



International Conference on Computational Science, ICCS 2010

An Abstract Virtual Instrument System for High Throughput Automatic Microscopy

A.B.M. Russel^{a*}, David Abramson^a, Blair Bethwaite^a, Minh Ngoc Dinh^a, Colin Enticott^a, Stephen Firth^b, Slavisa Garic^a, Ian Harper^b, Martin Lackmann^b, Stefan Schek^c, Mary Vail^b

^a Faculty of Information Technology, Monash University, Clayton, Victoria-3800, Australia

^b Faculty of Medicine, Monash University, Clayton, Victoria-3800, Australia

^c Leica Microsystems, Am Friedensplatz 3, Mannheim, D68165, Germany

Abstract

Modern biomedical therapies often involve disease specific drug development and may require observing cells at a very high resolution. Existing commercial microscopes behave very much like their traditional counterparts, where a user controls the microscope and chooses the areas of interest manually on a given specimen scan. This mode of discovery is suited to problems where it is easy for a user to draw a conclusion from observations by finding a small number of areas that might require further investigation. However, observations by an expert can be very time consuming and error prone when there are a large number of potential areas of interest (such as cells or vessels in a tumour), and compute intensive image processing is required to analyse them. In this paper, we propose an Abstract Virtual Instrument (AVI) system for accelerating scientific discovery. An AVI system is a novel software architecture for building an hierarchical scientific instrument – one in which a virtual instrument could be defined in terms of other physical instruments, and in which significant processing is required in producing the illusion of a single virtual scientific discovery instrument. We show that an AVI can be implemented using existing scientific workflow tools that both control the microscope and perform image analysis operations. The resulting solution is a flexible and powerful system for performing dynamic high throughput automatic microscopy. We illustrate the system using a case study that involves searching for blood vessels in an optical tissue scan, and automatically instructing the microscope to rescan these vessels at higher resolution.

© 2010 Published by Elsevier Ltd.

Keywords: Abstract Virtual Instrument System; Scientific Workflows; Automatic Microscopy; Grid Computing; Cancer Research.

1. Introduction

Modern therapeutic drug discovery increasingly requires observing biological processes at a very high resolution. For example, high-resolution observations are important in studying cell dynamics, such as how tumours create blood vessels that facilitate tumour growth. This type of scientific research requires repeated scanning of cultured

* Corresponding author. Tel.: +61-3-99020594; fax: +61-3-99020599.

E-mail address: abm.russel@infotech.monash.edu.au.

tumour samples at different resolutions in order to observe both the vasculature develop in the first place, and be destroyed as a result of drug treatment.

Existing commercial microscopes (and other imaging systems) behave very much like their traditional counterparts, where a user controls the microscope and chooses the areas of interest manually. An area of interest could be an individual cell or a cross-section of a vessel in a tumour specimen. This mode of manual discovery is suited to problems where it is easy to find a small number of areas that might require further investigation in order to draw a conclusion from observations. However, observations by an expert can be very time consuming when there are a large number of potential areas of interest, and compute intensive image processing may be required for each image that is captured.

Increasingly, scientists are interested in multi-modal instruments that observe one sample using different modalities. This allows them to observe a biological process more completely because each modality is suited to particular functions and operates at different resolutions and scales. In such case, a virtual multi-modal instrument could actually be built from multiple physical instruments, using software to provide the illusion of a single virtual instrument. In order to realise a virtual instrument, however, a new architecture is required for combining instrument control and parallel analysis software in a seamless, powerful and flexible way.

In this paper we propose a flexible and powerful Abstract Virtual Instrument (AVI) system using scientific workflows. An AVI system is a novel architecture for building an hierarchical scientific instrument – one in which a virtual instrument could be defined in terms of other physical instruments, and in which significant processing is required in producing the illusion of a single virtual scientific discovery instrument. Section 2 describes motivation behind this research and background. Section 3 discusses the abstract virtual instrument system. In section 4, we illustrate a case study by implementing an AVI prototype on an integrated scientific workflow platform for microscopy. The system is built on top of the Kepler workflow system [7], using specific components that control the microscope, store images and automatically identifies areas of interests to rescan. We also discuss the implemented prototype use case using a biological experimental design followed by conclusions.

