

# **Honours projects 2014**

# Projects

1. Project: How to Balance Two Poles? .....	3
2. Student Advisor .....	4
3. African Language Web.....	5
4. Agile Project Management Office .....	6
5. Cloudlets: making new co-located mobile media experiences.....	7
6. Project: Cooperation versus Competition .....	9
7. Testing and deployment framework for services in the ROC.....	11
8. Electronic Marking System .....	12
9. e-Sports Portal .....	13
10. Archives to Exhibitions .....	14
11. Adaptive Field D* path planning for mesh environments .....	15
12. GLYCANO: Building and visualizing sugars. ....	17
13. Project: Lifetimes of Migration.....	19
14. Vector Variations .....	20
15. A mobile visual design application for entry-level smartphones.....	22
16. Information system for protected areas in South Africa .....	24
17. Enriching UML class diagrams with the taxonomy of part-whole relations .....	25
18. Visualization of Radio Frequency Interference data for MeerKAT/SKA.....	27
19. Fast and accurate triangle mesh simplification for large heritage models.....	29
20. Tick-borne Diseases Database .....	31
21. Data Structures Tutor .....	32
22. Pseudo-natural language rendering (verbalization) of OWL 2 ontologies.....	33
23. Virtual Panning for Lecture Recording .....	35
24. Zamani Data Archive.....	36
25. Map based visual optimization of Hotel Booking Systems .....	37

# 1. Project: How to Balance Two Poles?

## Proposer:

Geoff Nitschke

## Abbreviation:

2Poles

## Brief Description:

In the double pole balancing task, two poles (inverted pendulums) are hinged to a wheeled cart on a finite stretch of track. The task is for both poles to be balanced simultaneously by applying a continuous force to the cart. The system becomes more difficult to control with poles of similar lengths and if the velocities are hidden from the controller. The pole-balancing task is a popular machine learning test-bed since it requires solving the temporal credit assignment problem, and is a good surrogate for a more general class of unstable control problems such as bipedal robot walking and rocket guidance. The pole balancing task has been solved using Dynamic Programming methods such as Reinforcement Learning, but recently methods for evolving artificial neural networks (Neuro-Evolution) have yielded significantly higher task performances.

The objective is to implement neuro-evolution methods for double pole balancing controller adaptation, testing new task parameter settings, to try and achieve higher task performance results than those reported as state-of-the-art in the literature. In this case, optimal task performance is minimising the number of evaluations before poles are balanced.

## Computer Science Content:

Optimisation problems, artificial neural network controller adaptation, neuro-evolution.

## Specific Learning Outcomes:

Experiment design and execution, system development and implementation, theoretical and statistical analysis of results, drawing of conclusions based on analyses as well as observations of empirical data.

## Skills Required by Team as a Whole:

Some knowledge of the pole balancing optimisation problem, and neuro-evolution is desirable. It is recommended that each team member implement a separate method to solve the double pole balancing task.

- **Theory:** Optimisation problems, artificial neural network controller adaptation, applying neuro-evolution to optimise controllers.
- **Implementation:** New methods should be implemented or current methods should be re-implemented, run and compared with given code for other optimisation methods. Code is already available for re-running other methods (necessary for experimental comparison).

## Facilities needed:

Honours lab computers.

## Supervision:

Weekly meetings, updates, and occasional demos.

## Number of Students:

2 or 3.

## 2. Student Advisor

**Proposer:**

Audrey Mbogho

**Abbreviation:**

Advisor

**Brief Description:**

Sometimes a student wants to know during the December holidays what courses to take in the coming year. Other times, student advisors are busy with other students, lecturing, or away at conferences. This project aims to develop a Web-based student advisor system which a student can access anytime anywhere. The system will combine inputs provided by the student via a user-friendly interface with student information residing in UCT's various databases in order to give comprehensive and intelligent advice. For example, if the system can access information about the student's health from a system at the Wellness Centre, this may influence the advice it gives. The student may not be willing to divulge this information if speaking to a human advisor, which may lead the human advisor to give less than optimal advice. Furthermore, supplying the human advisor with large amounts of information is likely to confuse rather than facilitate his or her decision making process.

