

Forms processing

Form and function

Jon Mell kicks-off a new series of workshops on automated forms processing by introducing the latest tools, techniques and technologies.

Filling in forms is one of the curses of modern life.

Complaining about the amount of time it takes – and getting confused about what to put where – are almost universal experiences. The automation of form-filling by computer systems and web applications has, from the user's perspective, merely removed the ink and Tipp-Ex from the process. From the developer's perspective, it has introduced tedious, complex and error-prone navigation and validation coding.

Let's start with simple definitions. What is a form? I would say that a form is a self-contained way of obtaining all the information that a person needs to provide to kick off or push forward a process; it is also a way of encapsulating that information for input to the next stage in the process.

Early computer systems had neither the technology (an easy-to-use graphical user interface) nor the 'intelligence' (separation of presentation from logic) to cope with either of these ideas satisfactorily. So forms were widely regarded as merely 'entry screens' to applications.

Even when graphical user interfaces (GUIs) were introduced, programming languages continued to enforce a tight coupling between the presentation and

By Jo· Mell

the logic. That made making simple modifications, such as changing the presentation of the form, a nightmare. Besides, an old-fashioned paper form can be filled in almost anywhere, at pretty much any time of the form-filler's choosing. When, where and how a customer might fill in a computerised form should ideally be their decision, not the application provider's.

Think of a physical, catalogue-based retail store, such as Argos in the UK: you browse through the catalogue at your leisure, find the item you want and fill in a form to place your order. This is exactly what we do when we purchase from an online store. However, we (and, more to the point, developers) do not necessarily view the screens we use as forms, let alone design them accordingly for ease-of-use. Instead, we view them as screens and fields in an application.

This view has led to a large amount of code being written to maintain state and perform validation as a user progresses through the form. Developers have to write code to capture the information on page one, then the program must scan and validate the information and return

the user back to that page if there is any incorrect information (along with a comprehensible error message and instructions). If not, it must hold the information in memory while page two is displayed. Also, if a user decides to return to page one for any reason, developers need to write code to check whether information already exists from the page one fields and re-populate that data.

This is obviously a tedious and error-prone process, as well as time-consuming. Developers end up writing code to handle forms processing, rather than business logic that delivers value to the end-users. Especially with early developments, this forms-handling code was intertwined with business logic code, meaning that simple requests from the business to IT to change the way data was collected took a long time to be delivered, due to the development and testing complexities brought about by this combination of presentation and business logic.

Forms and frameworks

The introduction of 'modal view controller' (MVC) frameworks such as Struts and Java Server Faces (JSF) has simplified the coding of online forms, but has not eliminated the problems



entirely, because these frameworks have no concept of an encapsulated form as a deliverable.

An encapsulated form, as the phrase suggests, contains all the information a form requires, not just the 'bits' the user fills in. This includes business rules (for example, if a credit application is for more than £1,000, then the form needs to be routed appropriately for approval), attachments (such as contractual terms and conditions) or signatures. The MVC/JSF approach merely thinks about what the user has filled in (such as his/her name and address).

The key to intelligent-forms management is not in the user interface, it is in how you deliver the encapsulated form. That had to wait for the near-universal adoption of a data encapsulation standard: XML.

XML, of course, was originally developed to provide a metadata standard for describing the many elements of electronic documents. This was intended to make it easier for one company to send business information to another because all the common elements would be described in a standard manner. Being a free and open standard, like the technologies that underpin the worldwide web, for example, hastened its universal adoption and its popularity has encouraged the Worldwide Web Consortium (W3C) to develop similar, related technologies.

One of those is XForms, which now enables us to treat computerised forms as forms in their own right, rather than as applications. A form can now be regarded not as part of an application, but rather as an entry point to a business process. This is how paper forms are treated within an organisation. They are *loosely-coupled* with a business process; a form can be changed without affecting the entire business process. The ability to emulate this in information systems provides tighter integration between IT and the business, enabling IT to respond to business requests at speed. This

therefore makes the whole organisation more responsive.

Furthermore, if we take a forms-centric, process-oriented view of the world, rather than thinking about applications with entry screens, a host of other opportunities opens up. Form design tools, such as IBM's Workplace Forms Designer, are specifically oriented to building forms, rather than

5. The CFO signs it;
6. The CFO posts the form back to the depot.

We could, of course, build a workflow application to assist with this business process, but an intelligent-forms-based solution has a number of additional benefits. The intelligence of the form helps in the following ways:

The introduction of 'modal view controller' frameworks such as Struts and Java Server Faces has simplified the coding of online forms, but has not removed the problem.

using a combination of HTML (for presentation) and JavaScript (to provide validation) that is slow to develop and hard to maintain.

Validation rules ought to use the language of the business, and it ought to be possible for forms to be filled in, amended and viewed online or offline. Finally, the information entered can be encapsulated, stored and processed as an XML message. This makes it easier to use the form in the context of a process as it has all the additional information that an encapsulated form requires. Such a form can genuinely be described as intelligent.