2. Motivation and Background

Modern microscopes allow us to monitor individual molecules at high resolution. These instruments will generate massive amounts of data over the next 5 to 10 years, allowing us to answer complex questions underpinning scientific research. For example, modern cancer research involves a variety of experiments such as observing drug delivery and tumour physiology, observing the host-tumour interaction that affects tumour physiology and therapeutic responses, and observing blood flow in a vessel within the tumour itself. Blood flow in a tumour is important in considering tumour growth, detection and treatment. One important step in such research is to find areas of interest from an initial scan to reduce search space and analyse these in more detail. For a specific study, scientists begin observation with low-resolution ‘tile’ scans (where the instrument stitches together multiple independent images) to cover a large area on the specimen. They then identify areas of interest for further observation and instruct the microscope to take higher resolution images of these using optical zooming (possibly involving addition wavelengths, or channels). This is an iterative process until the scientist reaches to a conclusion. Based on the type of study, and what scientists are looking for, high-resolution microscopes are used for further detailed study on identified areas of interest. However, finding interesting positions within a cancer tissue scan currently requires expert observation and the search process is time consuming and error prone when there are a large number of potential areas of interest. This search process is data intensive and also requires compute intensive image processing capabilities. A high throughput automatic microscopy approach could identify large areas of interest in a specimen to observe these positions at high resolution.

However, at present, commercial microscopes are predominantly stand-alone instruments, controlled by dedicated computer systems that can only provide limited storage and processing capabilities. Routinely, the experimenter interacts directly with the microscope, which captures and stores digital images on a local computer disk. The data is then transferred manually to various computer and storage devices for analysis, mining, archiving and visualization. Increasing data volumes are creating a bottleneck in which the processing times exceed the data acquisition times, significantly impeding real time observation. Modern image analysis techniques (such as volume rendering and surface reconstruction), and data mining software, require physical or virtual multi-processor systems for adequate performance and large data stores for file storage, but these require a range of ad-hoc and complicated

methods involving meta-data capture, file transport, job submission and data archiving and visualization.

In our earlier work [2], we demonstrated usage of scientific workflows for microscopes based discovery. Unlike the earlier work, here we define an Abstract Virtual Instrument that achieves high-throughput microscopy by combining the process of image capture, processing and reimagining in an integrated workflow platform. We illustrate our system with a case study in searching for blood vessels in an optical tissue scan, and automatically instructing the microscope to rescan these vessels at higher resolution. The system allows novel and flexible ways of acquiring biological knowledge.

2.1. High throughput microscopy screening system

Enormous potential exists in high content and high throughput microscopy [18]. High throughput microscopy involves the automation of image capture, data storage, processing and reimagining where necessary. Data from experiments are stored in archival storage systems and data analysis is performed using connected sophisticated image analysis software. High content screening system is an automated cell biology method [21], and allows rapid acquisition of images without human intervention.

Recently, Leica Microsystems, in collaboration with EMBL (European Molecular Biology Laboratory), developed a tailored software system called Computer Aided Microscopy (CAM). Using a matrix scan mechanism, CAM allows time-lapse fluorescence microscopy on live cells and instructs the same microscope to reimage a specimen with different parameters, and promises to revolutionize high throughput microscopy [14]. CAM integrates microscope, data storage systems, computing clusters and robotics in a flexible screening environment. The Mitocheck project [12], the largest integrated project on cell cycle control within the 6th Framework Programme (FP6) of the European Union, builds on this work and developed an automated pipeline for high-content RNA interference (RNAi) screening (RNAi is a powerful tool to study gene function in cultured cells). The developed system automatically processes images to search for particular attributes and patterns (for example, cell mitosis). In this paper we propose an Abstract Virtual Instrument system and implement a prototype that builds on high throughput microscopy screening systems. Our system allows workflows that are of arbitrary complexity, with arbitrary operations on the data involving data capture, management and image analysis.

2.2. Scientific workflows

Scientific workflows enable the specification of experiments involving a range of different activities, such as data integration, modeling and analysis, and visualization across a range of distributed resources, and leverage Grid computing middleware and approaches. Existing scientific workflow tools such as Kepler [7], Triana [20], Taverna [19], Knime [8], and Labview [9] (to name only a few) use a graphical programming environment to specify and manage the executions. Other systems, such as Swift [23], Karajan [6], and DAGMan [3] (to name only a few) achieve similar outcomes using scripting languages. A scientific discovery process such as drug discovery [10] may involve extracting data from a scientific instrument, use high performance cluster for data analysis, use distributed data repository for storage, and then visualize them on particular display technologies.