**Computer Science Content:**

Expert systems, decision support

**Specific Learning Outcomes:**

Software development and evaluation; user interface design and usability testing

**Skills Required by Team as a Whole:**

Programming and user interface development

- **Theory:** *Minimal*
- **Implementation:** Moderate; learning a new programming paradigm will be required

**Facilities needed:**

Lab computers will suffice

**Supervision:**

Weekly meetings with supervisor

**Number of Students:**

2

### 3. African Language Web

**Proposer:**

Hussein

**Abbreviation:**

*AFRIWEB*

**Brief Description:**

isiXhosa has more than 8 million native speakers (Statistics South Africa, 2011) but little ability to search through document collections, whether in isiXhosa or other languages, using queries that contain isiXhosa words. Digital documents in isiXhosa are still rare; as of March 2014, Wikipedia only contained 172 articles in isiXhosa. However, the number of such documents has the potential to increase as more speakers of the language become digitally literate, more government documents are produced, more documents on the Web are translated and more books are produced for teaching and learning and popular consumption. There are thus clear benefits for basic search systems in isiXhosa.

This project is further motivated by the objectives of the South African National Development Plan (NDP – National Planning Commission, 2012), where the aim is to create a society with fewer barriers among different groups of people and especially bring marginalized communities into the mainstream. The NDP, in particular, mentions: the need to address language and inclusivity issues in planning basic education; and the need for greater multilingualism and the support of local languages.

The goal of this Honours project is to build a simple information retrieval (IR) system for local African language (including isiXhosa) documents. The core of the project will be the development of a focused crawler to gather, and subsequently index, content in these languages from the Web. A Web interface will then allow users to search through this specialized subset of content.

**Computer Science Content:**

Digital Libraries, information retrieval, ICT4D

**Specific Learning Outcomes:**

Research Methodology; Experiment design and execution; Real-world software development

**Skills Required by Team as a Whole:**

- **Theory:** Nothing specific.
- **Implementation:** Knowledge of HTTP/WWW, XML, HTML, JS and Python – but some can learnt “on the job”.
- **Other:** Excitement about local languages and development!

**Facilities needed:**

Web server

**Supervision:**

Hussein, regular contact with Digital Libraries research group and ICT4D centre for feedback and assistance, collaboration with UCT Department of African Languages

**Number of Students:**

2-3

## 4. Agile Project Management Office

**Proposer:**

Sonia Berman

**Abbreviation:**

AgilePMO

**Brief Description:**

Derivco finds existing issue tracking and project management systems, by trying to cater for everyone, end up either being over-featured (so important stuff gets lost or too complicated) or forcing workflow onto a team that could counter the way they want to work. Most focus on the development team (who really only need issue tracking) and neglect the needs of the Scrum Master and Product Owner. Derivco is interested in a custom Agile PMO system. It should work in a drag and drop fashion as much as possible, i.e. replicating the manual process. The team should use their system as they develop it.

The Agile PMO should capture stories/issues and estimations, and track velocity and burn down. Scrum Master tools are needed to group issues into sprints and sprints into releases and schedule them, and to re-order stories in the backlog. Drag-and-drop scrum boards and a workspace for developers to manage their stories are required, as well as reports on sprints, on the entire project, etc. If there are 3 group members, a mobile interface to log stories and view backlogs could be tackled; if 2 students then the system would be purely Web-based.

**Computer Science Content:**

Databases, Software Engineering, HCI

**Specific Learning Outcomes:**

As above.

**Skills Required by Team as a Whole:**

- **Theory:** This is not a theoretical project.
- **Implementation:** This is a software engineering rather than a research project. Emphasis is on the development process and on good design, analysis and implementation skills.

**Facilities needed:**

No special equipment required.

**Supervision:**

Weekly meetings with supervisor. The project will involve collaboration with Derivco developers based in Durban, but this will be mainly done remotely.

**Number of Students:**

2 or 3

## 5. Cloudlets: making new co-located mobile media experiences

**Proposer:**

Edwin Blake

**Abbreviation:**

cloudlet

**Brief Description:**

The cloud is a global phenomenon. We make use of cloud services because they provide us with unique ways to collaborate, stay in touch, and engage with media socially. Such services have done wonders to connect, and make sharing easier for, people across the globe. The cloud computing paradigm has enabled a company such as WhatsApp to scale to 18M users and handle spikes in demand.

