Automatic Workflow Management for Spatial Heritage Data

Michiel Johan Baird
Department of Computer Science
University of Cape Town
Private Bag X3
Rondebosch 7701
Cape Town, South Africa
mbaird@cs.uct.ac.za

Lighton Phiri
Department of Computer Science
University of Cape Town
Private Bag X3
Rondebosch 7701
Cape Town, South Africa
lphiri@cs.uct.ac.za

Hussein Suleman
Department of Computer Science
University of Cape Town
Private Bag X3
Rondebosch 7701
Cape Town, South Africa
hussein@cs.uct.ac.za

ABSTRACT

The Zamani project archives high resolution 3D laser scans and derivative data sets. The process of creating these derivatives is not formally specified, repeatable or machine-executable. As a result, reuse and regeneration of derivative datasets is often poorly understood and difficult to execute. The solution proposed in this paper is a workflow management system to formally and graphically specify the interrelationships among the derivative datasets. The applicability of the resulting system was assessed through a user experience study and a series of applicability tests. It took the user study participants an average of 21 minutes—with an interquartile of seven minutes—to complete the most complex assigned task. Overall, 76% of the participants found the system useful, 71% found it easy and simple to use, 90% found it easy to learn, and 85% were satisfied with the experience with the system. The applicability of this proposed approach was validated though the successful integration of five 3-D modelling workflows, and a user experience study revealed that users would be able to successfully complete workflow tasks within the expected time. In addition, the user experience evaluation indicates that users were satisfied with the system and found it easy to use and learn.

Categories and Subject Descriptors

H.4.1 [**Information Systems Applications**]: Office Automation—*Workflow management*

General Terms

Design, Experimentation, Human Factors

Keywords

Django, Workflow, Zamani

1. INTRODUCTION

The Zamani Project[1], an active initiative in the Geomatics Department at the University of Cape Town, aims to accurately record the physical and architectural nature and dimensions of African cultural heritage sites. The project is part of a broader objective to preserve and protect many of Africa's natural and cultural heritage sites that are under threat (*African Heritage Database*). Heritage sites are mapped using sophisticated technology in order to create a variety of data types that include 3D computer models and panoramic photographs.

The project has thus far managed to record sites in Ghana, Ethiopia, Kenya, Mali, South Africa and Tanzania, and work to map additional sites is currently underway. These records are some of the best, and most accurate, heritage documentation in the world. Nonetheless, the documentation process of these heritage sites requires a significant amount of effort. The process includes the capture, storage, manipulation, analysis and management of the geographic, architectural and photographic data (Rüther et al., 2011). Data is typically manually copied to each point requiring data input, making the overall process extremely slow and laborious. To save time, an obvious solution is the automation of the various data processing phases.

Workflow management systems provide a potential solution to some of these problems due to their ability to decompose complicated procedures into smaller inter-dependent atomic tasks (Taylor et al., 2006; Deelman et al., 2009), subsequently increasing the overall efficiency of the processes. Their effectiveness is evidenced by the increase in productivity when implementation and integration is within business and scientific domains (Brahe and Schmidt, 2007). Most notably, workflow systems have greatly fostered reproducibility in the field of science (De Roure and Goble, 2009). However, the sheer size and diversity of data objects involved poses a challenge to traditional workflow systems.

Thus, driven by the unavailability of appropriate workflow systems for handling large datasets within the Geomatics field, a suitable prototype workflow system was designed and implemented by considering specific user requirements and general workflow system requirements.

While the bulk of this paper is focused on the workflow

[1] http://www.zamani-project.org

system design and implementation, which is highlighted in Section 3, Section 2 is dedicated to prominent work associated with the research conducted; Section 4 outlines the user experience study and system integration tests conducted as part of the evaluation of the system; and, finally, concluding remarks are outlined in Section 5.

2. RELATED WORK

Scientific workflow management has been quite successful by making it easier for the definition of repeatable experiments and, more importantly, facilitating reusability. The repeatability is especially helpful to most scientific experiments as replication of experiments is made much easier (De Roure and Goble, 2009).

There are various products available for composing scientific workflows. Kepler (Altintas et al., 2004) is one such popular workflow system, with pre-built models to facilitate complicated workflows. Aside from it being a Free and Open Source application, a notable feature that makes Kepler stand out is its ability to effectively separate the workflow from its execution, thus allowing seamless batch execution. However, Kepler is not ideally suited for user tasks as its primary focus is complex automation. Additionally, its performance is sub-optimal, as it needs to continuously poll to determine the status of a given task. A critical system requirement requires tasks corresponding to an operational model where user activities are core to the creation of the data. Furthermore, Kepler does not integrate well from an interface point of view. These shortcomings formed the basis for the design and implementation of an alternative system. Suffice it to say, Kepler's core functionalities were replicated. Another popular workflow system used within scientific circles is the Taverna scientific workbench (Oinn et al., 2004), which focuses on workflow sharing. Tarverna can be leveraged to utilise services a client has, in order to facilitate the flow by easily adding services. In addition, the Taverna language is a simple data-flow language—Simple Conceptual Unified Flow Language (SCUFL)–that can be encoded in XML.

A major challenge with existing workflow systems is their inability to process data sets in the Geomatics field with relative ease. Geomatics concerns itself with the organisation, representation and processing of geographic data, for the purpose of querying it and making decisions off the data (Di Martino et al., 2007). The workflow in Geomatics is especially distributed and the data sets operated on are relatively large and diverse. Perhaps the most notable reaction to the lack of appropriate tools for supporting geospatial workflow processes is work by Zhang (Zhang, 2012). Zhang proposes a practical approach to developing geospatial workflow systems by mashing up "open source and commercial packages in innovative ways". More recently, Ai and Xue (Jianwen and Yong, 2013) devised a grid-based processing workflow framework to handle large-scale data sets. Ai and Xue argue for the use of grid services by way of using an integrated unified processing environment.

