A Community Driven Micro-Services Architecture Supporting Long Term Digital Preservation

Mark Evans; Tessella Inc; Rockville, MD, USA Bill Steel; Tessella Inc; Rockville, MD, USA Robert Sharpe; Tessella plc; Abingdon, UK James Carr,; Tessella plc; Abingdon, UK Alan Gairey; Tessella plc; Abingdon, UK Jonathan Tilbury; Tessella plc; Abingdon UK

Abstract

This paper describes a digital preservation platform that has been developed over a seven year period in partnership with several national memory institutions. The platform provides a viable solution to the challenges of long term digital preservation by delivering a flexible, extensible set of micro-services that are orchestrated into processes to provide capabilities around ingest, storage, data management, preservation and access. The use of micro-services allows a policy neutral approach, allowing policy decisions to be changed without the need to modify the underlying architecture. In addition it allows them to become open source, which enables and encourages a community solution to commonly faced issues.

The platform, known as the Safety Deposit Box (SDB), has been built from the ground up to be compliant with the OAIS reference model from a functional and informational standpoint. This paper describes the main architectural concepts that are employed in the platform, in particular the flexible and extensible frameworks that allow functionality to be encapsulated in user or community provided micro-services and workflows.

An emphasis will be placed on the set of services that provide tools for guarding against the challenge of file format obsolescence, which include characterization, preservation planning and preservation action. Since a micro-services approach has been adopted the system can evolve over time; new functions can be added and, existing functions can be enhanced. This is particularly important as services within the digital preservation domain are in their infancy. Format migration and emulation have been adopted as the primary long-term preservation mechanisms and an approach to validating such pathways is described.

The use of micro services allows existing best-of-breed tools to be incorporated into the solution.. It has been configured to incorporate the DROID and JHOVE tools as well as other commonly-used open-source and commercially provided characterization and migration tools.

The paper includes a description of a community driven approach which includes the sharing of collective knowledge and experience the encouragement for the exchange of micro service and workflow, and the ability for the community to significantly influence the future road map of functionality.

Background

Safety Deposit Box (SDB) has been built specifically to deal with the problems of digital preservation in libraries, archives and other organizations facing the problems of very long-term or permanent retention of digital content; sufficiently long-term that the material will outlive the software or hardware used to create it.

Its primary purpose is to retain born-digital information objects in perpetuity and provide access to them in current technologies, while recognizing and working with the inevitable cycle of technology changes. The information held by the system must be retained reliably over numerous software and hardware refreshes and that has been the main driver of the design of the system.

The SDB platform was originally conceived in 2002, as a digital archive solution for the UK National Archives (TNA). It was developed around the same time that the initial draft of the Open Archival Information System (OAIS) reference model [1] was published. The first incarnation of the platform was broadly in alignment with the OAIS reference model, and consisted mainly as a set of major functions covering ingest, storage, access and some elements of preservation planning.

The Safety Deposit Box is a true archive containing both "passive" and "active" preservation functionality. In common with a number of other repository offerings, SDB provides "passive preservation" to ensure that information objects can be securely and reliably stored, managed and accessed. However, it also provides "active preservation" functionality to maintain the information objects despite changes in technology including those that alter the physical file structure.

Thus, while it is quite common to find systems that store what they are passed and allow access to it, SDB additionally provides a framework and real tools that allow archivists and librarians to actively preserve their content in the truly long-term. From the very beginning the SDB system (including the ingest, workflow and passive preservation functionality and the data model behind it) has been designed with the needs of "active preservation" in mind and this is the core driver of the functionality that SDB provides today

The concept of micro-services was not fully realized from the beginning. There were a number of cohesive functions that aligned with the concept such as a fixity service and basic characterization services (format identification and validation), but the concept was not utilized throughout the entire system

Since the first incarnation, in conjunction with the EU funded Planets project [2], additional “active preservation” capability has been added together with a flexible workflow component and the architecture has transitioned to more of a micro-services oriented paradigm.

Today SDB is in its 4th generation and now provides a core platform with the ability to “plug in” additional micro-services and workflows in a simple and flexible manner without the need for major changes to the core system. These additions can be made to all functional areas of the system through the use of a standard set of APIs and a provided software development kit. End users and third parties are encouraged to develop and share their own micro-services, leading to more of a community driven development approach.