Such is the reach of the cloud computing paradigm that it also affects the design and implementation of wireless access points, whose purpose is no longer to interconnect devices but to connect them to the internet and the cloud. For instance, it is not possible to create a wireless access point to share files locally on current mobile platforms without also sharing your 3G internet connection. But at the same time we, as Computer Scientists, know that using cloud services is an outright inefficient way of sharing files with the person next to you.

We also know the same technologies that power WhatsApp, namely web servers, data stores, and databases, can also run at a smaller scale on embedded systems, such as the Raspberry Pi. So even a Raspberry Pi extended with a battery, WiFi radio and local storage can provide a group of co-located friends with opportunities to share and engage with media, not with the public at large, but amongst themselves. Thus, transforming the cloud from a global panopticon to a hyper-localized, ad-hoc instantiations of the cloud: a cloudlet in short.

We call on hackers, makers, and activists alike to explore how to provide cloud-like services in localized context among a group of friends and using embedded hardware. This project will be split between team members and will include building/designing/making:

- Extending an embedded system, such as a Raspberry Pi, Beagle Board, or Intel Galileo SoC, to function as a wireless access point.
- Implement a co-located, group-based data store and exchange protocol.
- Design a co-located, mobile media sharing app (Android) that interfaces with the embedded system.
- Consider group interactions with shared resources and devices, such as a loudspeaker (music) and TV (photos and video).

This is part of a larger HCI project in the ICT4D lab, exceptional work may be taken further or considered for inclusion in publication.

**Computer Science Content:**

This project implements the full stack: ranging from group-based file ontologies, data stores and exchange protocols all the way to the social and cultural implications of sharing media in group contexts. As such there will be both technical and HCI/interaction design related challenges.

**Specific Learning Outcomes:**

ad-hoc networks, databases, mobile programming, and user experiment design

**Skills Required by Team as a Whole:**

This project requires a diverse team with networking, mobile programming and HCI skills

- **Theory:** Android Programming, familiarity with Linux, databases, networking, command-line use, HCI practices.
- **Implementation:** This project will include embedded and mobile programming, as well as user-interface design. Assistance will be available.

**Facilities needed:**

Raspberry Pi (or other embedded systems), Android mobile phones, access to users.

**Supervision:**

You will join in activities of the interdisciplinary ICT4D centre. You will have access to mobile programming and HCI expertise and will be co-supervised by Thomas Reitmaier and Pierre Benz (PhD students). Please see <http://www.cs.uct.ac.za/~edwin/honsProj.html> for more details.

**Number of Students:**

2 or 3 (4 possible depending on the scope tackled)



## 6. Project: Cooperation versus Competition

### Proposer:

Geoff Nitschke

### Abbreviation:

CoopComp

### Brief Description:

The project will study of the inter-play between cooperative and competitive behaviour in an Agent Based Simulation. The project will investigate the (simulated) evolutionary and environmental conditions under which it is best for agents to cooperate versus with each other. Evolutionary Algorithms (EA) will be used to adapt the behaviour of potentially thousands of agents “living” in a virtual world. Various EA selection parameters will be tested to gauge the impact on evolving cooperative versus competitive behaviour. To make the study relevant to real world tasks, the agent based system will simulate a collective construction task. The task is an abstraction of automated multi-robot construction of inhabitable structures, requiring agents to build “working” structures (built following construction rules) as efficiently as possible (minimising time spent). In collective construction, agents may compete, trying to connect as many building blocks as possible in building a structure, thus increasing their own reward (fitness). Alternatively, agents may cooperate, working with other agents to move larger blocks that could be individually moved, but at a much slower rate. In this case fitness must be shared between all cooperating, meaning less of an individual reward.

A key objective is to use the agent based simulation to find correct ratios of cooperative versus competitive behaviours for given task configurations (e.g.: number of agents, complexity of structure to be built, initial locations of building blocks, difficulty of moving and connecting building blocks).

### Computer Science Content:

*Agent Based Systems* – use adaptive methods (e.g. evolutionary and learning methods) to investigate adaptive behaviour and social phenomena, such as cooperation and competition, in computer simulations. Such experiments try to simulate the conditions that give rise to adaptive and social behaviour.

### Specific Learning Outcomes:

Experiment design and execution, system development and implementation, theoretical and statistical analysis of results, drawing of conclusions based on analyses as well as observations of empirical data.