3. SYSTEM DESIGN

The system design started off with a requirements elicitation phase, with design considerations outlined in Section 3.1 as the main output. The system was then implemented in three design iterations with the feasibility design, workflow component design and the user interface design done in the first, second and third iterations respectively.

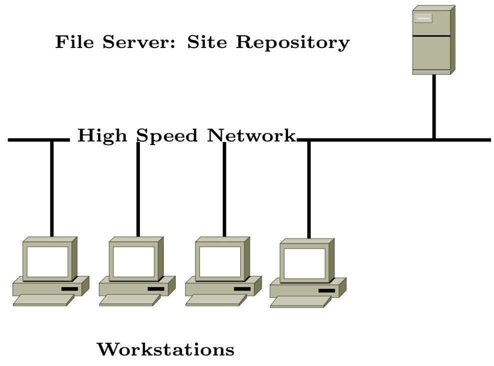

Figure 1: The Zamani Network Configuration

3.1 Design considerations

The system was generally designed to be integrated with the overall Zamani project workflow tasks. With that in mind, the requirements of the Zamani project team members were combined with general requirements of workflow systems (Curcin and Ghanem, 2008) to arrive at a set of core design requirements outlined as follows:

3.1.1 Workstation configuration

Potential end users of the system generally use workstations with varying specifications. The workstations are all connected to a high speed local network, as shown in Figure 1. Task dependent files are typically transferred over the network or via removable drives. The high speed network additionally hosts a server that acts as a repository for the sites.

3.1.2 Task variety

Task processing largely involves creation of derivative data from existing laser scans, photographs and auxiliary source data. The tasks are broadly divided into user- and server-oriented tasks. Server tasks do not require user interaction and include activities such as removal of duplicate points in input data, or file format conversion. User tasks are manual tasks performed by end users, largely involving creation of derivative data elements. Figure 2 shows a typical data flow of a given user task. The majority of tasks are automated, however, there exists some that require manual processing.

3.1.3 Dataflow model

A peculiar characteristic of the data processing steps involved is replication, which occurs when data elements are added to existing derivatives at almost every step. The sheer scale of data processed results in large costs during the transfer of data on the network.

3.1.4 Staff turnover

With a relatively small core team, the project generally relies on interns who regularly cycle in and out of the project. The impact on workflow tasks due to this high staff turnover rate would have to be sufficiently dealt with.

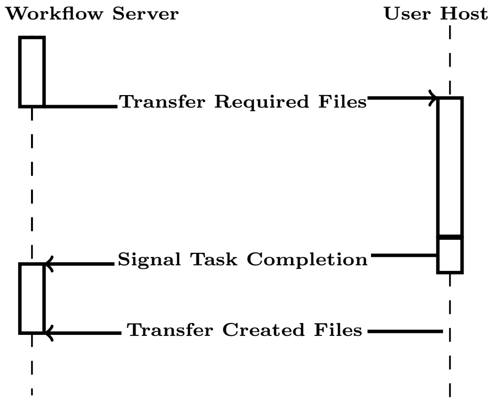

Figure 2: Data Flow in User Tasks for Creating Derivative Elements

3.1.5 *Repeatability*
Processes used to create data items are essentially repeated on all sites. It was thus vital to ensure the design of these processes was done in such a way so as to facilitate the reusability of pre-computed processes of other sites, in effect enabling easy repeatability of experiments.

3.1.6 *System integration*
An assortment of software tools and services are used during the modelling and processing of data. It was essential to ensure the seamless compatibility with existing software.

3.1.7 *System backup*
Processing sites each comprise of unique data items that are irreplaceable or otherwise time consuming to regenerate. In the event of system failures, the system would have to be implemented in such a way that backing up of this data was facilitated.

3.1.8 *Provenance*
The complete history of objects, processes and agents during the processing life cycle is a crucial component of workflow systems (Davidson and Freire, 2008). This ensures appropriate analysis of systems and aids reproducibility. Existing models such as the Open Provenance Model (Moreau et al., 2011) make it possible for workflow systems to be implemented with the required provenance in mind.

3.2 Implementation
A three-tiered application was implemented using the Django[2] Python Web framework. The system supports multiple users, each being able to execute a portion of the workflow assigned to them. System end users are required to log in in order to track and manage their associated tasks. Furthermore, end users are categorised into two groups: privileged and unprivileged users. Unprivileged users only have access to system functionalities to enable them to perform assigned tasks while privileged users are, additionally, able to perform administrative tasks such as site setup and task management.

Working with Yu and Buyya's four-element classification for workflow management systems (Yu and Buyya, 2005), the system was implemented to support the following core features:

3.2.1 *File transfer*
Through the use of user tasks, files can be transferred to user workstations from original input file lists. A trigger on a user task initiates the transfer, with the server connecting to the host and copying over all required files. The transfer is manually triggered by system users, owing to the fact that users can have more than one host. Successful completion of the task at hand initialises a complementary file transfer in the opposite direction, uploading all files in the Output Directory.

3.2.2 *Task automation*
The ability of the system to facilitate workflows is dependent on the automation of tasks. A control system within the server determines tasks that have unmet dependencies; the candidate tasks are subsequently run via an asynchronous task. This is job dependent, however, two outcomes are possible: file transfer or script execution. Once a task is completed, the generated files are committed to the database. On the other hand, if a task fails, user intervention is required to make required changes and restart the process.

