System Architectures for Real Life Workflows

Peter Nacken
Océ Technologies BV
Venlo, the Netherlands

Abstract

In this paper, an approach to build systems from software building blocks is described, that makes it possible to hide the inherent complexity from the user but that is still flexible enough to support a wide variety of applications.

Such a modular approach is very powerful, but requires the producers of soft- and hardware to see their products as building blocks and faces users with the challenge – and opportunity! – to configure a system that best suits their needs.

Introduction

The design of software packages, such as workflow tools for document production centers or office environments often starts with the analysis and modeling of the tasks and workflows in the application area. This may lead to complex applications, which are supposed to cover a large number of situations.

In this paper, an approach to build modular systems from software building blocks is proposed, that makes it possible to hide the inherent complexity from the user but that is still flexible enough to support a wide variety of applications.

Some Observations on Work Flow

The description of workflows is sometimes treated as a complex matter, in which the work that people actually do on a day to day basis is disentangled into a large number of basic steps.

Yet, in many cases, a fixed sequence is executed time and time again in a particular situation. For example, there are environments where scanning is always followed by printing, but in other environments, scanned documents are almost always sent to a digital archive, perhaps with OCR and indexing capabilities. In an abstract work flow analysis, many successive steps are described, but to the user, the successive steps are tightly linked.

Taking this observation into account, we are faced with some challenges when defining system architectures for real life work flow.

- The complexity that is inherently present in the workflow or that originates from the use of different tools must be hidden from the user. Most of the time, the user does not want to be aware of the individual processing steps.
- The operator must always have the feeling that he is in control, yet he must not have the feeling that he has to do the same things time and time again.

These challenges can be met by keeping the 'green button approach' in mind. On copiers and multifunctionals, the green button is the button that immediately does what the operator wants – most of the time. More complex operator interaction is only required when a deviation from the 'main stream' workflow is required.

To key to a 'green button' approach at the system level is configurability. As different users have different work flows, their wishes as to what is under the green button will differ. In this paper, an approach is suggested to for building complete systems by linking applications.

A Model for Software Architectures

In Ref. [1], Hamberg described a software architecture concept for workflow software by making analogies to the way people co-operate in a workflow to perform a task. In a workflow executed by a group of co-operating people, each person is responsible for one step in the workflow, which is transforming his or her 'input' to some form of 'output'. This transformation can be seen as a change of the state of a document. After a number of state transitions of the input, the final result is reached.

In the software architecture concept that Hamberg described different 'workers' all have the task to perform a specific action on a document in an intermediate state: receive print data from the network, print bitmaps, et cetera. The workers communicate indirectly via an information repository, from which each worker gets its input and where each worker places its result.

In this paper, the analogy between human work flows and software architectures for work flow is taking one step further by focusing on the way documents are passed form one human worker to the next in many real life environments.

In many document production centers, documents in a 'transition state' are put on a table or in a cupboard for the next person to pick them up. Documents in an intermediate state can be:

- A set of paper originals to be copied.
- If black-and-white and color prints have to be merged, the combination of a print file for the black-and-white printer and a number of color prints can also be seen as an intermediate state.
- A box of loose prints that have to be stapled or bound by the finishing department.

An important part of the workflow is captured in the physical place where intermediate results are placed – to the extent that a document can be 'lost' or processed in the wrong way if it is put in the wrong place.

In the approach for building systems described here, there is a direct analogy. At the most simple level, the tables and cupboards are replaced by folders on a file system. Each piece of software ('worker') monitors the content of a dedicated folder (sometimes called a 'hot folder'), processes the input that arrives there, and sends the output to a second folder.

A next level of complexity – and therefore flexibility – can be added if 'workers' do not pick up directly work from their hot folder, but if they are aware of the type of input they need. Again, an analogy with a 'human' workflow can be drawn. If an operator knows that he has to print a document on a black and white printer and insert pre-printed color sheets in the job, he will not start until both the digital document to print as well as the color sheets are present. In a workflow software architecture, one can think of the following examples:

- A digital document is submitted to a printer when both a digital document (for example a PDF file or a collection of TFF images) and a job ticket with instruction for the job, are present.
- A scanned report is mailed to the addressees after an OCR application read the list of addresses and a database application found the appropriate e-mail addresses.

Advantages and Challenges

The approach described here has a number of specific advantages.

- There is a very loose coupling between the different workers. This makes it straightforward to combine software form different vendors and even software running on different platforms.
- Workers in software and human workers can co-operate in the workflow. This can be very relevant if you want to insert a Quality Assurance step in a workflow. Just create a folder for 'work to be QC-ed'.
- The programming skills required to configure a complete work flow are relatively low.

In the field, we now see this type of software architectures for workflow emerging, containing just a few building blocks. Yet, one can imagine far more complex workflows and services based on this approach. Think for example of a service to scan paper documents, clean the scans, OCR the result, convert them to PDF, make the PDF searchable, and mail the result to a number of people.

There are two types of challenges related to the approach described here. The first challenge is for providers of soft- and hardware, who need to look at their products not only as complete systems, but more and more as half-products or building blocks, that can be combined for (or even by) their customers. This requires them to think about open digital interfaces and configurability of their products.

An example of a half-product in the scan-to-file option on the Océ 3165. This multifunctional scans documents as TIFF or PDF and places them in a 'hot folder'. The scanner can be configured with 'profiles', that can be given names that are meaningful to the operator ('scan to archive'; 'scan to printer'). The scan profiles are encoded in the filename.
An application (for example Océ File DispatchIT) that watches the 'hot folder' can read the file name and can be configured to execute the corresponding next step.

The second challenge – and opportunity! - is for users that want to benefit most from the possibilities of combining half-products, will have to think about the workflows they want to build. This is not easy, but may enable them to deliver unique services that the producers of the half-product did not even think about.

Conclusion

In many situations, documents have to undergo complex work flows with many successive steps. Yet, the sequence of steps is often fixed, and exposing users to all steps is unnecessarily complex. An approach to build configurable systems in which this complexity is hidden is proposed. This approach is potentially powerful, but forces producers of hard- and software to think of their products as half-products or building blocks, and confronts users with the challenge – and opportunity! – to configure the system that best suits their needs.

References

1. Roelof Hamberg, Document Workflow Support: Applications and Controllers, International Conference on Digital Production Printing and Industrial Applications, Antwerp, Belgium, 2001.

Biography

Peter Nacken holds MSc degrees in Physics and Mathematics and a PhD in Computer Science. From 1994 to 1999 he worked in the oil industry for Shell and Jason Geosystems. Then he joined Océ Technologies, where he has worked on software for Document Production Center and on the architecture of high volume printers.