### Skills Required by Team as a Whole:

Knowledge of, and experience with EAs is needed. Some knowledge of agent based systems is desirable. The team can be split into the following roles: 1) Methods for evolving cooperation EA. 2) Methods for evolving competition. If there is a third team member; 3) Investigating impact of task complexity on evolving cooperative versus competitive behaviour.

- **Theory:** Evolutionary Algorithms, Artificial Neural Networks, Agent based simulations.
- **Implementation:** Agent based system – can be written in a variety of languages.

### Facilities needed:

Honours lab computers.

**Supervision:**

Weekly meetings, updates, and occasional demos.

**Number of Students:**

2 or 3.

## 7. Testing and deployment framework for services in the ROC

### Proposer:

Michelle Kuttel

### Abbreviation:

DvOp4Grd

### Brief Description:

The AfricaArabia Regional Operations Centre (AAROC) is the point of interoperability between our regional infrastructure, the European Grid Infrastructure (EGI.eu) and its peer infrastructures in the US, Latin America, China and India. The ROC is coordinated by the regional NREN Alliances (UbuntuNet and ASREN), and consists of several sites running diverse hardware, OS and middleware configurations. As there is widely varying site administration capability at different sites, the ROC has proposed a more centralised configuration and orchestration model, in order to ensure efficient operations at continental scale, while promoting sustainable inclusivity.

This project will develop a robust testing and deployment framework for services in the ROC.

### Computer Science Content:

Distributed systems, grid computing, virtualisation, cloud computing, HPC, data infrastructures, unit testing.

### Specific Learning Outcomes:

Orchestration and Automated deployment of distributed systems. Best practice of HPC and data infrastructures operations. Several middleware stacks and their components, OGF standards.

### Skills Required by Team as a Whole:

There are distinct roles of “developer” and “integration testing” here. Code developed for deploying new services, versions of services or sites (as collections of services) should be thoroughly tested in a development environment.

- **Theory:** *modelling of distributed systems, revision control, continuous integration, PKI*
- **Implementation:** *you will need to be proficient with revision control tools (especially git). Service configuration and orchestration should be done with Ansible, puppet and related tools.*

### Facilities needed:

All facilities will be provided by SAGrid member institutes and EGI.eu core services.

### Supervision:

Bruce Becker (SAGrid) will co-supervise. Work will be supported by the CHAIN-REDS FP7 project, GRNET and EGI Operations. Work may be presented at the EGI Community Forum, as well as the annual CHPC conference.

### Number of Students:

2 (or 3)

## 8. Electronic Marking System

**Proposer:**

Gary Stewart

**Abbreviation:**

EMS

**Brief Description:**

*Marking paper based theory tests, although necessary, is a time consuming process fraught with administrative challenges,*

- very occasionally students don't receive or lose their marked scripts
- scripts are returned to students, leaving no persistent record
- detailed analysis and research is not possible

*The envisioned system will have a simple workflow with a component which allows completed test scripts to be scanned/captured, tablet and web based components which allows marking to be done and component(s) which allows marks to be captured and marked scripts to be returned/emailed to students.*

**Computer Science Content:**

Image Processing, Human Computer Interaction (HCI)

**Specific Learning Outcomes:**

Mobile development, web development, meeting particular end-users needs.

**Skills Required by Team as a Whole:**

Web Development (HTML, CSS, Javascript, Server Side Scripting (PHP, JSP or Python)), and Mobile Development (Java for Android)

- **Theory:** Web Development, Mobile Interaction Design, Image Processing
- **Implementation:** *The challenge for the whole system is developing an automated, useable interface which simplifies the paper-based test marking process. The web-based component will involve mastering complex web technologies. The tablet-based component will involve mastering the usability complexities of mobile interaction design for relatively inexperienced users.*

**Facilities needed:**

PCs and Mobile devices

**Supervision:**

Regular meetings with supervisor, approximately every two weeks.

**Number of Students:**

2

## 9. e-Sports Portal

### Proposer:

Sonia Berman / Hussein

### Abbreviation:

eSports

### Brief Description:

Derivco is a global company with offices scattered around the world. Each office has game consoles, and players want to challenge and play games against each other remotely. Derivco is setting up E Sports leagues to facilitate this and needs a Web-based system to run this process. The system should manage players, teams, leagues, results, game fixture calendars, etc. Tools should allow administrators to create leagues (based on games, e.g. FIFA 2014, DOTA, Battle Field etc, and with progression information - round robin, knockout ladder or both), to generate fixtures based on availability and taking time zone differences into account, and to manage result verification (e.g. uploading screenshots and mutual confirmation) and result disputes. If there are 3 group members, a mobile interface should be tackled; if 2 students then the system would be purely Web-based.