As will be discussed later the use of micro-services and workflow as an extension to the core system allows the platform to support specific end user business processes and policy in the areas of ingest, storage, preservation and access. This flexible and extensible approach has drastically reduced the time and cost to provide future system customizations and enhancements to the user community.

SDB is currently in use by over 10 institutions worldwide. [3]

Microservices

The term “Microservices” was popularized by Abrams et al [4], [5] and describes the concept of devolving large complex functions in a digital preservation system into a set of small independent interoperable functions with well defined persistent interfaces – micro services. These small functions are, by nature, easier to define, develop maintain and enhance over time. Micro services can be orchestrated or ‘chained’ to provide a more complete process in a flexible and extensible manner. There are many other advantages associated with this approach:

- Relative low cost to add additional services
- Efficient re-use of functionality
- Easier to deprecate functionality when it has been outlived, or a more mature implementation is available
- Wider impacts of implementation changes are minimized
- Reduces the dependence on a particular technology stack
- Can be shared across, projects, and communities
- Can be widely distributed across multiple domains

Such a concept is not a new one; indeed it is one of the foundations of the service oriented architecture (SOA) approach [6]. However in the context of digital curation and preservation the micro-services approach is gaining popularity in use. iRODS [7] have based their entire architecture on the concept, Univeristy of California Merritt system [8] and the Archivematica toolkit [9] have also adopted the concept.

In each case the granularity of a micro-service’s functional scope is somewhat arbitrary, and there appears to be few hard and fast rules to guide system developers; [5] and [7] provide some degree of guidance. Having too fine a granularity can lead to very complex orchestrations, and too large a granularity diverges from the micro-services concept, and reduces flexibility of use. As examples the University of California Curation Center (UC3) have identified an initial set of 12 micro-services that cover their curation and preservation capabilities; at the time of writing iRODS currently provides over 150 core micro-services. On inspection it is clear that a much more fined grained approach is being used by iRODS.

The micro-services concept was not the initial foundation of SDB. However, driven by the need to both generalize the system to support new organizations and the need to allow the system to be specialized to meet individual organization’s needs, the need for a core framework into which plug-in functional components can be added became clear. Hence, the platform architecture was radically upgraded and is now in a position where micro-services are a key component. Hence, micro-services provide an extension to a core system adding the set of small functions and business rules that turns the core framework into the system a user requires. This has been a natural progression as we see the need to provide a flexible and extensible solution that can be adapted to a specific designated community became clear. In other words, there is no “one size fits all” solution.

SDB Architecture

Overview

As previously mentioned the architecture of SDB has evolved throughout a number of iterations. Each subsequent iteration has moved the platform closer to a micro-services based architecture. Not only has the existing core system evolved in this nature but there has been significant addition of open source and commercial tools and services that have been incorporated as new micro-services. In future iterations it is anticipated that the majority of enhancements will be micro-services based. Many of these will come from within the SDB community but the framework will also enable this community to leverage external work (e.g., through the addition of more characterization and migration tools).

Current Architecture

SDB has a number of major architectural components that can be combined to deploy a complete Digital Archiving solution:

- **Workflow Manager** – The workflow component contains a set of core frameworks into which micro services can be deployed. Micro-services within the frameworks are orchestrated together through as a series of workflow steps to provide business functions across ingest, storage, preservation, access and data management.

- **Administration and Reporting** - Functions that allow administrators to control and report on the system

- **Metadata Storage** – This stores archival and process information and associated metadata in a series of published database schemas.

- **User Interface components** – Allow a user to access the system via supplied screens or execution of workflow
- **Storage System Interface** – A framework that allows content to be stored in and retrieved from bulk storage in a manner defined by storage adaptors
- **Application Programming Interface** – provides access to all functions that access and manipulate the stored information
- **Technical Registry** – which contains both factual and policy information and is used by the micro-services and workflow to implement business rules and policy. The technical registry is based on the Planets Core registry [10] which is in itself an enhancement of the PRONOM file format registry.[11]

The logical architecture is illustrated below.