3.2.3 *Logging*
In order to identify and control task failures, a logging feature was incorporated within the automation framework. All events linked to a task are added to an append-only log file. The logging includes file transfer logs for user tasks, and stdout and stderr.

3.2.4 *Site visualisation*
A key feature of the system is its ability to represent tasks as a Directed-Acyclic-Graph. To make the system more usable, the tasks are represented as graphs within the Web interface (see Section 3.2.8) by calculating the graph of the sites and applying a layout algorithm. A spring layout was used, resulting in nodes being joined by edges, with a summary of each task displayed on the node.

3.2.5 *Site setup*
Privileged users are able to compose workflows with relative ease through the creation of Jobs and Tasks. Once a task is added, dependencies can be added visually. In addition, existing site structures can be cloned, thus simplifying the creation of new sites.

3.2.6 *Database*
SQLite[3] was used as the system data source, but any Relational Database Management System could easily be used. An authentication module was used to support user data, in order to manage users and permissions on the system.

3.2.7 *Workflow framework*
A key feature of the workflow framework is its ability to execute tasks asynchronously, with each Job type handled separately.

[2] http://www.djangoproject.com

[3] http://www.sqlite.org

One to One Job type. This Job type executes a given script on all input files. The system generates a list of absolute paths and then executes the script using two parameters: an input file and output directory.

Many to One Job type. The execution path of this Job type is similar to One to One Job type, however, the corresponding script is only run once. In addition, all input files are passed as arguments, along with the output directory.

User Job type. A User Job is split into multiple parts. The main object is to ensure effective network communication between workstations and servers during the transfer of large files with minimal redundancy. rsync[4] was used to facilitate file transfer of segments not yet transferred. Figure 3 shows the components used to execute user tasks. Each component is implemented using an asynchronous object and each component can be used at any point during the task's life cycle.

3.2.8 User interface

The use of a Web interface was appropriate, in part, due to the possibility of accommodating multiple system users. Django's View subsystem was used to serve HTML via the following views:

Task overview. This view displays users' outstanding tasks as well as a team overview—providing a project-wide overview. Links are also available to enable easy access to individual tasks. Furthermore, users are able to filter tasks on specific sites.

Task control. This view provides an overview of actions associated with specified tasks. Privileged users additionally have access to a link to enable task editing. Logging information is also visible to provide the complete history associated with the selected tasks.

Task editing. This view enables privileged users to specify task parameters during initial task setup, and also to assign tasks to appropriate system users. Task dependencies, such as input files, are equally specified using this view.

Site view. The purpose of this view is to provide a site-wide visualisation, with each node indicating the current task status. Figure 4 shows a screenshot of the site view.

4. EVALUATION

The successful integration and utilisation of the system into the existing production environment was hinged on two primary goals: its ability to be incorporated into the daily users' workflow with minimal disruption; and its effectiveness with regards to managing data items and task coordination between end users and the system.

[4] http://rsync.samba.org

The overall assessment of the two goals was performed through a user experience evaluation exercise and system integration tests.

4.1 User experience

The user experience evaluation involved assessing the system quality in terms of its usability. To that end, the following usability attributes were evaluated:

- Learnability: the system's ability to enable end users to easily learn how to use it
- Efficiency: the relative timeframes required to complete system-defined tasks
- Satisfaction: end users' attitude towards the system as it pertains to difficulty; confidence performing tasks; and system likes/dislikes
- Error: frequency of logical errors, such as deviations from intended path
- Effectiveness: general user efficiency based on predetermined levels with respect to speed, error frequency and performed steps
- Simplicity: user effort required to complete system tasks

4.1.1 Approach

In order to avoid potential confirmation bias from occurring, system stakeholders were not involved in the user evaluation exercise. 24 participants were thus randomly recruited from the university population—a sample pool comprising participants with no knowledge of modelled system tasks.

Ethical clearance approval was sought prior to conducting the user study. The participants had technical competencies ranging from novice to expert. The participants were given small rewards for participation in the study.

The participants were required to complete two tasks that corresponded to two main system user activities: management of individual tasks; and workflow setup.

Task #1: Complete a simple workflow. A web-based application was created to simulate typical actions of an unprivileged user performing outstanding tasks. A site was created containing three tasks that were assigned to the test user. General usability aspects of the system were being evaluated and, as such, the artefacts used during the evaluation were non-cultural heritage artefacts.

Three Python programs were set up to represent user tasks to be executed. For Task one and Task two, participants were simply required to run the desktop application. Task three, however, was aimed at testing participants' understanding of the system.