### Computer Science Content:

Databases, Software Engineering, HCI

### Specific Learning Outcomes:

Orchestration and Automated deployment of distributed systems. Best practice of HPC and data infrastructures operations. Several middleware stacks and their components, OGF standards.

### Skills Required by Team as a Whole:

- **Theory:** This is not a theoretical project.
- **Implementation:** This is a software engineering rather than a research project. Emphasis is on the development process and on good design, analysis and implementation skills.

### Facilities needed:

No special equipment required.

### Supervision:

Weekly meetings with supervisor (Sonia Berman **OR** Hussein). The project will involve collaboration with Derivco developers based in Durban, but this will be mainly done remotely.

### Number of Students:

2 or 3

## 10. Archives to Exhibitions

**Proposer:**

Hussein

**Abbreviation:**

EXHIBIT

**Brief Description:**

Have you seen the much-talked-about indigenous rock art first-hand? Have you read the original stories of the first people of South Africa?

Archaeological data and heritage collections encode and preserve key elements of South African heritage. While such collections are an invaluable tool to researchers, they are not generally accessible to the public or anyone outside the tertiary education sector. This project is on the presentation of multiple forms of archaeological/heritage data for the express purpose of pre-university learning in various places. In particular, these preservation-oriented archives should be converted to visual representations and user experiences suitable for people who visit museums, people who surf the Web and people who use mobile devices for information.

The key goal of the project is to create modern interfaces that encourage people to learn about heritage. In all instances (Web, museums, mobile devices), information must be selected and customized for particular end-user experiences.

The Nelson Mandela Centre of Memory (<http://archive.nelsonmandela.org>) is a good example of a Web experience based on selected archival documents.

**Computer Science Content:**

Digital Libraries, heritage preservation, exhibition

**Specific Learning Outcomes:**

Research Methodology; Experiment design and execution; Real-world software development

**Skills Required by Team as a Whole:**

- **Theory:** Nothing specific.
- **Implementation:** Knowledge of XML, HTML, JS and a scripting language – but some can learnt “on the job”. Mobile device programming of some sort.
- **Other:** Excitement about the preservation of Cultural Heritage!

**Facilities needed:**

Web server, Data and Metadata – all will be provided

**Supervision:**

Hussein, regular contact with Digital Libraries research group and ICT4D centre for feedback and assistance, client in UCT Departments of Archaeology and Fine Arts

**Number of Students:**

2-3

# 11. Adaptive Field D\* path planning for mesh environments

Proposer:

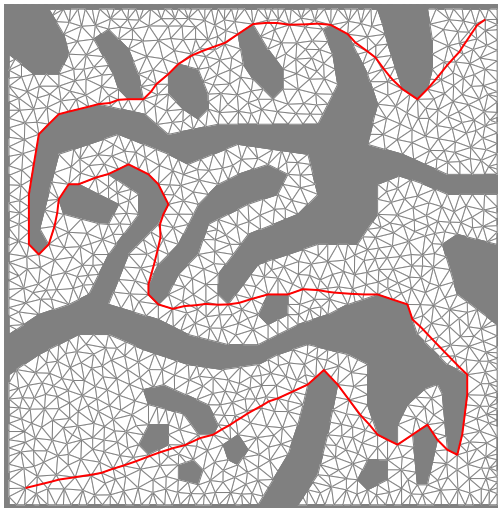
Patrick Marais

Abbreviation:

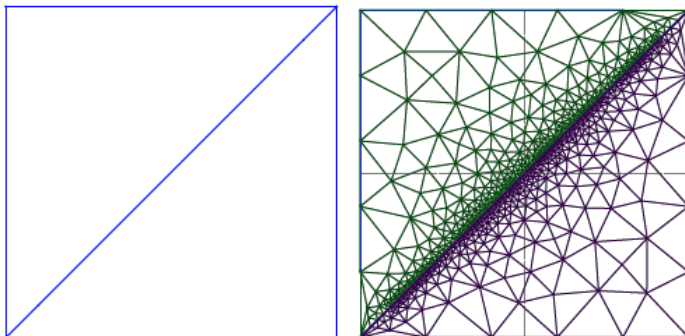
FDSTAR

Brief Description:

Path planning is a vital component for autonomous agents that need to find their way through complex environments. Applications include the navigation system of the Mars Rover, moving across the cratered surface of Mars, or AI agents in a game as they try to path towards enemy units. For games and simulations these environments are often represented as height-fields represented as 3D meshes. One of the newer techniques used to find optimal paths in such a mesh environment is the *Field D\** algorithm, a generalization of the better known A\* pathing algorithm.