Scientific workflow tools are powerful. They are also flexible and support high performance computing with parallel execution [1]. A typical scientific workflow system, Kepler, is built upon a Java based software framework called Ptolemy II [15]. In Kepler, models are built based on the assembly of pre-designed components (called actors) [11]. An actor is an encapsulation of parameterized actions performed on input data to produce output data. Ports and parameters are the interfaces of an actor. A framework is an environment that actors reside in, and defines the interaction among actors. A Models of Computation (MoC) defines the communication semantics among ports and the flow of control and data among actors. Directors are responsible for implementing particular MoCs, and scheduling and overall execution semantics of a workflow. Because of the cost and complexity of building virtual instruments, we have built our system on top of an existing scientific workflow engine. However, scientific workflow engine must have the ability to integrate control system and analysis software to realise a single virtual instrument. In the next section we show how a workflow system can be used to implement the execution logic of an Abstract Virtual Instrument.

3. An Abstract Virtual Instrument System

In this paper we introduce a novel architecture for building an hierarchical scientific instrument – one in which a virtual instrument could be defined in terms of other physical instruments, and in which significant processing is required in producing the illusion of a single virtual scientific discovery instrument. For example, combining a laser scan microscope or an optical scan microscope with a heating apparatus for a study requiring heating the specimen periodically will produce a virtual instrument called, say, a “heating microscope”. Further, including image-processing software to analyse different experiment conditions will produce a virtual instrument of an image processing heating microscope. An Abstract Virtual Instrument (AVI) system synthesises multiple physical instruments of similar types or multi-modal types with multiple analysis software.

Scientific discovery process with complex compositions of hierarchical parallel and distributed processes needs a powerful scheduling strategy to coordinate scientific instrument control and analysis processes on the fly across Grid infrastructure. Scheduling such processes on Grid infrastructure is significantly challenging, as tight integration is required between instrument control and analysis processes. Besides, modern event driven approaches feed acquired data into analysis pipeline after the completion of the entire data acquisition phase [22]. There is also a need to address the requirement of reproducing time consuming and resource intensive experiment conditions for multiple critical observations. In contrast to existing approaches, an AVI system aims to accelerate the scientific discovery process by using a workflow engine as experiment specification and execution environment addressing the complexities of the Grid infrastructure

3.1. Formal Definition of an AVI system

An AVI system specification can form arbitrary compositions of instrument control and analysis software. Such compositions may require significant data processing. For example, producing an array of cells or vessels positions from a microscopic tissue scan would require compute resources running image segmentation software. We formally define an Abstract Virtual Instrument (AVI) system as a software infrastructure that builds a hierarchical scientific instrument. It combines scientific instrument control system and analysis software to provide access to a new set of physical or virtual scientific instruments with high-end data processing capabilities. Importantly, we leverage Grid infrastructure to provide a rich range of platforms, from distributed data to computational services. An AVI system is a concurrent system in terms of execution sequences specified within a workflow engine execution environment. An AVI can be specified in terms of controls, infrastructure and communications layers. Each layer is represented by input, set of operations in each layer and output. Layers of abstractions are essential in AVI to simplify the architecture and to encapsulate each layer without compromising generality. Synthesis of instrument control system and analysis software system requires an abstract specification of the AVI system and provides incomplete implementation to derive a complete implementation that satisfies the specification. We formally model an AVI system using Calculus of Communicating Systems (CCS) process algebra [13]. We illustrate the model using instrument control (IC) system and analysis software (AS) system that interact using communication interface z satisfying AVI system specification:

$$AVI ::= IC + AS \mid IC \parallel AS \quad (1)$$

AVI can act either as IC or as AS otherwise put IC and AS in parallel.

$$IC ::= 0 \mid z?ic \mid z!ic \mid ic + ic \mid ic \parallel ic \mid zic \quad (2)$$

IC can be 0, input on z and proceed as ic ($z?ic$), output on z and proceed as ic ($z!ic$), act either as $ic1$ or as $ic2$ ($ic+ic$), put $ic1$ and $ic2$ in parallel ($ic\parallel ic$) and treat z as local channel and visible only in ic (zic).

$$AS ::= 0 \mid z?as \mid z!as \mid as + as \mid as \parallel as \mid zas \quad (3)$$

AS can be 0, input on z and proceed as as ($z?as$), output on z and proceed as as ($z!as$), act either as $as1$ or as $as2$ ($as+as$), put $as1$ and $as2$ in parallel ($as\parallel as$) and treat z as local channel and visible only in as (zas).