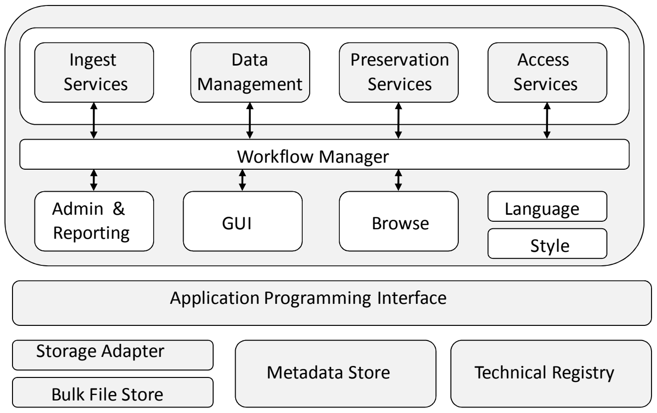

Figure 1: SDB logical architecture

Micro-Services within SDB

Micro-services provide the backbone of the functionality within SDB. Each step within a workflow can execute one or more micro-services. A micro-service may be fully automated, or require some user input. An example set of current micro-services used within SDB is illustrated in the table below.

Micro-services can be invoked at any point in the lifecycle of content and thus a single micro-service can be present in multiple workflows. As an example, the characterization services are executed as content is ingested, (ingest workflow) and may be reused during a preservation process to characterize any output of the process (preservation workflow). Similarly, fixity checks are used at ingest and also to enable on-going integrity checking.

Table 1: Examples of SDB micro-services

Major Function	Sub Function	Micro Services
Ingest	Quality Assurance	• Integrate to content source • Upload SIP • Transform Schema • Fixity Check • Virus Check • Metadata Consistency check • Content Consistency Check
Storage		• Store Files • Store Metadata • Update Index • Storage adaptor* • Integrity checking
Preservation	Characterization	• Format Identification • Format Validation* • Property Extraction* • Component Identification
	Planning	• Identify files at risk • Select preservation pathway
	Action	• Migrate file* • Emulate file* • Migration validation*
Data Management		• Edit Metadata • Approve Changes
Access		• Create DIP • Export content
Reports		• Create new report • Execute report

An asterisk in the table above indicates that there is a family of micro-services that provide the functionality. For example SDB provides a number of migration services that are each file format specific.

Micro-services can be implemented in a number of different ways. They can be a combination of custom code, open source products and commercial off the shelf (COTS) tools. This allows the use of best of breed approach to fulfill a specific function and somewhat reduces any specific technology requirement. However, all micros-services require simple wrapping with a web services interface so that they can be made available within the system.

A sample of SDB micro-services is illustrated in the following table:

Table 2: Example micro-service implementations

Micro-Service	**Implementation**	**Functionality**
Format Identification	DROID –open source	Identifies file formats. Returns a PRONOM identifier
Property Extraction for PDF	JHOVE – open source	Returns a set of extracted properties from a specified PDF file
Property Extraction for AudioVisual	MediaInfo – open source	Returns a set of extracted properties from a specified audio or video file
Component identification	Custom code	Identifies conceptual components of web based content
Format Migration	Stellent COTS	Migrates a file format from one technical representation to another

Embracing new micro-services

As existing implementations mature, and new tools become available the principle benefits of a micro-services approach begin to be realized. For example the JHOVE tool has recently been re-architected and now provides a more flexible and extensible approach. This tool can simply be made available to SDB workflows, by replacing the existing JHOVE service and maintaining the interface.

Importance of workflow

The presence of micro-services alone is not sufficient to constitute an automated digital archiving solution. As previously mentioned the micro-services have to be orchestrated together to implement a range of business processes. A principle approach for this is to make use of a workflow engine. There are many existing examples of workflow engines, many of which are open source, hence there is no real need to custom build one. Workflow standards such as BPMN [10] are now maturing and gaining wide adoption. Such workflow engines provide more than just the ability to execute workflow, they provide a comprehensive set of management features including:

- Ability to start, pause, restart and terminate workflows.
- Ability to execute multiple workflows concurrently
- Ability to maintain the state of workflows, allowing seamless restart in the event of a server crash.
- Framework for handling exceptions generated by individual workflow steps
- Flexible triggering of workflow execution – manual, automated and event based

The Drools Flow workflow engine was chosen for use in SDB for a number of reasons:

- Ease of deployment and simple API
- Ability to handle workflows that have both automated and manual steps (GUI based)
- Ability to ingrate business rules within workflows
- Ability to define workflows through a graphical user interface
- High level of adoption

The following diagram illustrates the workflow architecture as used within SDB. The workflow engine resides on the main SDB server, and workflow steps (hence micro-services) are executed on one or more job queue servers. All workflow steps that require manual input are executed on the main SDB server and normally require interaction with a web page or form.

