simple icon based 2D visualisations [4] and more complex 2D visualisations [5]. Some have explored the use of 3D extensions to 2D diagrams [6] to 3D representations using such techniques as Cone Trees [7] to full virtual reality implementations for distributed interaction with detailed process models [8]. Some used abstract representations such as Self-Organising Maps (SOM) [9].

A body of research has been carried out into visualisation of business process data and is collectively known as BizViz (Business Visualisation) [10–12]. Bizviz consists of the visualisation of data alone, and not business process information. At present, it is ad hoc in nature, without a rigorous assessment of a number of the following factors: potential valid visualisation techniques from other fields and business requirements for such visualisations.

While there has been evidence of research into user requirements for business process modelling [13, 5], much work still remains with regards to the following:

- Data gathering for requirements analysis, the current research is often tied to software implementations which restrict creative solutions;
- No real evidence of systematic analysis of sophisticated 2D and 3D visualisation techniques for use in complex business process models;
- Abstract representational techniques are often ignored despite their power in representing multi-dimensional data that occurs in business systems;
- Application domain information is not factored into the representations;
- No assessment of visualisation effectiveness via real case studies.

What is needed is a thorough data gathering-based analysis of user requirements for the visualisation of business processes, and the analysis of the many 2D/3D techniques and visualisation wisdom for such representations. In particular, the area of concurrent process visualisation [14] is expected to provide many useful visualisation techniques. Furthermore, there is a need to provide an approach to visualisation of business processes that accounts for domain specific factors in their representations. Such a visualisation approach needs to allow for both the designers [15] and the users of the business process model [3, 13], as both these people have different requirements for visualisations, with regards to design, analysis, and usage tasks.

What these other workflow visualisation techniques lack is a focus on supporting information to assist the worker in managing the tasks in a worklist. Each of the techniques provides a presentation of the worklist that is rudimentary in nature, lacking any support information for the main task required by a workflow system; deciding to accept, delegate, suspend a presented task. We believe this should be the main reason for such workflow visualisations, and that an analysis

of this choice process and derivation of appropriate visualisation techniques is required to support this process. Analysis of such requirement is best taken from a resource oriented point of view [16], as the available resources in an organisation control the acceptance or rejection of the task[2] into the active worklist of the worker. We now proceed to analyse this worklist management problem from a resource perspective, in order to derive appropriate worklist visualisations.

3   Resource-centric views of worklists

In this section, we introduce a notion of resource-centric views for worklists. It should be noted that we use the term resources specifically to refer to any work environment element or context that may be considered when workers make decisions in managing their tasks.

3.1   Example scenarios

To illustrate our concept, we use the following two simple workflows as running examples throughout the paper[3]. The first case represents a process given to a university student who needs to obtain proper access to the university facilities. According to Fig. 1, the student will have to visit several service centres situated in different locations in the campus to obtain a computer account and a student card. The second case describes a stocktaking process given to an asset management officer who has to record all computer assets managed by a company. Fig. 2 describes that, after stocktaking is announced, the officer has to plan and schedule field trips to various sites to physically locate an asset and record the asset number using a barcode scanner. This process will continue until all the sites have been visited.

3.2   Analysis of resources for worklists

For the workers to be able to carry out each task, some context information may be required. For example, the student, from the first scenario, may want to know where the buildings are located in the university campus. Also, the office hour information showing opening and closing times of each service centre will help him find the optimum route. The same principle applies to the asset management officer from the second scenario. Extra information such as how far rooms are located from each other, how many assets are to be collected at each location, etc. may help him schedule the field trips efficiently.

The resources to be considered by the workers may differ depending on the nature of the tasks, the skill level of the workers, or the kind of roles the workers play in an organisation. In deed, we believe that a thorough study into the

---

[2] By rejection of tasks, we mean choosing not to accept the task. Such task can be delegated, suspended, or re-allocated by the workflow system.

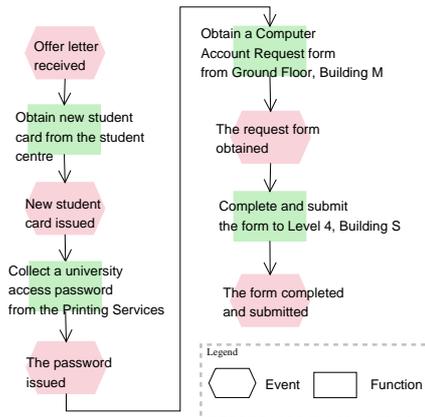[3] The readers are noted that these examples are simplified for illustration purposes.