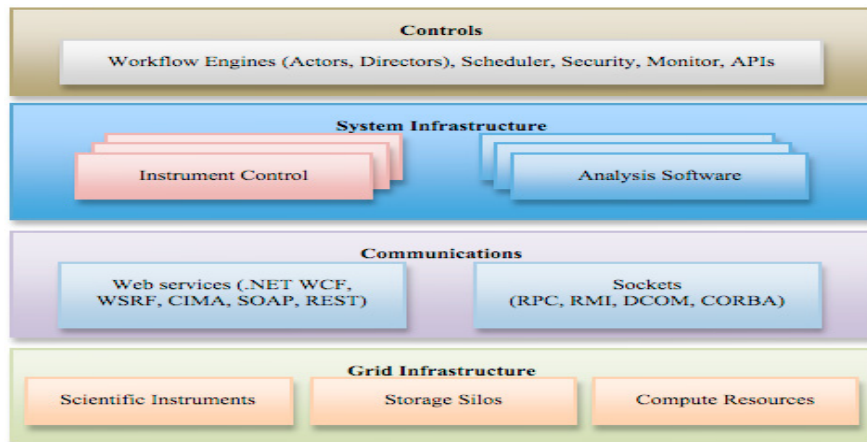


Fig. 1. Architecture of an Abstract Virtual Instrument (AVI) system.

3.2. Architecture of an AVI system

An AVI system abstracts the details of underlying scientific instruments operations for the users who no longer need knowledge of, expertise in, or control over the complexity in building virtual instrument. Our proposed AVI system has the following layers as illustrated in Figure 1:

- Controls:** This layer provides workflow engine with actors and directors, scheduler, security, monitor and APIs to provide experiment specification and execution environment to facilitate coordination among components.
- System Infrastructure:** This layer consists of arrays of instrument control system with low-level device drivers and arrays of analysis software system.
- Communications:** This layer facilitates communications in between controls and underlying infrastructure.
- Grid Infrastructure:** This layer consists of physical instruments; storage silos and compute-resources along with associated low-level middleware and distributed services.

3.3. AVI system design principles

We propose the AVI system in accordance with six design principles:

- Scalable:** An AVI system is scalable if it can perform just as efficiently on a network of instruments with pool of analysis software as it can on a single instrument with one analysis software.
- Modular:** An AVI system is modular if it is composed of two or more instrument control systems and analysis software that can be combined to form a more complex virtual instrument.
- Extensible:** An AVI system is extensible if its implementation allows system layer extension. Instrument control system or analysis software can be modified or extended without impact on existing AVI system.
- Secure:** An AVI system is secure if all interactions among AVI layers are protected.
- Platform independence:** An AVI system is platform independent if it is implemented in platform independent programming languages.
- User-friendly:** An AVI system is user-friendly if it is integrated with existing research workflow where details of underlying scientific instruments operations are abstracted from the users who no longer need knowledge of, expertise in, or control over the complexity in building virtual instrument.

We consider these six design principles to ensure that an AVI system is more flexible and powerful system to build more complex virtual instruments. In the next section we discuss a case study on implementing an AVI system prototype for dynamic high throughput automatic microscopy.

4. Case Study: Implementing an AVI System Prototype for Automatic Microscopy

An AVI system aims to accelerate scientific discovery by building virtual scientific instruments. Scientific discovery acceleration can be achieved through reducing bottlenecks exist in every step of research process. An AVI system provides an illusion of a single virtual instrument to scientist by abstracting and automating intermediate processes. As a case study, we have implemented a flexible and powerful system for performing dynamic high throughput automatic microscopy. We present an AVI system reference implementation by building a virtual instrument of a high throughput microscope that produces a stream of high-resolution images from a simple optical low-resolution scan. The resulting system automatically discards irrelevant images from a complete run.