Consider this simple business process for a purchase-order sign-off:

1. An employee raises the request form and e-mails it to their manager;
2. The manager signs it and faxes it to credit control;
3. The credit control manager can sign off if it is under £1,000. If it is more than that, it is sent by post to accounting management (faxing again would render the form unreadable);
4. The accounts department can sign off purchase orders of up to £10,000. If it is more than that, accounts takes it across to the chief financial officer (CFO) for approval;

1. The 'intelligence' in the form will know that it is the employee's manager that needs to sign it rather than anyone else and can interrogate the company's directory to get his e-mail address. The employee does not have to remember the process, as the form is process-aware;
2. Faxing is eliminated, as digital signatures replace handwritten ones;
3. The form understands the £1,000 limit and routes the form appropriately, so process mistakes are eliminated. Again, e-mail addresses and the correct personnel are identified by the form;
4. If any annotations, comments or attachments are required at any point, these can be attached to the form. Everything around the process is captured in a single entity;
5. The form can take several levels of signature, in this case potentially employee, manager, accounts and the CFO. The signatures are layered, so once one area of the form has been signed it cannot be changed. Digital signature pads can be attached to a PC, if a 'real', handwritten signature is required;
6. The process should be completed in significantly less time than previously. Also errors should be much reduced.



In a traditional information-systems environment there are often different owners and versions of forms depending on their context: application data-entry panels, paper forms (PDF and hardcopy), Java Server Pages (JSPs) or Microsoft Active Server Pages

Microsoft. As you can see, the real business value comes once printing is taken out of the equation – as shown in the example of the purchase-order process above. It is only when an intelligent form becomes part of an *automated* process that its true value

of websites already do this. In addition, the form can be displayed in a ‘wizard’ based fashion, gently guiding the user through the process and only asking relevant questions.

If there is only one applicant, for example, the section for a second applicant’s name and address will not be displayed. If Section B of the form (all two pages of it) is only required if the user answered ‘yes’ to question nine, it will only be displayed if this is the case.

This can be especially useful if, generally, only ten per cent of respondents answer ‘yes’ to question nine – it will cut the number of respondents who drop out, frustrated at the form’s length.

Take, for example, an application form for a managed-investment savings fund. It may ask how the applicant would like their money invested. This approach can be combined with the validation capabilities of intelligent forms by ensuring that the split between various funds in the application do add up to 100 per cent – alerting the applicant should they not add up correctly – reducing the chance for error and speeding up the process of taking on a new customer.

Agility

So far, we have concentrated on increasing business agility, but there is no point in being agile if the back-end systems you rely on are not available. An intelligent forms-based solution naturally lends itself to being able to fall back to a manual process – an intelligent form can be printed, signed with a pen and posted in the mail.

So intelligent forms can help a business in terms of agility, speeding up business processes and aligning business and IT. We can also revert easily to manual processes in case of a business continuity event. There is a significant reduction in the amount of paper used, assisting companies in their corporate responsibility challenges.

It is only when an intelligent form becomes part of an *automated* process that its true value can be realised.

(ASPs) for web self-service. When we think of the world in a forms/process manner, we have a single representation of a form which can be used in a manual or an electronic fashion.

Changing times

One of our customers changed their name. Because they had an application focus rather than a forms focus, and separate management systems for printed and computerised forms, changing all the printed forms took 18 man months and cost £500,000 (around \$1 million).

In fact, the largest amount of time and cost was spent trying to find all the different places where PDF versions of the printed forms were kept in order to change them. If they had had one copy of every form in a repository used for both electronic and manual forms it would have been hugely beneficial.

Now, take a look at this table showing example estimated cost savings per form for various uses of forms:

■ Print and read	£0
■ Print and fill	£7
■ Fill and print	£8
■ Fill, sign and submit	£51
■ Store and process	£81

These figures were calculated by analyst group Gartner in a study commissioned by software giant

can be realised. For this to happen, the intelligent form needs to be integrated into existing applications.

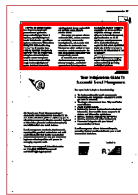
Fortunately, the prevalence of open standards makes this perfectly achievable. XForms is based around XML and most common forms design and delivery tools have application programming interfaces (APIs) for Java and C coding languages. They are also compatible with standard web services technology.

In the purchase order example, we described how an intelligent form could query a corporate directory to know who the correct contact was for the next step in the process. This could be applied to any back-end system. Product names and descriptions (even photos) could be provided from a part number.

By combining this back-end integration with process awareness, intelligent forms become even more powerful. Forms can be dynamically altered as they are entered to make the user more comfortable filling in the form. A classic example relates to financial services products.

These forms are often long and laborious to complete and the mere sight of them can put people off. Intelligent forms can guide applicants through the process, only showing the user what they need to see.

For example, a user could enter their postcode and have the rest of the address automatically completed – lots



However, the removal of paper from the process does involve questions around governance. Storage and archiving of forms is important, especially in legal and financial institutions in a post-Enron regulatory environment. We have been working with customers for many years in terms of scanning faxes, orders and claims that come into an organisation from the outside world, storing them electronically in an enterprise content management system that is secure and available enough for the original piece of paper to be deleted, dramatically

reducing physical storage requirements and providing instant access to archived records.

This content management solution understands XML, so many customers are now looking to remove the scanning element, and to receive their information via an electronic form. The form can still be archived to meet compliance regulations and receive the same access benefits as the items which were previously scanned.

In conclusion it can be seen that forms are a core component of every enterprise – even in their paper versions. By having a robust

and integrated forms IT strategy, an organisation can obtain significant competitive advantage. Business agility can be dramatically improved, paper can be taken out of the equation yet manual processes can still be invoked should they be required. In this way, all organisations should take their forms strategy as seriously as any other element of their IT strategy. ■

Jon Mell is software services manager at computer services supplier Logicalis. He can be contacted directly by e-mailing Jon.Mell@uk.logicalis.com.