Fig. 1. First Case: Event-Process-Chain diagram of the student process
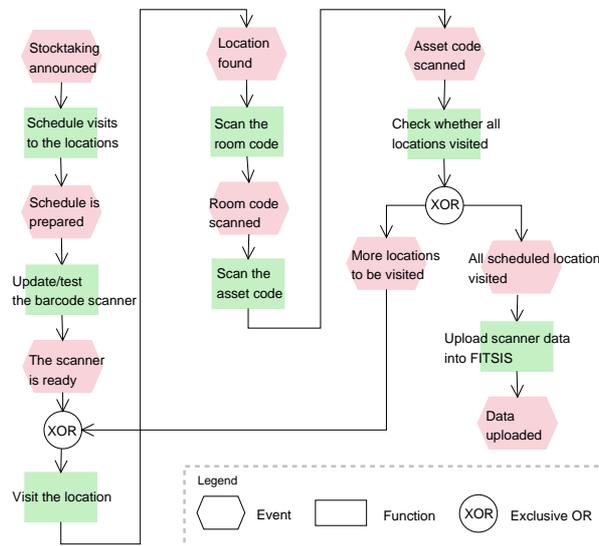

Fig. 2. Second Case: Event-Process-Chain diagram of the stocktaking process

requirements of the workers in making decisions as well as a survey of effective visualisation techniques have not been explored. This is an important part of our current on-going investigations, in which we look at identifying various types of resources that a worklist can provide to help the workers carry out the tasks. Also, we examine multiple aspects of the issue involving the nature of tasks, workers background, etc. Ultimately, the study aims to develop a detailed analysis of resources that will help identify the factors and their inter relationships that affect the worker's worklist management.

<div align="center">

Table 1. Generic types of resources

</div>

Resources	Descriptions
space	size or spatial information relevant to the tasks. It may be a diagram showing available storage rooms and their sizes, or meeting rooms and their capacities. This type of resource may be used to determine, for example, where 20 computers should be stored. It is separate from location, as sometimes the visualisation may not relate space with actual location of the space – space to store computers, but not interested in where.
materials	materials or consumable information relevant to the tasks. It may be an inventory list of materials to be used in the task, and whether you have enough of those things: number, volumes, weights, dimensions.
equipment	equipment information relevant to the tasks. It may be, for example, an inventory of barcode scanners showing their availabilities.
services	internal or external services information. It may be a list of travel booking agencies, printing services, or messaging services and their contacts/availabilities.
time	any "time" information relevant to tasks. It could be deadlines (ie., the time each task should be completed by), opening hours (ie., the time a particular service, for example a printing centre, is available) or a simply calendar showing working days. This type of resource will be useful in one Rs planning of the sequence of task executions.
location	geographical "location" information relevant to tasks. It could be a map of a campus showing locations of university facilities, a floor plan of an office block, or a diagram showing relative distances between locations. This type of resource also can be used in scheduling of tasks. We separate this resource from space as our model uses location in both the sense of a resource (maps), and as a generic place holder for the work item location in the visualisation (grid layout).
people	information about people and their roles in an organization. It could be an organizational chart showing roles and responsibilities of people. This type of resource may be used in finding the right person to seek for specific help or delegate a task to.
active worklist	current (active) tasks that are being carried out by the worker. This type of resource will help the worker determine the desirable workload, and effectively manage the current/future tasks. This can be represented by an arbitrary grid arrangement, where each cell represents a task to be performed, and may or may not contain other resource information regarding the task.

In this paper, we identify a few common types of resources that may have some generic applications. A list of the resources we have identified so far is presented Table 1[4].

---

[4] More resources will be added as the analysis becomes more complete.

4 Mapping resources to a worklist visualization

In this section, we describe our generic framework built for worklist visualization based on the resources we presented earlier. Again, for illustration purpose, we choose the four resources we described in the previous section; time, location, people and active worklist.

The visualisation framework is based on a layered approach, in which a background and overlay planes are used. A 2D representation of any of the resources forms the background layer. For example:

- The time resource uses a time line form of representation (eg., GANTT chart);
- The location resource uses a map representation that shows whereabouts and distance between locations (eg., Street maps)[5];
- The people resource uses a chart or diagram form of representation (eg., organisational charts);
- The active worklist resource uses a regular grid spatial arrangement; an arrangement different to irregular different layouts like maps.

The overlay plane consists of the tasks in a worklist. Each task is given (x,y) coordinates in relation to the background, which indicates the resource information allocated to the task.