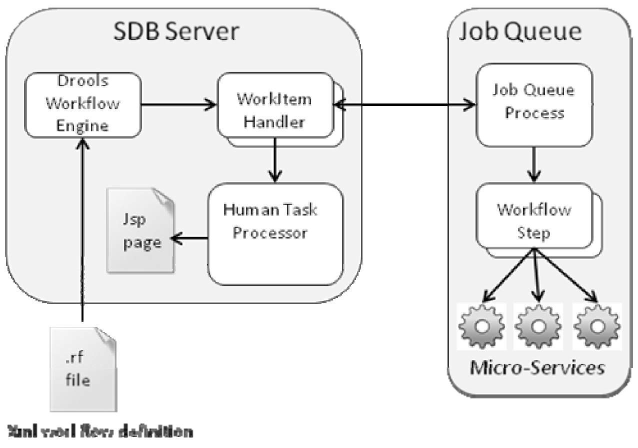

Figure 2: Workflow architecture

Within Drools Flow, workflows are defined as xml files. Each xml file describes the set of workflow steps their sequencing and a list of parameters that each step expects. A graphical user interface can be used to define a workflow, by dragging and linking components onto a canvas. Available components include workflow steps (automated and human), split and join points, parallel flows, decision points, wait points and loops. It is also possible to invoke the execution of other workflows from within an existing workflow step. An example of an ingest workflow that contains parallel paths and decision points is illustrated below. Note that the Characterization step actually consists of a set of micro-services that when used together constitute a characterization step.

Using this approach it is possible to create a set of workflows that fulfill a similar business process but operate on very different content types that may come from a variety of sources, requiring different policy decisions. For example a set of ingest workflows can be deployed that provide the following capabilities:

- Harvest and ingest web content

- Ingestion of content from a document management system
- Fully automated high throughput ingest of images

These different workflows will share a number of workflow steps in common but will integrate them with task-specific steps and then combine them in a unique orchestration. Since most common steps now exist adding a new workflow usually only involves writing one or two (if any) new steps, registering these with the system and then combining them in the drag-and-drop tool to create the appropriate flow of steps.

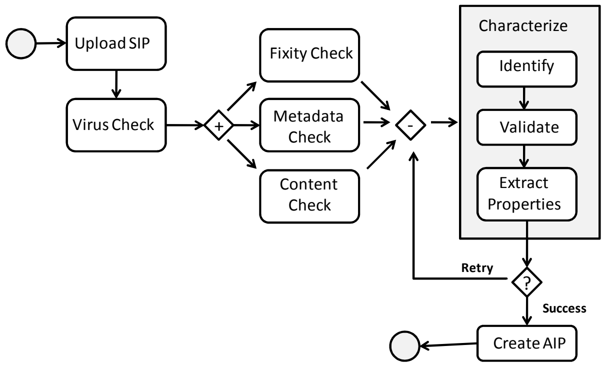

Figure 3: Example ingest workflow

The Drools Flow workflow tool has been extended in SDB to provide the following additional features:

- Extended parameterizations – Once registered the same workflow can be run in a series of "workflow contexts", each with different parameters
- Rule based issue handling, including logging and continuation, workflow pausing for human intervention and automated termination with notification. This is especially important since it allows decisions on how to handle an issue in a step to be postponed until the workflow context is clear. It also allows important audit trail information to be automatically maintained (e.g., inability to complete characterization owing to lack of an ideal tool) without unnecessarily informing humans.
- Provision of management dashboard to monitor the execution of workflow, and workflow status (figure 4).
- Use of a job queue system to ensure that workflow steps are load balanced, ensuring efficient use of multiple servers.