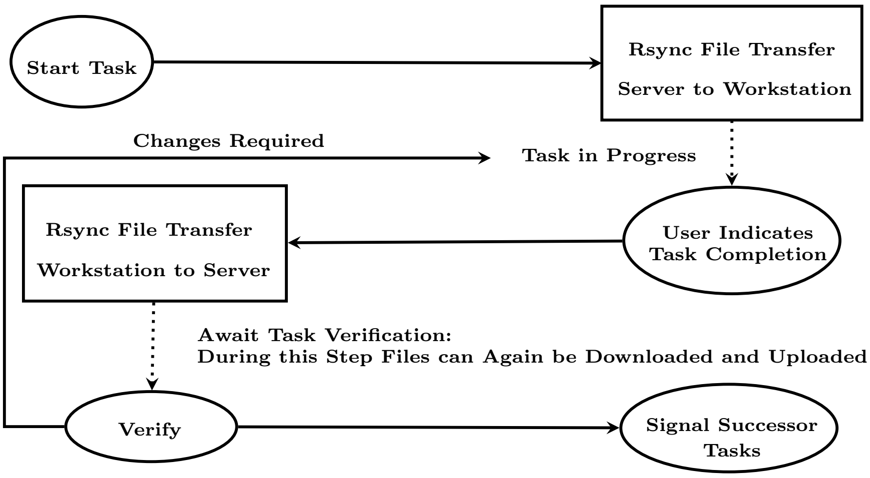

Figure 3: High-level System Components that Support User Jobs

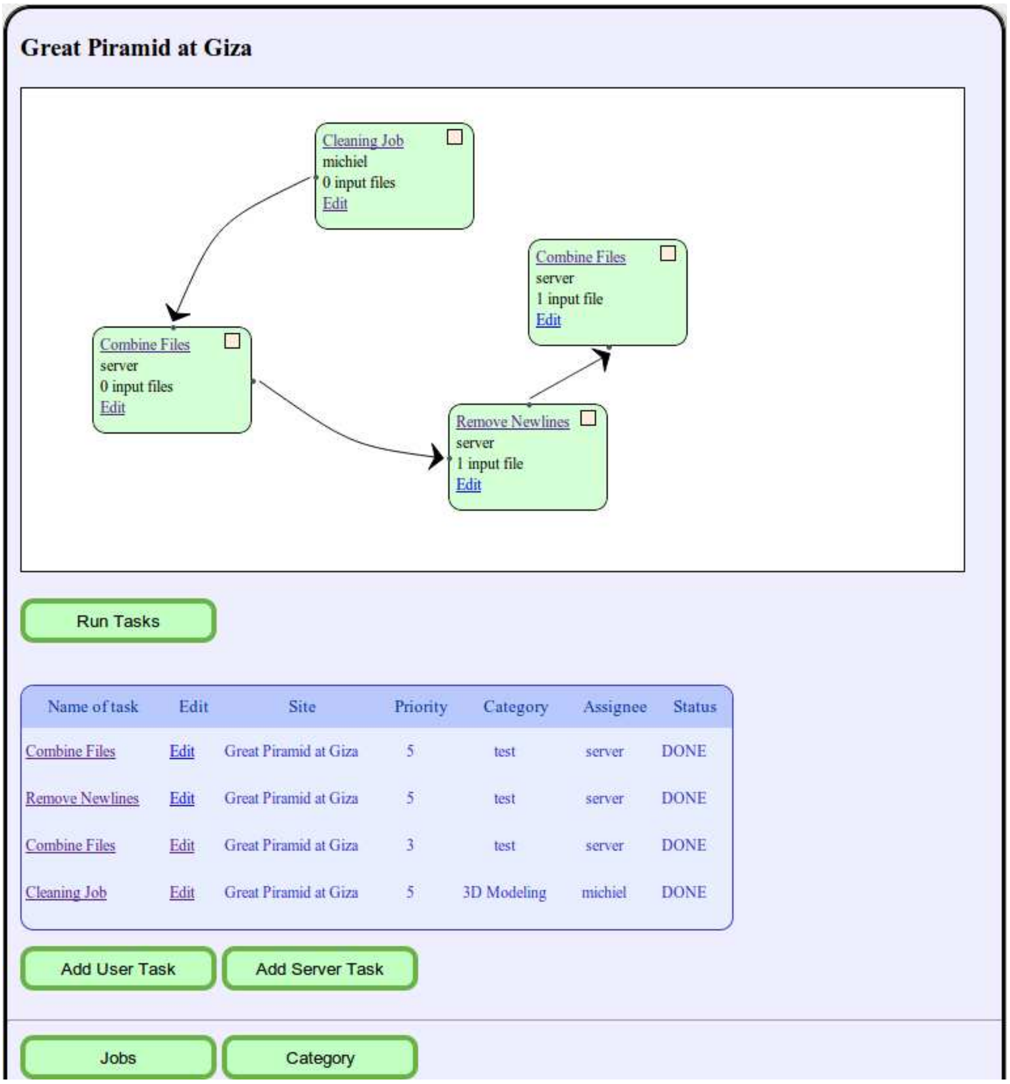

Figure 4: System Site View Implementation

Task #2: Set up a simple workflow. In order to simulate the role of a privileged user, participants were required to set up a sample workflow. The tasks were designed to represent a workflow for generating a PDF file with a text file as initial input.

Figure 5 illustrates the basic workflow steps required to generate the final PDF output. The same illustration was provided to participants to act a guide on the actions required to complete the task. To ensure that each user experienced the system in the same way, it was restored to its previous state after each test.

Post-survey USE questionnaire. A post-survey online questionnaire, prepared using LimeSurvey[5], was used as the primary method for data collection. The USE questionnaire (Lund, 2004) was adopted to measure participants' user experience. The questionnaire was designed to trigger emotional responses in order to elicit usability attributes in terms of: Usefulness; Ease of Use; Ease of Learning; and Satisfaction. Additionally, the questionnaire was designed to duplicate participants' responses in order to avoid Acquiescence Response Bias (Winkler et al., 1982).

4.1.2 Results and discussion

Task #1: Complete a simple workflow. The results indicate that Participants generally found the system useful and felt it would enable them to become more productive. This is evident from the stacked bar plot shown in Figure 7. Significantly, 76% of the participants found it useful. In addition, most users were observed to easily use the system. Notably, there was a slight delay, for roughly 30% of participants, between completion of the first task and beginning of the second task. Owing to the fact that data was being transferred back or tasks awaiting validation, the participants waited for the system to indicate the status change. However, the validation was manual; an interactive task reporting mechanism may thus be desirable.

Participants' responses for the Ease of Use aspect are shown in Figure 8. The number of steps required to complete the task was considered minimal by 80% of participants, and very few inconsistencies were noted. 71% of participants indicated that the system was easy and simple to use. The ability of the system to seamlessly recover from mistakes led to most participants agreeing with the flexibility aspect. Nonetheless, participants initially found the explicit specification of input files and output folder confusing.