Let us consider the first scenario. The student has four tasks to complete and he has to visit different service centres in the campus;

1. Obtain new student card from the student centre
2. Collect university access password from the printing service
3. Obtain a computer account request form from Ground floor, Building M
4. Complete and submit the form to Level 4, Building S

With our worklist visualization framework, we can present the four tasks as shown in Fig. 3. On the left-hand side of the figure, the background layer shows a campus map (ie., a form of the location resource). Each task (shown as a (round) coloured icon in the figure) is given (x,y) coordinates in relation to the map which indicates the location where the task is supposed to be carried out. For instance, the first task (a green icon on Building A) is placed on top of the building where the student centre is situated, and so on. The same worklist can be presented from the point of view of other resources. On the right hand side of Fig. 3, the chart diagram illustrates a visualisation of the same worklist from the perspective of the people resource. In this view, each task is given (x,y) coordinates in relation to a chart of organisational units. It shows which organisational unit is responsible for administrating each task. For instance, the first task (a green icon on Student Centre) is placed on top of the administration unit the student needs to contact if he/she needs help.

---

[5] Note that this could be different from the spatial map used to represent the space resource that shows occupying space, eg., building plan.
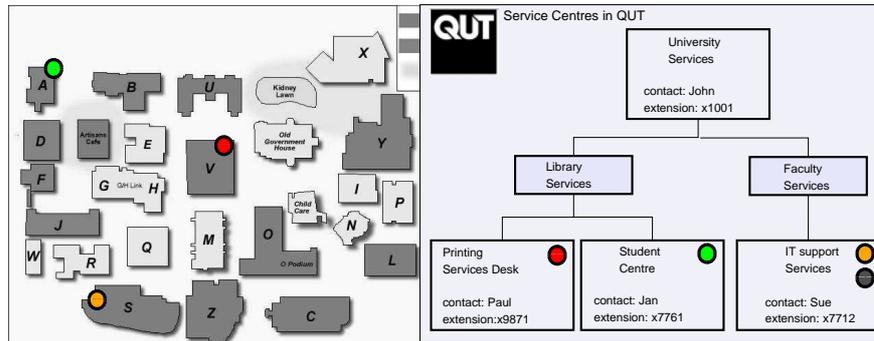
Fig. 3. Illustration of background and overlay structure of visualisation

All of the worklist items on a case appears on every "view". If the system selectively placed worklist items on a view, then the worker may miss items, or get confused by the changes from view to view. In addition, the item may require assessment from a number of resource viewpoints before being accepted or rejected by the worker. Therefore, we have the entire worklist shown on each view. The workers can easily switch from one view to another.

The framework is simple yet generic in that any types of resources can be presented using the overlaying technique. The same worklist can be viewed from different resource perspectives.

While having separate "views" can be useful, there is a requirement for cross-over with regards to the background-foreground information. For instance, the time resource may be shown as an overlay plane resource on a grid background; via the use of numbers overlayed on the grid locations. In Fig 4, examples of different views are shown; a worklist (in a grid format), location map and timeline. In each case the task is given a coordinate to arrange it in 2D on the surface of the background.

Each is a representation that can be used within a push-based task dissemination system to decide about task choices, with regards to the relevant resources. They can be turned on and off by the designer of the workflow visualisation to allow or deny access to extra information regarding tasks. Each one can be modified to suit a particular application area, thus leaving room for development of novel visualisations tailor-made for different applications [1]. So the general rules are able to be modified but can still be encapsulated in the development of a set of visualisation tools. Table 2 maps each resource type to an appropriate visualisation. The table is by no means exhaustive, and is only limited by the number of application areas intended for the visualisations. We present some that are appropriate for general visualisation applications [2].
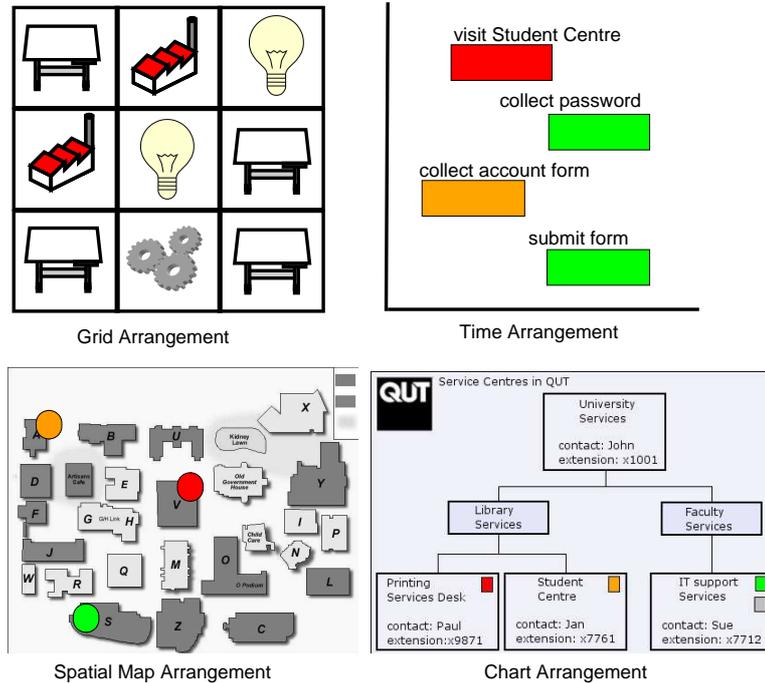
Fig. 4. Examples of the visualization categories

