Developing a workflow System that integrates scientific development

Daniel A Silva Soto[1], Steven Wood[2], David Jones[2], Rod Hose[1]

[1]The University of Sheffield, [2]Sheffield Teaching Hospital

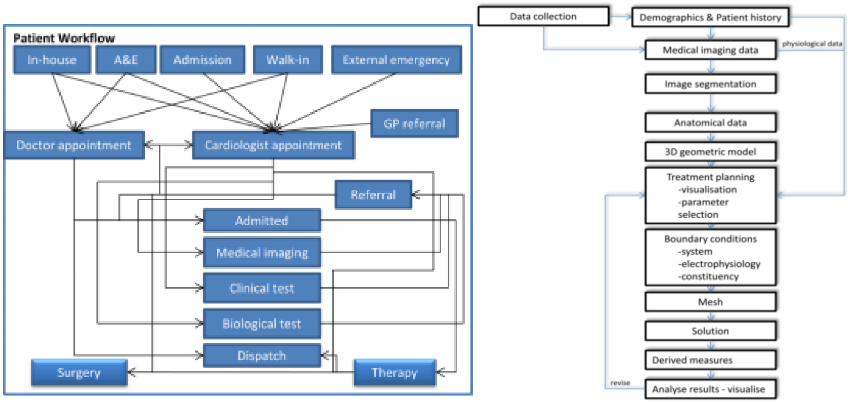
Correspondence: d.silva@sheffield.ac.uk; Beech Hill Road, Sheffield, S10 2RX, UK

A major challenge in health informatics is narrowing the gap between research and the clinical deployment of new technology; this concept is often coined as translational research. However, it is the inherent methodology differences between the scientific and clinical domains which make the translation of tools technically difficult.
Collaboratory projects are shifting towards being workflow-centric as investigators are finding added value to integration of tools, data formats and provenance. Research workflows are heavily data-driven and clearly structured. However what is also required in clinical applications is a good mechanism for human interaction. In "business-orientated workflows", event-driven workflows are control-centric, organise operating principles nicely and deal with human interaction much better.
Scientific workflows often maintain the state of the workflow permanently in computer memory whereas business workflows will be persisted to disc and remain "dormant" until reactivated by an external event or timer. The event will typically take a short time to process before the workflow is again persisted to disc. This model facilitates scalability in the number of workflows to a clinical scale. Our effort concentrates around the challenge of introducing biomedical modelling workflows into the clinical decision making protocol.

Clinical pathways are best modelled using a state machine - this is because they are long running workflows with no pre-established order of events. On patient arrival, little can be said on what the illness might be and what treatment will be used. On the other hand, scientific workflows are modelled as sequential workflows, being highly data driven with carefully ordered processing steps and a more predictable time-span.
Within the constraints of a sequential scientific workflow framework, the steps themselves will often change rapidly as the research problem is explored and understood better. For clinical application, the protocol would be standardised, but still provide for the possibility of future adaptations and improvements as new research informs the process. One of our goals is to show how the design of a patient workflow can remain fairly static if decoupled from individual tasks.
Whilst sequential tasks may form part of a state machine workflow, the opposite is not true in a realistic implementation. Additionally, sequential tasks are little different from sequential workflows. An event-driven clinical workflow can be used to initiate new scientific workflows for long-running data intensive tasks. This allows decoupling of the problem domains. The high-level clinical decision making process can evolve separately to the technical simulation process.
A workflow engine was built with the patient-focus and the ability to start sub-workflows decoupled from the engine core. This approach allows for enterprise level scalability as well as human-control over workflow logic. The engine stores information in an SQL database, including running workflow information as well as an extensive audit trail.

The workflow engine consists of a main host, a user interface, a set of workflow services and an SQL database. The main host executes individual runtime instances, handles events; it is mostly decoupled from the services that allow interaction. The user interface allows visualisation of human-tasks and interaction with process flow controls. The obvious choice to handle web service calls was a web page. Workflow services act as an extension to the host; one does not communicate with the host but through services that channel calls. Services make use of the communication framework to raise events, allow threading to run several instances in active memory at a given time and perform read/write actions on the database. The SQL database stores workflow types, tasks and rules, and the information of the executed task (active, paused, terminated or completed workflows). Individual workflow types might subscribe to service to make dependent information persistent.

The engine is built using the Microsoft Workflow Foundation (WF) libraries, as these were found to allow a generic workflow-agnostic engine to be developed, whilst including the best solution for workflow persistency. In addition, it integrates with hospital systems, including security and databases systems.
This approach also decouples workflow orchestration from clinical practice; a modelling workflow gets triggered by the existence of certain medical data and creates a report that adds knowledge to the diagnosis. The modelling workflow is executed asynchronously and does not interfere with the patient workflow. In addition to proper anonymisation tools, this scheme allows easier and more integrated sharing of clinical information.
The most widely used workflow authoring languages have a similar background (mainly UML) allow converting existing workflows into an XML format that can be validated by the workflow engine.
The fact that all calls are web service based allows any existing tool (executable) to be converted into a service that exposes it to the workflow engine; this further decouples the engine from workflow tasks.



Sequential workflows are not suitable for complex patient workflow (left)
but they are appropriate for data-driven modelling workflows (right)

This work demonstrates a successful implementation of true collaborative investigation between a teaching hospital and a research group. The main advantage of this project is a truly translatable and scalable workflow engine that will be able to perform in a real hospital environment. This has been achieved by creating a data-agnostic environment that focuses on the long-running patient workflow and allows execution of data-driven workflows. This approach contrasts with other attempts at translating biomedical research to clinic in the enterprise-centric approach and division of patient-flow and tools.
The next step in development of the engine is to increase the clinical departments and other scientific groups who use it in order to investigate it's flexibility upon new clinic management and tools.

Acknowledgements
This work has been partly funded by EPRSC Grand Challenge.

References
1. Curcin, V and Ghanem M. Scientific workflow systems - can one size fit all?. Biomedical Engineering Conference, 2008. CIBEC 2008. Cairo International. 2008.
2. Goble, C. Position statement: Musings on provenance, workflow and (semantic web) annotations for bioinformatics. 2002.
3. Hasman, A. Challenges for medical informatics in the 21st century. International Journal of Medical Informatics, 1997. **44**(1): p. 1-7.
4. Haux, R. Medical informatics: Past, present, future. International Journal of Medical Informatics, 2010. **79**(9): p. 599-610.