Our prototype virtual instrument implementation builds on top of Kepler scientific workflow tool [7] to both control the microscope and perform image analysis operations to return high-resolution images meeting predefined filtering constraints. This virtual instrument system combines (according to formal definition in section 3.1) Leica Application Suite Advance Fluorescence (LASAF) microscope control system and ImageJ analysis software. Instrument control system and image processing are specified using Actors in Kepler workflow engine at *Controls* layer of our AVI system architecture described in section 3.2. Underlying Grid infrastructure includes Leica microscopes exposed as web services as well as Grid based distributed storage and compute resources.

Good scientific observation is a critical step in scientific discovery. However, scientific observations by an expert can be very time consuming and error prone when there are a large number of potential areas of interest (cells or vessels in a tumour), and compute intensive image processing is required to analyse them. These observations can be accelerated using an AVI system, which presents a virtual instrument by abstracting, and hiding complexity of underlying systems. Our prototype system identifies areas of interest from low-resolution scans and performs rescan (as an example virtual instrument) at high-resolution on identified positions. This provides an illusion of a single virtual instrument that returns a set of images meeting scientific study criteria. Similar case would be reheating (as an example virtual instrument) a sample during scientific discovery process. In this paper, we illustrate our implemented system using a case study that involves searching for blood vessels in an optical tissue scan, and automatically instructing the microscope to rescan these vessels at higher resolution.

Our system expose microscope controls through web services and implement web services on .NET platform using Windows Communications Framework (WCF), which encapsulates Sockets, DCOM (Distributed Component Object Model) and other underlying low-level communications protocols. Image processing controls are using ImageJ [5], an open source Java application to aid our reference implementation for the case study.

4.1. Microscope control

The microscope is controlled by a set of defined communication protocols. Our implementation utilizes SOAP based web services protocol to overcome limitation of other existing remote procedure call protocols such as RPC, RMI, DCOM, and CORBA. Microscope control via scientific workflows involves a subset of complete sophisticated controls that are exposed via web services and published using Web Services Description Language (WSDL). Our implementation includes the following pure virtual control actors:

- Set Data Store: This actor allows user-supplied setting of virtual data storage location. This location can be a network shared storage system mounted path or a FTP directory.
- Initiate Microscope: This actor initiates all critical hardware parameters.
- Initiate Focal Point: This actor resets stage position (initial focal point) according to the supplied value pair (X, Y positions) for scan.
- Start scan: This actor triggers the scan process.

4.2. Image processing control

Detecting patterns in biology often requires segmentation of microscopy images. In our case study, to identify areas of interest from a low-resolution specimen scan, our implementation includes the following pure virtual processing actors:

- Create Virtual Tile: This actor produces a virtual tiled image. A tiled image is a matrix formation of a collection of images.

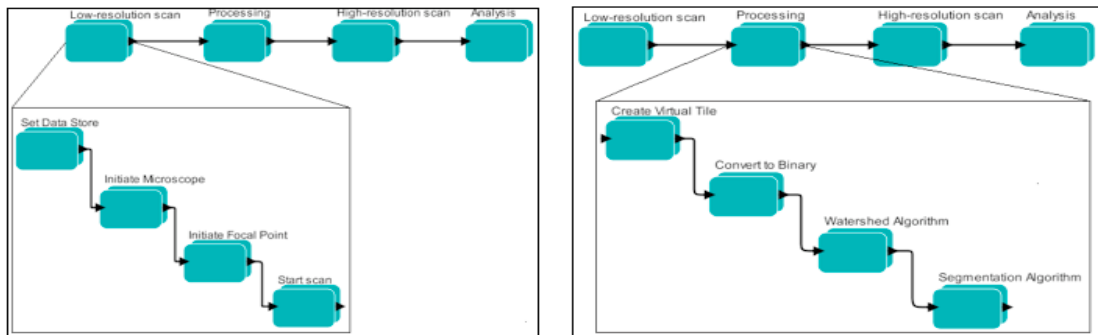


Fig. 2. Workflow of scan phase (left); Workflow of processing phase (right).