5 Worklist Management with Visualisation

In this section, we explain how the workers can interact with the visualized worklist. First, we introduce a generalized algorithm that workers use to manage their tasks, and then we explain the interactions between the workers and the visualised worklist.

5.1 A generic algorithm for managing a worklist

The resource view specified by [16] treats a resource as being human or non-human, and will have tasks directed to them by the workflow system. In this paper, for the sake of clarity, we use the term worker to differentiate human resources from non-human resources. Worklist items are distributed to workers within an organisation according to the process illustrated in the life cycle diagram in Fig. 5.

Inherent in this distribution process by the workflow system is the choice by the system of whom to give the task, via the offer actions. The workflow system will have a resource view that evaluates the capabilities of the intended recipient of the task. Furthermore, some of the resources will have an optimisation performed upon them by the resource view; eg. time and space, and will have this

information offered to the worker to help them with their decision. The worker upon receiving the task, must make a decision for themselves about accepting or not accepting the task. This process is out of the control of the workflow sys-

Table 2. Table enumerating the broad categories of resources, their use in worklist choice decisions and appropriately mapped visualisations.

| Resources	Worklist Choice Function	Visualisation
time	To compare the relative start and finish times for each task and insert it into the worklist at appropriate moments if time resources are available, either by leaving the task as whole, or dividing it into smaller components for insertion into small time gaps.	Gantt Chart showing all available tasks on a time line in stacked manner to identify insertion points for the worklist components.
location	To compare the spatial locations of tasks to be performed for logistical purposes.	Map detailing the arrangements of tasks in space, to aid the worker in identifying efficient ordering of the work.
people	To show visualisations of number of people available for task and their capabilities to assess who is appropriate for the task.	Overlays of people available to meet task with encoding of match between people and the tasks – colours/textures, including hierarchical views, social network views.
space	To compare the space resources required for a task to the space resources available.	Map detailed with space allocations showing empty spaces at certain times.
active worklist	To see a list of active tasks which can be checked out; user chooses according to the number of tasks they are able to perform.	Worklist dialogue, with overlaid data for comparison and choice of task.
materials	To view materials to be used in the task, and whether you have enough of those things: number, volumes, weights, dimensions	Overlays of information onto base background for any of the visualisations to compare materials with other materials available at that location, or a calculated indication of ability to meet this role.
equipment	To view equipment to be used in the task	Overlays of information onto base background for any of the visualisations to compare Equipment count with equipment available at that location, or a calculated indication of ability to meet this role.
services	To view availability of services from internal or external agencies in order to complete the task	Overlays of information onto base background of the availability of these services to meet the task.
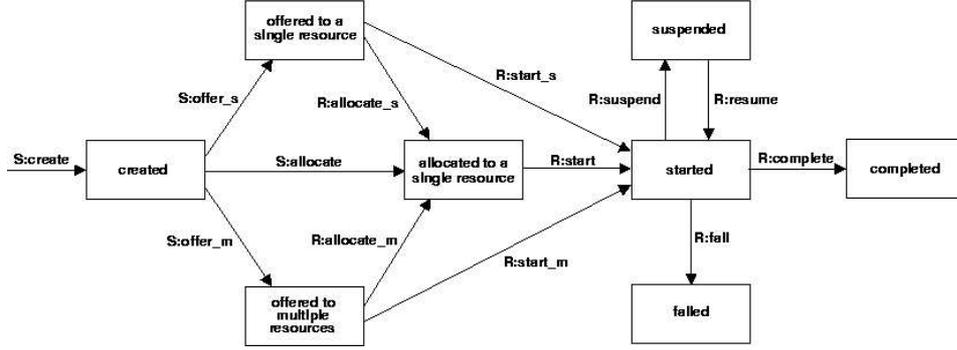
Fig. 5. Illustration of task life cycle; modified from [16]. Each box is the state of the task in a running workflow case. The prefix S and R refer to actions enacted by the Workflow System and the Resource (Worker) respectively.