Preservation – An Example of a Micro-services Approach

The OAIS model describes only some of the features needed to manage the process of dealing with technical obsolescence. A number of projects have defined a more complete preservation process that includes the following main steps:

- **Characterization** – This consists of 2 components - The capture of technology-dependent properties of the content files, and the capture of technology-independent significant properties of the information object. The former is used to determine if something is in need of attention, and the latter is used to determine the essential characteristics that should be preserved
- **Preservation Planning** – Determines what is obsolete and the most appropriate action to take.
- **Preservation Action** (Migration or Emulation) – Performs the actions specified in the preservation plan, re-characterizes the output and determines the success or failure of the action

To enable this SDB extends the OAIS model with unique, automated "Active Preservation" functionality that enables long-term preservation. It allows new manifestations of information objects to be created both at ingest (if required) or after ingest. In either case, all manifestations are retained in the repository. In order for any active preservation to take place, a set of preservation policies and standards must be in place. These policies are manifested inside the technical registry along with factual information about file formats.

Characterization

Characterization can be performed at a batch level (e.g., across a whole accession during ingest) or at an information object level (e.g., as a result of a preservation action). In all cases this is a multi stage process involving

- Technology dependent characterization of every file
- Technology independent characterization at an informational object level

In the case of technology dependent characterization the following set of micro services is executed:

- **File format identification** – To achieve this, a request is made to the Technical Registry to return the preferred format identification micro-service. The current preferred tool is DROID which is wrapped and deployed as a micro-service. Since no one tool is able to uniquely identify every known format, warning messages may be returned from this step.
- **Format validation** – For each identified, a further request is made to the Technical Registry to return the preferred micro-service to perform format validation for that format. The format validation step checks that the file in question confirms to the specification it purports to be.
- **Property extraction** – Every file has key properties extracted. Some of the properties are common for all files e.g., size, last modified date, but the majority of properties are format-specific, and are extracted by means of a format-specific tool. A request is made to the Technical Registry for the set of properties to extract, and the preferred micro-service to be used. In some cases there may be more than one micro-service that is can be used for a particular format.

Most importantly, and uniquely in SDB, characterization also occurs at the information object level. This is important since the number and structure of files can change from each technical representation resulting from a preservation action and thus it is not possible to validate a preservation action just by comparing file properties. Hence, SDB detects the presence of technology-independent “components” (e.g., a document) and records its properties regardless of whether it is, say, a PDF with embedded images or a Web page consisting of multiple HTML, CSS and image files. The existence of these components and their properties are considered to be the “significant properties” of the information objects and thus the set of characteristics that need to be preserved. Although this process is attempting to measure technology-independent characteristics it needs to do this from the technology of the original information object. Hence, the tools that need to be used are format-dependent and thus once again the characterization framework asks the Technical Registry to determine which is the most appropriate micro-service to use.

It should be noted that characterization is implemented as an extensible framework. With its current toolset, it can’t deal with every format but it can easily be extended to utilize new tools that can perform some part of the characterization process on other formats or improve on the characterization on formats where there is an existing tool.

Preservation Planning

Preservation planning is the act of determining what information is at risk, and defining the actions that need to be taken to mitigate the risk. A preservation plan requires the following information:

- **Preservation Plan Type** – Three types of preservation planning are currently available: Container extraction to remove the risk of files that may be in a compressed container format; Presentation migrations and Preservation migrations needed because the information object is manifested in a technology that is not suitable for archiving.

- **Obsolescence Criteria** – Either one or more of file formats can be explicitly specified as being at risk, or a risk threshold can be set. Each format in the technical registry has an associated risk score. This score is derived from a set of identified sustainability factors and can be set according to a users policy. If a risk threshold is specified then all formats with a risk score greater than the threshold are included.

Once the plan type and obsolescence criteria has been set, a micro-service is executed to determine the set of files and hence information objects that are at risk and in need of a preservation action. One or more information objects can be selected for receiving preservation action.

The final step in defining the preservation plan is to identify the preservation pathway. A request is made to the Technical Registry to provide the set of available preservation pathways for each of the formats that has been identified to be at risk. Each preservation pathway consists of a target format and a tool that supports the migration to the target format. Each preservation pathway tool is wrapped as a micro-service.

Preservation Action

Now that the preservation plan has been defined it can be executed in the form of a preservation action. The primary mechanism for preservation action is file format migration. Upon execution of the preservation plan, the tool specified by the preservation pathway is executed and a new technical representation of each information object covered by the plan is generated. Once complete the new technical representation undergoes characterization at both the technical and informational object level. This enables the technology independent properties of the old and new representations to be compared, and hence informs the success of the preservation action event. In addition it is possible to run additional micro-services to determine if there has been any significant changes in the actual content of the two representations. e.g. after an image migration a check can be made to determine if there has been a significant shift in the distribution on colors.