- **Convert to Binary:** This actor converts an image to 8-bit binary format for analysis.
- **Watershed Algorithm:** This actor applies watershed algorithm [16] to the given binary image.
- **Segmentation Algorithm:** This actor applies segmentation algorithm utilizing threshold technique [17] and returns a set of coordinates meeting threshold criteria set by user.

4.3. Biological Experimental Design

In this section we describe a biological experiment design based on our implemented prototype system for automatic microscopy. We aim to search for blood vessels in an optical tissue scan, and observe them at higher resolution to study the environment of cell in a tumour tissue in particular interest of identifying whether antibody has passed the blood vessel that may affect cancer growth. As part of a pre-clinical therapeutic trial, we take a xenograft from a mouse treated with the anti-EphA3 antibody and stain blood-perfused vessels. We remove the tumour from mouse and then cut the frozen tumour in 6 μ m sections to mount on microscope slide and stained to identify nuclei (blue) and blood vessels (green) to analyse on a Leica AF6000LX microscope. We initiate the tile scan at lower resolution (10x) to identify nucleus and vessel signals using our implemented dynamic microscopy prototype system. The objectives of running this biological experiment is twofold:

- Bottleneck in communication and execution:** This is to assess how bottlenecks in communication and execution impede the research process. Most commercial microscopes have a static scan speed. However, scalability of analysis software depends on software design as well as on architecture of compute resources it gets executed. To achieve this objective, we measure scanning speed for both low and higher resolution scans. We also measure analysis speed for finding areas of interest to facilitate study on characteristics of blood vessels within a tumour tissue.
- Bottleneck in human interaction and reusability:** This is to assess how bottlenecks in expert observation and (absence of) reusable modules impede the research process. Observations by an expert can be very time consuming and error prone when there are a large number of potential areas of interest (cells or vessels in a tumour). To achieve this objective we run multiple iterations of analysis steps to calculate areas of interest matching specific study requirements. We also reuse common modules for both low-resolution and high-resolution scans.

4.4. Prototype Implementation & Execution

To achieve the objectives outlined above, we implement a dynamic microscopy prototype having four phases (Figure 2 shows the workflows utilized in scanning and in processing phases):

- Low-resolution scan phase:** In this phase a low-resolution tile scan is conducted to produce virtual images to get an overview of the tissue sample.
- Processing phase:** In this phase virtual tile scan images are collected and analysed to identify area of interests.

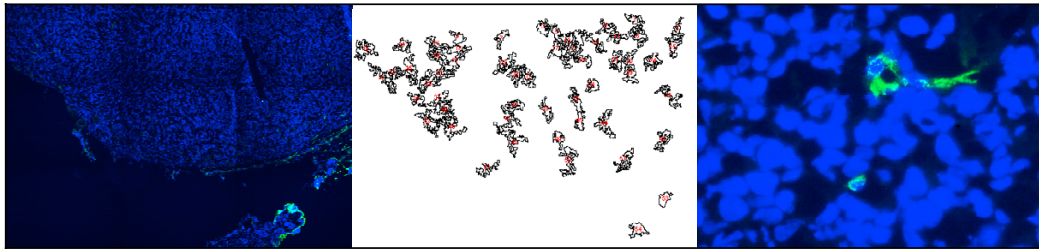


Fig. 3. (a) Low-resolution 3x3 tile scan image; (b) Areas of interests; (c) High-resolution scan image.

Table 1. Processing iterations and statistics

Iteration	Count	Total Area	Average size	Area fraction
1	1743	52338 pixel ²	30.028 pixel ²	25.709%
2	384	46295 pixel ²	120.560 pixel ²	22.740%
3	149	30239 pixel ²	202.946 pixel ²	14.854%
4	54	15519 pixel ²	287.389 pixel ²	7.623%

- c) High-resolution scan phase: In this phase a high-resolution scan is conducted based on the identified areas of interests.
- d) Analysis phase: In this phase all high-resolution scan images are analysed.

Phase one (low-resolution scan) conducts a 3x3 tile scan where each image has 2 channels, producing total of 18 images. While the scan is in progress, each section scan triggers a data format converter that reads a memory mapped file, and stores the converted data in a specified location in the real file system (along with metadata information).