tem, as it only can *push* tasks to the worker to request acceptance. The workers' responses have been characterised by *detour process*. A worker may *delegate* – hand to another worker, *de-allocate* – reject a task, *re-allocate* – task is handed to another worker by the system, *suspend/resume* – halt and then recommence a task allocated to a worker.

The question for the worker is the choice of adding or rejecting (ie., detouring) a task from his/her worklist. The task allocation can be a *push* or *pull* approach; push being system selected, pull being worker selected. Assuming a more pull oriented model of worklist task selection, our resource centric views of a worklist will aid the worker in this worklist management task, as they are able to decide which item to choose based upon critical resource issues.

The workflow system may offer a number or only one instance of the task to the worker, and at this point the worker may decide to perform the task by checking them out and adding them to a list of active tasks, or the user may decide to return the task to the unallocated pool via the detour process. Furthermore, the worker upon completion of the task checks the task in, thus removing it from the active checked out worklist. This task acceptance process may be represented by the following formula for the acceptance process, them being the check out processes respectively:

$$W_r = W_r \cup \{I\} \iff C_{W_r,T} > C_{\{I\},T} \tag{1}$$

where:

- W_r is the set of worklist items for worker(s) r;
- I is the new worklist item to be added;
- $C_{x,y}$ is the capability for the task(s) x of type y;
- T is the type of resource being processed (eg. Computer Equipment).

So, at any stage a worker will make a decision about whether to add a worklist item to its set of worklist items, by looking at the capabilities of the worker for

the present worklist as compared to the requirements of the new task. This can be automated, but the worker must be allowed to make such decisions as well, in order to promote a healthy attitude within the workforce. But it must be recognised that people will simply decide not to do a task, if they do not want to or decide to prioritise using undefined criteria. Furthermore, these visualisations may give information to the worker regarding the reasoning behind the choice of been allocated the task, and so the worker is left in an informed state about the reasons for work allocation.

5.2 Interacting with the visualized worklist

In the visualised worklist, each task is represented by a coloured icon. Some workflow systems support the generation of a number of instances of tasks, that may be disseminated to workers [17]. Thus at times an aggregated icon has to be used to represent multiple instances of the task in question. Fig. 6 shows an example of a task with multiple instances. An aggregated icon is shown with four icons with numeric information regarding the number of instances and their status within the system. The state of any delivered task at one time may be the following: inactive, available, checked out and suspended, and included is the colour we have mapped to the state using the traffic light metaphor of red, green and amber:

- Inactive – unavailable to the worker (grey);
- Available – available to the worker to check out (amber);
- Checked Out – has been checked out by the worker (green);
- Checked In – has been checked in and completed by the worker (red);
- Suspended – has been checked out by the worker, is still incomplete, but checked in to the user (amber – dashed);

Prepare Stock Check Report

Fig. 6. Illustration of an aggregated icon made up of single task icons. The example shows a task titled "Prepare Stock Check Report" with zero checked in, one checked out, three available and one task unavailable.

We use the traffic light metaphor due to its intuitive mapping to the status of the tasks: green active (go), red completed (stop) and amber available (in between go and stop). Furthermore, the available state is refined to have a

dashed amber appearance for those items that are suspended, and so the dashed appearance represents a partially completed task.

The worker interacts with the icons in a similar manner to previous worklists, by clicking on the icons to check out available tasks, and by clicking on checked out icons to check in completed tasks. Whenever appropriate, a form will be presented by the workflow system, to obtain data from the worker.

6 Implementation

A major test of any workflow visualisation approach is its ability to be incorporated into a modern client server-based workflow system. We have built a prototype of the proposed visualisation framework, and interfaced it with the workflow system YAWL. This section discusses the system architecture and implementation in detail.

6.1 The YAWL Environment

Our implementation is based on the open source workflow environment named YAWL (Yet Another Workflow Language), which is a research initiative at Queensland University of Technology [4, 17]. YAWL is based on a set of workflow patterns developed via analysis and comparison of a number of commercial workflow systems. It provides powerful and formal workflow description language, as well as an execution environment.