The results from the Ease of Learning aspect is shown in Figure 9. 90% of the participants found the system easy to learn, and most of them indicated that they could easily remember steps required to perform the task. Interestingly, 75% considered themselves skillful at performing the assigned task, although they had only used it for a short period of time. This was further confirmed through observations made during the test; most of them started off at a slow pace, but their task action speed gradually increased and became more decisive.

[5] http://www.limesurvey.com

As shown in Figure 10, 80% of participants were satisfied with the experience with the system and were confident of completing the assigned task with relative ease. From the observations made, some users showed clear signs of satisfaction as they became more familiar with the interface.

From the system transactional logs, participants took an average of seven minutes to complete Task #1. The maximum time taken to complete the task was 12 minutes. The interquartile range was determined to be between five and eight minutes, confirming that most participants efficiently accomplished the assigned task.

Task #2: Set up a simple workflow. With 85% and 80% of the participants finding the system effective and productive, respectively, and 94% acknowledging its usefulness, it is safe to assume that most users found the system useful. This is further supported by additional responses shown in Figure 11. From the observations made during the exercise, most of the participants were quick to understand what was expected of them and effectively built the workflow. All users managed to complete the assigned task.

A number of user responses shown in Figure 12 indicate the general perception with regards to the ease of use of the system. Most notably, 85% of the participants indicated that the system was simple and easy to use. 75% were easily able to recover from logical errors, and a similar number commented favourably of the ease of use of the system. While monitoring user interaction, it was observed that users navigated the entire page before deciding on how they would add specific tasks.

Participants were quick to determine general system functionalities, and how to perform actions efficiently. As shown in Figure 13, 80% of their responses point to the fact that they perceived the system to be easy to learn.

The participants' satisfaction is evident from Figure 11, with 90% indicating they were satisfied and 75% finding it fun to use. The emotional responses provided more insight into their experience and results show a significant number of them perceived it to be something they would need to have.

As with the first task, it took the participants an average of seven minutes to complete the assigned task. However, the interquartile range was between five and nine minutes. While most of them immediately became efficient at performing the task, some of them spent a significant amount of time staring at the visualisation since they thought the previous tasks had been removed. This was as a result of the drag-and-drop functionality not being apparent.

4.2 System integration tests

Since it was essential to assess the applicability of managing and coordinating production environment tasks, system integration tests were performed by implementing a subset of production environment tasks. The lengthy process required to build 3D models necessitated the need to only test a subset of the entire workflow. The initial five steps—Scanning the Object, Cleaning the Scans, Preparation for Registration, Meshing and Transformation of Scans—of the Zamani

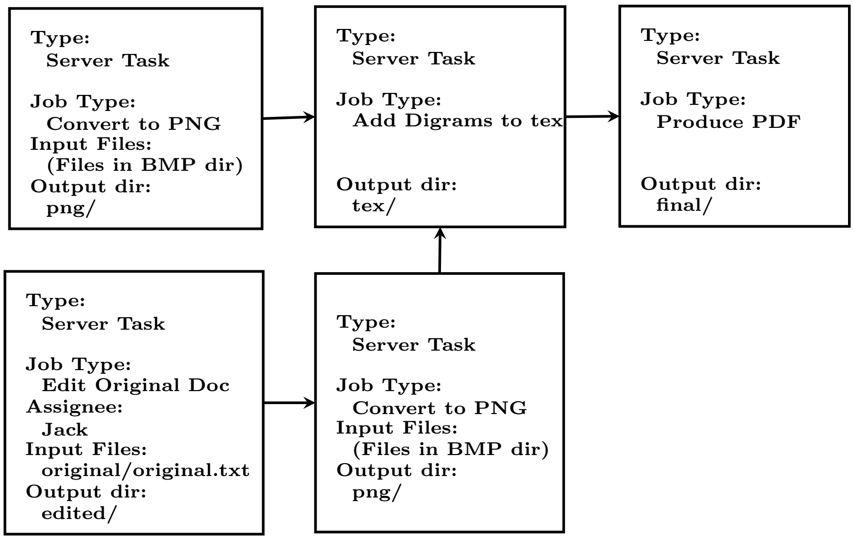

Figure 5: Workflow Tasks Required During Experimentation

3D modelling process (Rüther et al., 2011) were successfully implemented using the workflow system. Figure 6 shows a screenshot of a typical workflow execution process as would be performed by the system.

Movement and processing of data is the most expensive phase of the process and, as such, the system is designed in a way that enables data movement to be directly offloaded to rsync, whereas the data processing is offloaded to native applications. The system management of these activities outperforms these tasks, thus binding system performance to them. It is due to this reason that an explicit system performance evaluation was deemed unnecessary.

The system test revealed that filtering workflow at the site level is not ideal. In addition, the integration tests identified the need to store workflows at finer levels so as to decrease set up time, by preventing workflow replication at building level. A viable alternative is nesting in hierarchies.

5. CONCLUSION

Working with explicit user requirements from end users currently making use of a manual processing workflow for handling large-scale data sets, a workflow management system was designed and implemented. Partial integration of production environment workflow steps proved the feasibility and effectiveness of the proposed approach. A strong focus on end users necessitated the need for a user experience study. The results from the user experience study indicate that potential system end users would find it useful and easy to use.

The development and evaluation of this prototype workflow system for geographical information systems shows that it is indeed feasible to automate such workflows, in a manner that affords users a satisfactory user experience. As research data becomes a prominent topic in digital archiving, the specific issues of large data sizes and derivative datasets that are characteristic of GIS data need to be addressed. The prototype in this work suggests that specialized workflow and automation systems can successfully improve on the user experience.