In summary, SDB has a flexible and extensible active preservation capability that is facilitated by the use of the micro-services concept. This allows automated preservation to occur with administrators controlling the policy information stored in the Technical Registry.

Community Driven Approach to Continued Development

The concept of micro-services has enabled SDB’s functionality to grow quickly and easily through a number of initiatives. This means more functional choices are available to the community (and in some cases means there is a choice in how to achieve a given function, e.g., a choice of migration tools to go from one specified format to another specified format).

One growth method is via the SDB user community. This has continued to grow over the years and is increasingly involved in the active development of the future SDB road map. A user group has been established which meets for a 2-day meeting on an annual basis. Participants have the opportunity to share experiences and discuss functionality that they would like to see included in future releases of SDB. This influences the future road map and thus new features are added in accordance to the community’s desires.

Another method is through the continual improvement of the working systems of the participant organizations. SDB users are encouraged to share any new workflows, workflow steps and micro-services that they may have developed for their own use. In addition to the user group, a set of user discussion forums has been established. This allows users to continue discussions concerning future technology outside of the regular meetings.

Finally, the addition of new organizations with new needs provides further growth. These new needs means new micro-services are created and are made available to the whole community.

References

[1] Consultative Committee for Space Data Systems (2002). Reference Model for an Open Archival Information System (OAIS). CCDS 650.0-B-1, Blue Book http://public.ccsds.org/publications/archive/650x0b1.pdf

[2] Planets Project – http://planets-project.eu

[3] SBD current users; retrieved March 15 2011 from http://www.digital-preservation.com

[4] Abrams,S., Kunze, J., Loy, D., (2009) An Emergent Micro-Services Approach to Digital Curation Infrastructure, iPress 2009 conference presentation

[5] Abrams,S., Kunze, J., Loy, D., (2010) An Emergent Micro-Services Approach to Digital Curation Infrastructure, *The International Journal of Digital Curation Issue 1, Volume 5*

[6] Reference Model for Service Oriented Architecture 1.0 OASIS Standard, 12 October 2006; retrieved March 16 from http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

[7] iRODS Micro Services; retrieved March 7 2011 from https://www.irods.org/index.php/iRODS_Micro-services

[8] University of California Merritt repository; retrieved March 1 2011 from http://www.cdlib.org/services/uc3/merritt/index.html

[9] Archivematica; retrieved March 3 2011 from http://archivematica.org/wiki/index.php?title=Micro-services

[10] BPMN specification; retrieved March 10 2011 from http://www.bpmn.org/

Author Biographies

__Mark Evans__ is the Digital Archiving Practice Manager for Tessella Inc, and provides oversight for all of Tessella's archiving projects in North America. Mark has been involved in the specification, architecture and design of digital archiving solutions for the past 8 years, and has worked on several national programs including the NARA ERA program and the NSF funded DataNet program.

__Bill Steel__ is the Technical Lead and and a senior software engineer in Tessella Inc's Rockville, MD office. Bill has carried out development work on Tessella's SDB digital archiving platform and the NSF funded DataNet program. He is also a PMP certified senior project manager within Tessella Inc managing a variety of projects including management of Tessella's staff on the NARA ERA digital archiving progam.

__Robert Sharpe__ is Tessella's Head of Digital Archiving Solutions. Robert Sharpe has worked in the IT industry for over twelve years of which he has spent most of the last seven years project managing projects for leading national archives and libraries. These include the specification and development of a number of systems for the UK National Archives including the Digital Archive (winner of the inaugural DPC/Pilgrim Trust award for digital preservation), PRONOM (now its sixth He is responsible for the development of the Safety Deposit Box system

__James Carr__ is a system architect for SDB. James has been involved in a number of Digital Archiving engagements include the Swiss National Archives. James has supported SDB clients for both Tessella Inc and Tessella plcfrom both a training and consultancy perspective

__Alan Gairey__ is a system architect for SDB, focusing on the workflow framework and integration with external systems using open protocols. He has been involved in digital archiving solutions for over 5 years, and has recently been involved in the design and implementation of archives for two major European memory institutions.

__Jon Tilbury__ is the Director responsible for Tessella's Abingdon Division. Jon has been with Tessella for over 20 years and has worked in many roles within the company including programming, system design, project management, technical management and business development in many different industry sectors.