To understand the architecture of our visualisation framework, let us first present the overall architecture of YAWL. Workflow specifications are created in the YAWL designer which is a graphical editor, and deployed to the YAWL engine. The engine performs verification of the specifications and stored them in the YAWL repository. The specification can be loaded and launched for execution via the YAWL manager, and is hereafter referred to as a schema. The execution itself is managed by the YAWL engine.

The YAWL engine interacts with the components labelled as *YAWL services* through *Interface B*. The YAWL services (worklist handler, web services broker, interoperability broker and custom YAWL services) are based on the web services paradigm and all are abstracted as services in YAWL.

How the engine communicates with the YAWL worklist handler is of particular interest in our work. The worklist handler is the component that is responsible for dispatching tasks to the workers. Through the worklist handler, the workers accept tasks and mark their completions.

In conventional workflow systems, the worklist handler is part of the workflow engine. However, in the YAWL environment, it is a separate component that interacts with the engine through Interface B. Through the interface, a custom service or application can be developed to extract worklist information for display in whatever manner is required.
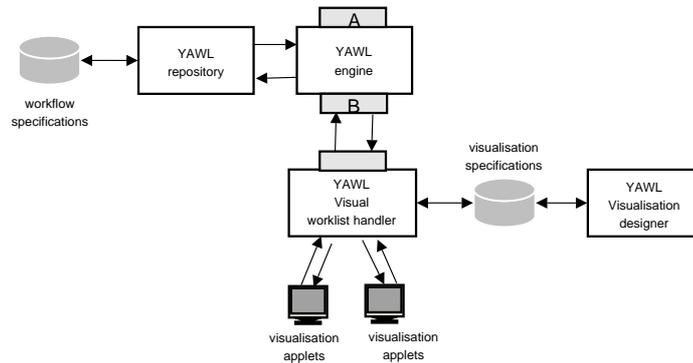
Fig. 7. YAWL Visualisation Framework: Overall architecture

6.2 Worklist Visualisation Architecture

Based on the existing YAWL architecture, we have developed a new type of YAWL worklist handler which interacts with the engine through Interface B. The overview architecture is shown in Fig. 7. It has capabilities to (i) display the visualised resources and (ii) dispatch tasks like a normal worklist handler. The architecture consists of two components which have designed and partially implemented: a visual worklist handler and a visualisation designer.

The visual worklist handler can view multiple cases of running workflows, with multiple resource-centric views matched to the requirements devised by the YAWL schema designer. The worker loads the cases and is presented with a list of tasks, and a tabbed view list to switch between difference representations of the worklists. In the following two sections we describe the two components, and illustrate them with mock ups containing partially developed examples.

6.3 YAWL Visualisation Designer

The designer application is the most complete at this stage. It is designed around the structure of the visualisation approach we have developed, and is implemented in Java, as is the rest of the YAWL implementation. The visualisation designer allows the user to load Scalable Vector Graphics (SVG) files as backgrounds and icons for the overlay planes. This allows easy modification of images via other drawing tools. The SVG component of the designer is managed by the BATIK Java package [18]. This is thus an implementation of the task coordinates scheme we detailed earlier. This designer allows the easy outlaying of tasks as icons across the background in the program.

The process of designing a visualisation view for a schema is as follows:

1. First decide on the background and overlay images, editing them in a separate tool and saving them as an SVG file;

2. Decide on the spatial arrangement of the tasks to be displayed according to the resources that need to be analysed, for example a map for logistics on QUT campus that will help a worker to decide where to perform their tasks;
3. Load the workflow schema into the editor to obtain the tasks in the system, which appear in a mouse menu on a right click at the chosen location on the background;
4. Load the background image;
5. Set the current icon to be used by choosing from the list in a dialog;
6. Move pointer to location of worklist item and right click to choose a task, and icon, repeating for all worklist items.
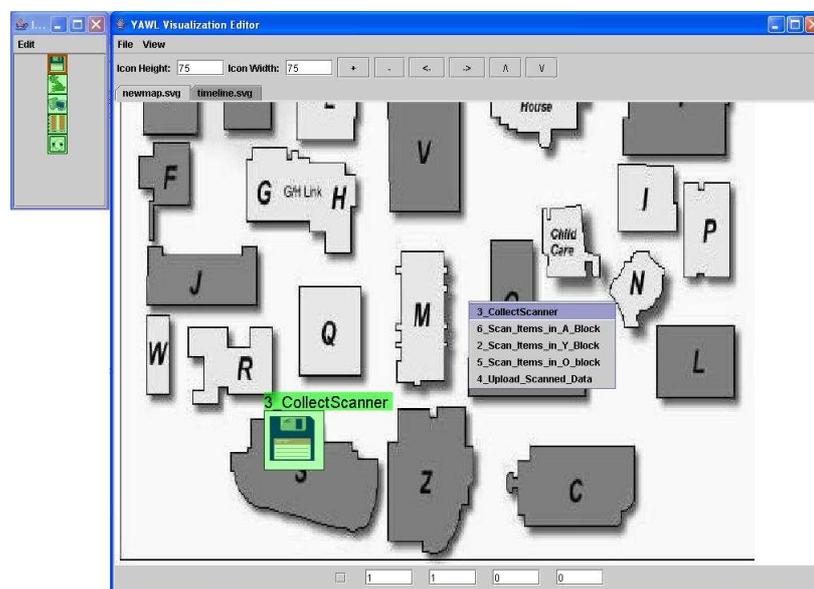