Phase two (processing) produces a virtual tiled image from all converted images and to identify areas of interests. We iterate the search process till we find desired areas of interests from the virtual tiled image. Table 1 shows iterations statistics produced from the virtual tiled images. For this case study, we were interested in finding vessels areas with average size greater than 200 pixel² where area fraction is less than 10% of the reconstructed virtual tiled image. The initial iteration produces 1743 positions but did not meet the study parameters. After three iterations the system identifies 54 positions. Phase three (high-resolution scan) conducts a higher resolution (63x) scans on the identified positions to observe the condition of the identified blood vessel. Then in phase four (analysis), the system analyses all acquired higher resolution (63x) images. A complete run of all four phases of the implemented AVI system results in a virtual instrument of a high throughput microscope that produces a stream of high-resolution images whilst discarding the irrelevant images for our biological case study. In all phases, the AVI system efficiency remained consistent in all scanning and processing iterations.

4.5. Discussion

The biological case study illustrated in this paper requires significant human interactions, which could cause delay in research process. We observe bottlenecks in communication, execution and human interaction (occasionally by an expert). Communication time varies with amount of data produced in an experiment and on operations required in a study. Execution time depends on the type of infrastructure being used. Ideally, the execution time remains constant with similar set of experiment parameters. Human interaction time depends on the level of human expertise and experience. In this experiment, the manual processes are in setting up initial focus on sample, setting up tile-scan parameters and changing objective for high-resolution scans. We used instrument parameters to initialize all required microscope hardware setups in each scan phase. Remaining steps are managed automatically. The high-resolution scan phase is fully sequential because, for obvious reasons, the microscope cannot conduct concurrent scans with different experiment parameters. Figure 3 illustrates results from a complete run on a dual core 3Ghz workstation for microscopic control and on a dual core 2.66Ghz laptop for image processing.

Table 2. Summary of average time (sec) taken in each phase

Phases	Avg	Avg	Avg	Avg	Total
Phase1: Hardware setups	30 sec				30 sec
Phase1: (sec/scan) Scan - 3x3 Tile	10 sec				90 sec
Phase2: (sec/image) Collect - 9 images		5 sec			45 sec
Phase2: (sec/iteration) Process - 4 iterations		5 sec			20 sec
Phase3: Hardware setups			30 sec		30 sec
Phase3: (sec/scan) Scan - 54 images			30 sec		1620 sec
Phase4: (sec/image) Collect - 54 images				5 sec	270 sec
Phase4: (sec/image) Analyse - 54 images				1 sec	54 sec

Table 2 summarises average time taken in each phase of the experiment. There are delays both in scanning and data analysis phases. In the imaging phase, the scan process is sequential and depends on the speed of microscope. Besides, additional time is required for loading hardware parameters and changing objectives to conduct experiments at different resolutions. The image possessing in the analysis phase needs to wait for the scanning process is complete. To overcome this waiting time, our system initiates analysis steps earlier rather waiting to complete the entire scan.

Finally, we analyse our implemented system and verify whether it meets the design principles outlined in section 3.3. Our system returned a set of high-resolution images by combining Leica microscope control system with ImageJ image analysis software into an integrated workflow (satisfying *modularity* criteria) and kept efficiency consistent in all scanning and processing iterations (satisfying *scalability* criteria). The underlying Leica microscope and ImageJ software can be modified or extended without impact on existing implemented AVI system (satisfying *extensibility* criteria). This system is using Java on top of Kepler workflow tool (satisfying *platform independence* criteria). It is also integrated into existing workflow to abstract details of underlying instrument operations from the users who no longer need knowledge of, expertise in, or control over the complexity in building virtual instrument for scientific research (satisfying *user-friendly* criteria). An implemented virtual instrument based on AVI system can be represented both as a web service and as an executable for workflow engines mentioned in section 2.2. One of the advantages of AVI system is that any virtual instrument developed can be used by more complex higher-level workflows. We use web services and socket communication protocols for controlling microscope and analysis software. Communication channel can be secured at transport layer using SSL to allow secure HTTP connections. At present, the microscope control actors provide direct communication between workflow engine and control system or analysis software. We aim to extend them to allow extensive study involving complex combinations of experiment parameters. These would require specifying a common and powerful set of APIs. We aim to further extend the prototype system to improve data storage and analysis with more extensive control of microscope and with more sophisticated processing capabilities incorporating parameter sweep experiments.