During the course of this work, various possible extensions and improvements to the system were identified. The additional features would especially improve the system in terms of performance, usability and set up times. A crucial aspect of the system is its ability to scale and handle significantly larger and complex workflows. The system would thus have to be distributed in order to reduce the bottleneck in processing Server Tasks. Creation of derivative data items is central to the system, however, it is often the case that some files within a site change without creating an additional copy. An additional filtering technique needs to be utilised, preferably through the use of rule-based filtering, to place greater control on output files. Finally, abstracting workflows through hierarchies would allow for better control and reusability of tasks.

6. ACKNOWLEDGEMENTS

Special thanks go to the UCT Geomatics Department for allowing us to use their laser scans and related data for this research.

7. BIBLIOGRAPHY

[1] Ilkay Altintas, Chad Berkley, Efrat Jaeger, Matthew Jones, Bertram Ludascher, and Steve Mock. "Kepler: an extensible system for design and execution of scientific workflows". In: *Proceedings of the 16th International Conference on Scientific and Statistical Database Management, 2004*. IEEE, 2004, pp. 423–424. DOI: 10.1109/SSDM.2004.1311241.

[2] Steen Brahe and Kjeld Schmidt. "The Story of a Working Workflow Management System". In: *Proceedings of the 2007 International ACM Conference on Supporting Group Work*. New York, NY, USA: ACM, 2007, pp. 249–258. DOI: 10.1145/1316624.1316661.

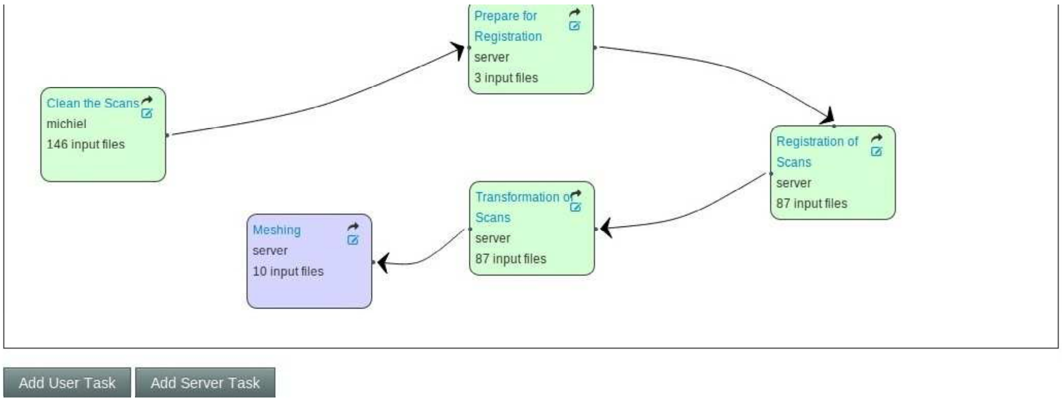

Figure 6: Partial Implementation of the Zamani Modelling Workflow

[3] Vasa Curcin and Moustafa Ghanem. "Scientific workflow systems - can one size fit all?" In: *2008 Cairo International Biomedical Engineering Conference*. IEEE, Dec. 2008, pp. 1–9. DOI: 10.1109/CIBEC.2008.4786077.

[4] Susan B. Davidson and Juliana Freire. "Provenance and scientific workflows". In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08*. New York, New York, USA: ACM Press, 2008, p. 1345. DOI: 10.1145/1376616.1376772.

[5] David De Roure and Carole Goble. "Software Design for Empowering Scientists". In: *IEEE Software* 26.1 (Jan. 2009), pp. 88–95. DOI: 10.1109/MS.2009.22.

[6] Ewa Deelman, Dennis Gannon, Matthew Shields, and Ian Taylor. "Workflows and e-Science: An overview of workflow system features and capabilities". In: *Future Generation Computer Systems* 25.5 (May 2009), pp. 528–540. DOI: 10.1016/j.future.2008.06.012.

[7] Sergio Di Martino, Filomena Ferrucci, Luca Paolino, Monica Sebillo, Genny Tortora, Giuseppe Vitiello, and Guiseppe Avagliano. "Towards the automatic generation of web GIS". In: *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems - GIS '07*. New York, New York, USA: ACM Press, 2007, p. 1. DOI: 10.1145/1341012.1341081.

[8] Ai Jianwen and Xue Yong. "A Processing Framework of Grid Workflow for Remote Sensing Quantitative Retrieval". In: *Proceedings of the 2013 International Conference on Remote Sensing,Environment and Transportation Engineering*. Paris, France: Atlantis Press, 2013. DOI: 10.2991/rsete.2013.227.

[9] Arnold M. Lund. "Measuring Usability with the USE Questionnaire". In: *Usability& User Experience* 8.2 (Oct. 2004). Retrieved May 1, 2014, from STC Usability Website. URL: http://www.stcsig.org/usability/newsletter/0110_measuring_with_use.html.

[10] Luc Moreau et al. "The Open Provenance Model core specification (v1.1)". In: *Future Generation Computer Systems* 27.6 (June 2011), pp. 743–756. DOI: 10.1016/j.future.2010.07.005.

[11] Tom Oinn et al. "Taverna: a tool for the composition and enactment of bioinformatics workflows." In: *Bioinformatics (Oxford, England)* 20.17 (Nov. 2004), pp. 3045–54. DOI: 10.1093/bioinformatics/bth361.