Fig. 8. Yawl Visualisation Designer: main components

Fig. 8 illustrates the major components of the visualisation designer user interface via the stocktaking example on campus map (Fig. 2). The large window (right) is the main window for visualisation design, and the smaller window (left) shows a list of potential icons to be placed at locations on the visualisation. Each view is placed into a tabbed list, as they are to be displayed in the visualisation agent. The menu is displayed using a mouse right click, showing the tasks defined in the schema. The icon can be placed at the location of the right click of the mouse, or using actual coordinates in the text entry boxes at the bottom of the screen. The icon at the bottom left of the image is the current task icon, "CollectScanner" and is shown using a disk icon.
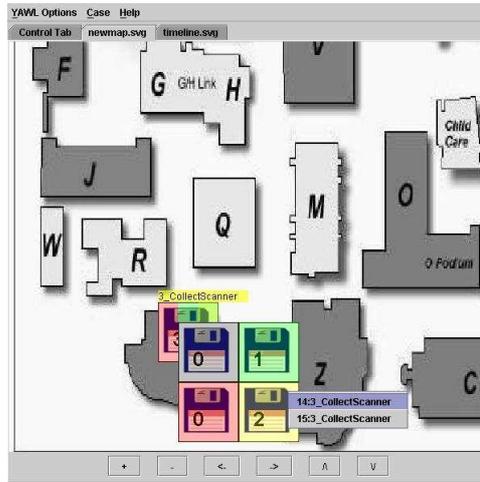
Fig. 9. Screen dump of a running visualisation handler, showing a campus map visualisation with an icon showing the "CollectScanner" icon from the PC Stocktake example as an available worklist item in orange amongst multiple instances.

This visualisation design information is stored in an XML file that defines an arbitrary number of views per schema, and the task icons, gained from the number of tasks within the YAWL schema. This file is then read by the Visual Worklist Handler to form the visualisation structure for communication to the YAWL engine. The following is a snippet from a visualisation specification. A specification may have a number of <view>s, and each view may have a number of <task>s. A view is associated with a background representing a resource. Each task is assigned a color for the description, coordinates, and an icon.

```
<specification id = "TSSstockTake.ywl"
               uri = "file:/D:/Yawlstuff/batik/demo/TssStockTake.xml">
<view id = "file:/D:/Yawlstuff/batik/demo/map-1/newmap.svg">
<task id = "3_CollectScanner">
<color> -16777216</color>
<coordX> 240</coordX> <coordY> 760</coordY>
<icon width="75" height="75">file:/D:/Yawlstuff/demo/floppy.svg</icon>
</task>
</view>
</specification>
```

We have implemented the beginnings of a visualisation editor and visualisation viewer, which we show in this paper. In a final implementation, additional resource information will be selected from the resource view of the YAWL schema as it is running. For now we are able to design worklists arranged according to grid, spatial and time arrangements.

6.4 YAWL Visual Worklist Handler

Worklists are disseminated in YAWL via the default worklist handler as simple dialogs containing lists of tasks, with no other resource information being displayed. We have begun implementing a visual worklist handler that is an extension of the default handler. The YAWL workflow implementation is structured around a component architecture that communicates via XML formatted commands. Thus the worklist handler is able to utilise the B interface to the running YAWL case in the same manner as the default worklist handler. The visual worklist handler is able to execute the visualisation developed with the designer that is stored in a file (see Fig. 7).