5. Conclusions

In this paper, we presented an Abstract Virtual Instrument system, its architecture, and design principles along with a prototype implementation applied in life science research. We implemented a flexible and powerful prototype for dynamic high throughput microscopy system. We illustrated the prototype using a biological experiment to scan, process to identify areas of interests and then rescan for automated microscopy. Our prototype builds on the existing execution frameworks in workflow systems, and we have demonstrated its applicability in Kepler addressing new type of computation requirement within scientific research workflows. Our implemented system overcomes bottlenecks exist in scanning and analysis to speedup scientific research process as illustrated in our biological experimental case study.

Acknowledgement

This research is supported by an Australian Research Council (ARC) Linkage grant fostering collaboration between Monash University and Leica Microsystems.

References

1. Abramson D., Enticott C., Altintas I.: Nimrod/K: Towards Massively Parallel Dynamic Grid Workflows. IEEE SuperComputing 2008, Austin, Texas (November 2008).
2. Abramson, D., Bethwaite, B., Dinh, M., Enticott, C., Firth, S., Garic, S., Harper, I., Lackmann, M., Nguyen, H., Ramdas, T., Russel, A.B.M., Schek, S. and Vail, M.: Virtual Microscopy and Analysis using Scientific Workflows", IEEE e-Science 2009, Oxford, UK (December 2009)
3. DAGMan, <http://www.cs.wisc.edu/condor/dagman/>
4. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
5. ImageJ Project, <http://rsbweb.nih.gov/ij/>
6. Karajan, <http://wiki.cogkit.org/wiki/Karajan>
7. Kepler Project, <https://kepler-project.org/>
8. Knime Project, <http://www.knime.org/>
9. Labview Project, <http://www.ni.com/labview/>
10. Lang P., Yeow K., Nichols A., Scheer A.: Cellular imaging in drug discovery. *Nature Reviews Drug Discovery* 5, 343-356 (2006).
11. Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao J. and Zhao, Y.: Scientific Workflow Management and the Kepler System. *Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows* (2005).
12. Mitocheck Project, <http://www.mitocheck.org/>
13. Milner, R.: A Calculus of Communicating Systems. Springer Verlag, ISBN 0-387-10235-3 (1980).
14. Neumann, B., Held, M., Liebel, U., Erfle, H., Rogers, P., Pepperkok, R., et al.: High-throughput RNAi screening by time-lapse imaging of live human cells. *Nature Methods* 3(5), 385-390 (April 2006).
15. Ptolemy-II, <http://ptolemy.eecs.berkeley.edu/ptolemyII/>
16. Roerdink, J., Meijster, A.: The watershed transform: definitions, algorithms and parallelization strategies. *Fundamenta Informaticae* v.41 n.1-2, p.187-228 (January. 2000).
17. Sahoo, P., Soltani, S., Wong, K.C., and Chen, Y.C.: A Survey of Thresholding Techniques. *Computer Vision, Graphics, and Image Processing* Vol. 41, pp.233-260 (1988).
18. Starkuviene, V., Pepperkok, R.: The potential of high-content high-throughput microscopy in drug discovery. *Br J Pharmacol* 152(1), 62-71 (September 2007).
19. Taverna Project, <http://taverna.sourceforge.net>
20. Taylor, I. Shields, M. and Wang. I.: Resource Management of Triana P2P Services. *Grid Resource Management*, Kluwer, Netherlands (June 2003).
21. Taylor, D., Haskins, J., and Giuliano, K.: High Content Screening A Powerful Approach to Systems Cell Biology and Drug Discovery. 10.1385/1-59745-217-3:33.
22. Wollman, R., Stuurman, N.: High throughput microscopy: from raw images to discoveries. *Journal of Cell Science* 120:3715-3722 (2007).
23. Zhao Y., Hategan, M., Clifford, B., Foster, I., vonLaszewski, G., Raicu, I., Stef-Praun, T. and Wilde, M.: Swift: Fast, Reliable, Loosely Coupled Parallel Computation. *IEEE International Workshop on Scientific Workflows* (2007).