[12] Heinz Rüther, Christoph Held, Roshan Bhurtha, Ralph Schröder, and Stephen Wessels. "Challenges in heritage documentation with terrestrial laser scanning". In: *Proceedings of the 1st AfircaGEO Conference*. 2011.

[13] Ian J. Taylor, Ewa Deelman, Dennis B. Gannon, and Matthew Shields. *Workflows for e-Science: Scientific Workflows for Grids*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[14] United Nations Educational Scientific and Cultural Organisation. *African Heritage Database*. Retrieved August 1, 2014, from UNESCO Website. URL: http://www.unesco.org/new/en/natural-sciences/science-technology/space-activities/space-for-heritage/activities/open-initiative-projects/african-heritage-database.

[15] John D. Winkler, David E. Kanouse, and John E. Ware. "Controlling for Acquiescence Response Set in scale development." In: *Journal of Applied Psychology* 67.5 (1982), pp. 555–561. DOI: 10.1037/0021-9010.67.5.555. URL: http://content.apa.org/journals/apl/67/5/555.

[16] Jia Yu and Rajkumar Buyya. "A Taxonomy of Scientific Workflow Systems for Grid Computing". In: *SIGMOD Rec.* 34.3 (Sept. 2005), pp. 44–49. DOI: 10.1145/1084805.1084814.

[17] Jianting Zhang. "A practical approach to developing a web-based geospatial workflow composition and execution system". In: *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications - COM.Geo '12*. New York, New York, USA: ACM Press, 2012, p. 1. DOI: 10.1145/2345316.2345341.

APPENDIX
A. USE QUESTIONNAIRE RESPONSES

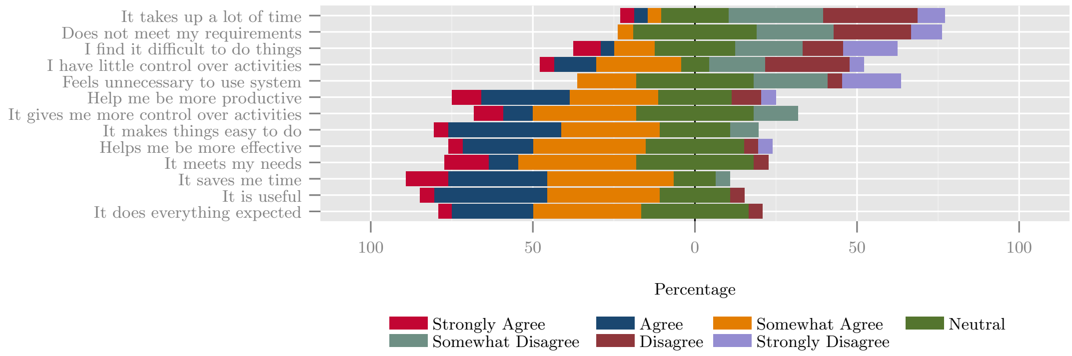

Figure 7: Task #1: Rate the system in terms of Usefulness in the following categories

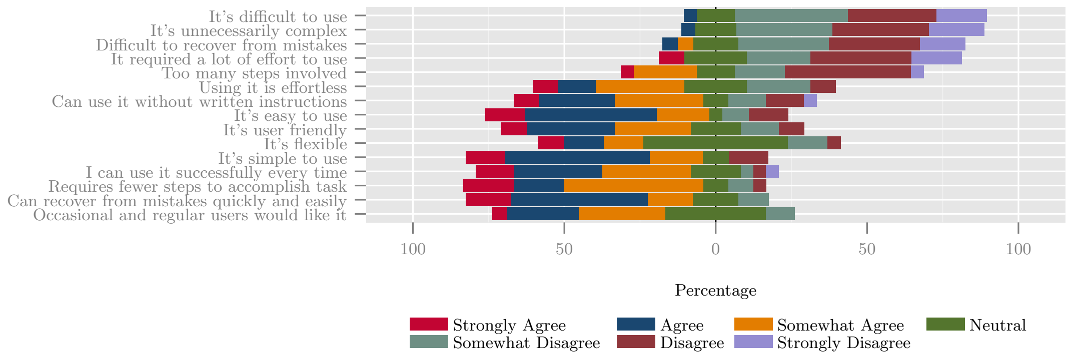

Figure 8: Task #1: Rate the system in terms of Ease of Use in the following categories

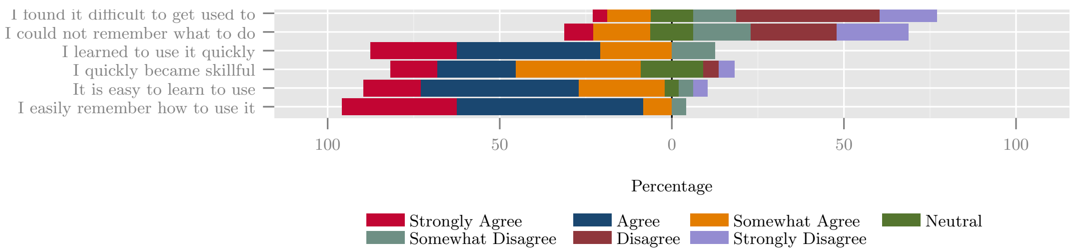

Figure 9: Task #1: Rate the system in terms of Ease of Learning in the following categories