The new worklist handler allows a more intuitive mapping of task coordinates to the check in and check out process. The user is able to check items in and out by simply clicking on the potential worklist item in its location on a map or hierarchy diagram. right image is a mock up of the student enrolment example visualisation running within the visualisation handler.

With a spatial organisation to the tasks, the person doing this registration process can evaluate the task, using the map to make a decision about the acceptance of the worklist item in consideration of the location resources.

7 Conclusion

We have described the beginnings of a thorough analysis of workflow visualisation; its theoretical basis, resource centric approach and appropriate visualisation techniques. Analysis in our paper showed to use these techniques within a typical workflow system. The task coordinate approach was described, showing how this can be generalised across a number of visualisations using a background and overlay approach. We have also begun the development of a visualisation development environment, with an editor and visualisation agent that uses SVG files and is easily integrated into the YAWL workflow system created by the BPM group at QUT. Thus we have indicated that this visualisation approach can be used within a fully featured workflow environment.

Further analysis will continue to refine the visualisation mappings to produce a knowledge base for development of visualisations within workflow applications. In particular, there will be refinement of the broad categories of resources into more fined grained categories to derive a rule-base for an intelligent design agent to be incorporated into the visualisation designer. Evaluation experiments will be performed within a case study in order to ascertain the effectiveness of the resource centric visualisation approach with users of workflow tools.

In addition, we will exploit the latest resource view developments that are being implemented within YAWL, to enable the run time specification of resources associated with a task, and thus extend the implementation to include automated run time visualisations of resources such as people and equipment, associated with the tasks. We will thus extend the visualisation editor and agent to accommodate these resources in a structured manner, according to our table of visualisation mappings.

Acknowledgement

References

1. Keller, P., Keller, M.: Visual Cues. IEEE Press., Piscataway USA (1993)
2. Tufte, E.: The Visual Display of Quantitative Information. Graphics Press, Cheshire, USA (1983)
3. Andersson, T., Bider, I.: Introduction of bps systems into operational practice: Achievements and setbacks. In: Proc. of BPMDS, Riga, Latvia (2004)
4. Aalst, W.M.P.v.d., Aldred, L., Dumas, M., Hofstede, A.H.M.t.: Design and implementation of the YAWL system. In: Proc. of CAiSE, Riga, Latvia (2004)
5. Luttighuis, P., Lankhorst, M., van de Wetering, R., Bal, R., van den Berg, H.: Visualising business processes. Computer Languages (2001) pp.39–59
6. UNISYS: 3d visible enterprise (2004) www.3dvisibleenterprise.com/3dv/.
7. Schonhage, B., van Ballegooij, A., Elliens, A.: 3d gadgets for business process visualization:a case study. In: Symposium on Virtual Reality Modeling Language, Monterey, California, ACM Press (2000) pp.131–138
8. Systems, I.S.: Interactive software (2004) www.interactive-software.de/.
9. Grohn, M., Jalkanen, J., Haho, P., Nieminen, M., Smeds, R.: Visualizing human communication in business process simulations. In: Visual Data Exploration and Analysis VI., San Jose, SPIE-Int. (1999) pp.140–148
10. Hsu, C., Yee, L.: Model-based visualization for enterprise information management. In: Proc. of 4th Annual Conf. on Artificial Intelligence, Simulation, and Planning in High Autonomy Systems, Tucson, Arizona (1993) pp.324–327
11. Gershon, N., Eick, S.G.: Information visualization. Computer Graphics and Applications **17** (1997) pp.29–31
12. Wright, W.: Business visualization adds value. Computer Graphics and Applications (1998) p.39
13. Latva-Koivisto, A.: User interface design for business process modelling and visualisation. Technical report, Department of Computer Science, Helsinki University of Technology, Helsinki (2001) Masters Thesis.
14. Leroux, H., Exton, C.: Coope: a tool for representing concurrent object-oriented program execution through visualisation. In: Proc. of 9th Euromicro Workshop Parallel and Distributed Processing. (2001) pp.71–76
15. Jennings, N., Norman, T., Faratin, P.: Adept: An agent-based approach to business process management. ACM SIGMOD Record **27** (1998) pp.32–29
16. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow resource patterns: Identification, representation and tool support. In: Proc. of CAiSE, Porto, Portugal, Springer Verlag (2005) (to appear).
17. Aalst, W.M.P.v.d., Hofstede, A.H.M.t.: YAWL: Yet another workflow language. Information Systems **30** (2005) pp.245–275
18. Batik: Batik svg toolkit (2005) http://xml.apache.org/batik.