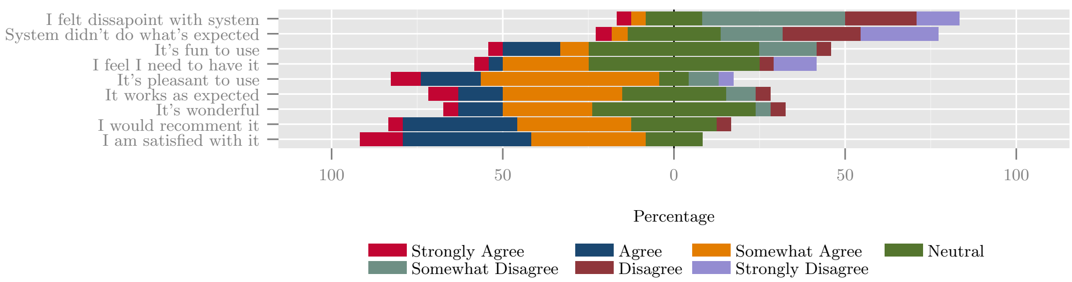

Figure 10: Task #1: Rate the system in terms of Satisfaction in the following categories

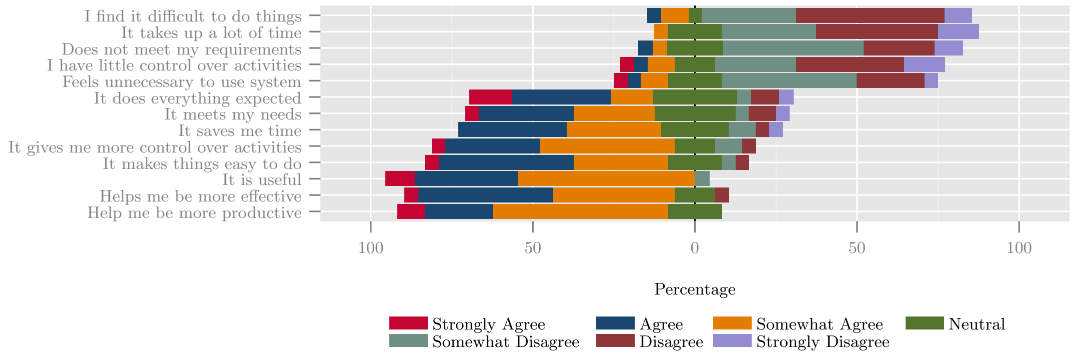

Figure 11: Task #2: Rate the system in terms of Usefulness in the following categories

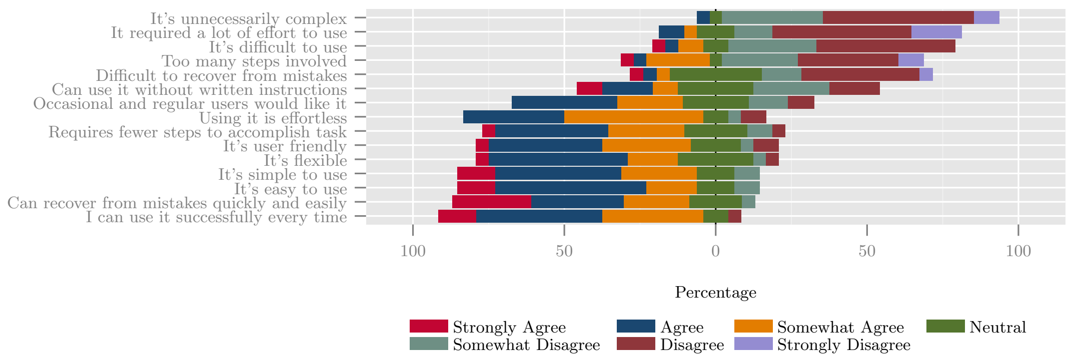

Figure 12: Task #2: Rate the system in terms of Ease of Use in the following categories

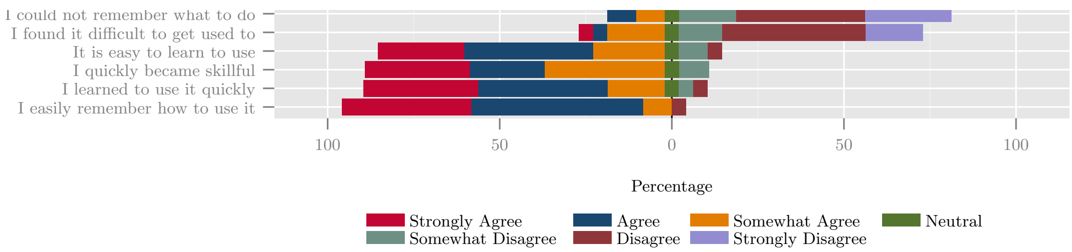

Figure 13: Task #2: Rate the system in terms of Ease of Learning in the following categories

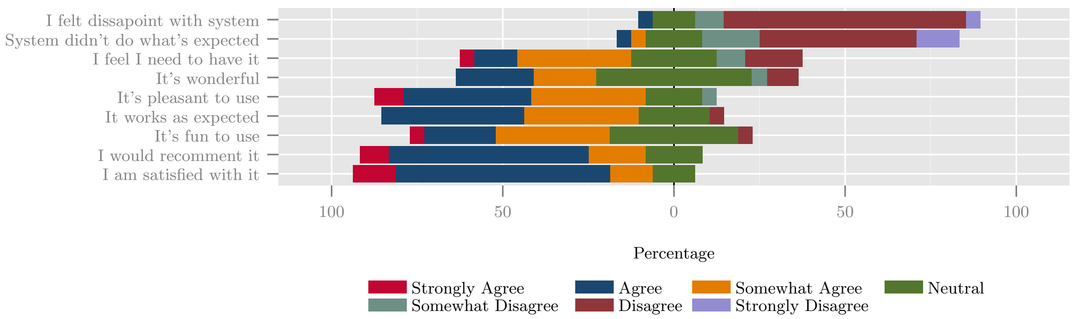

Figure 14: Task #2: Rate the system in terms of Satisfaction in the following categories