This project aims to reduce the time and space requirements of the Field D\* pathfinding algorithm through the use of adaptive meshing, estimation via A\* and remeshing. Field D\* finds paths on weighted triangle meshes, compared to traditional algorithms such as A\* which operate on the weighted edges of a graph. Central to the algorithm is a cost function, which minimises the cost of travelling over a weighted triangle.



The cost function generally has a minimum which can be calculated analytically. However, calculating this minimum, as well as the cost of travelling through a triangle is far more computationally expensive than evaluating travel costs on graph edges with fixed values.

This project will investigate using the above-mentioned techniques in a bid to significantly reduce the number of cost functions evaluated by the algorithm. Briefly, this will be attempted through:

- Only performing computation on parts of the environment in which the path is actually located.
- Testing whether it is possible reduce the main computational expense burden of Field D\* via approximation with the A\* algorithm.
- Finding ways to reduce the size of the environment via Progressive Meshing.

#### Computer Science Content:

This project lies within the general area artificial intelligence, since it involves search techniques such as tree searches and graph searches. More specifically it involves a pathfinding algorithm that aims to find the shortest path in a geometric environment. As such, it also involves some computational geometry. The meshes will need to be evaluated and displayed.

#### Specific Learning Outcomes:

Understanding of a complex graph searching algorithm for practical applications such as robotic path planning; understanding of graph search optimization

#### Skills Required by Team as a Whole:

Basic computational geometry (Working with Meshes); Pathfinding (Knowledge of algorithms such as Dijkstra, A\* and heuristics). These skills can also be acquired during the project

- **Theory:** Moderate. Basic graph theory (as taught in CS2) and knowledge of search algorithms like A\*. Understanding of mesh data structure and path planning algorithms.
- **Implementation:** Moderate. Simon will provide code which evaluates and manages the Field D\* pathing. Most of the work will revolve around new metrics and mesh construction. This should be handled by the CGAL computational geometry library. The implementation should be in C++.

#### Facilities needed:

Existing Field D\* Implementation by Simon Perkins

#### Supervision:

Patrick will provide general supervision; Simon will provide code and assistance with the field D\* algorithm and CGAL libraries.

#### Number of Students:

3 preferred, could reduce it to 2.



## 12. GLYCANO: Building and visualizing sugars.

**Proposer:**

Michelle Kuttel

**Abbreviation:**

GLYCANO

**Brief Description:**

We have an on-going project to develop scientific software tools to aid research into the structure of carbohydrate molecules (these molecules play key roles in disease such as HIV and TB, and are the focus of many modern vaccines). Previous (published) Honours project developed CarbBuilder<sup>1</sup> – a system for building 3D structures of carbohydrates from 2D specification – and novel visualizations for 3D structures<sup>2</sup>. This project will extend this work, developing a tablet app to enable touch-based building of 3D models of carbohydrate molecules as well as new visualizations for carbohydrate 3D structures. This is essentially a design and implementation project, but there is considerable scope for developing the app as an educational game, along the lines of recently published games for building protein structures.<sup>3</sup>

**References:**

1. CarbBuilder: an adjustable tool for building 3D molecular structures of carbohydrates for molecular simulation, Michelle Kuttel, Yue Mao, Göran Widmalm and Magnus Lundborg, Proceedings of the 7th IEEE International Conference on e-Science, 5-8 December 2011, Stockholm, Sweden, pages: 395-402.
2. Techniques for visualization of carbohydrate molecules, M. Kuttel, J. Gain, A. Burger and I. Eborn, J. Mol. Graph. Model., 25, 380-388, (2006)
3. Predicting protein structures with a multiplayer online game, Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popovic & Foldit players, NATURE |, Vol 466, 5 August 2010, 756-760

**Computer Science Content:**

Interface and software design, app development, scientific visualization.

**Specific Learning Outcomes:**

You will learn about principles of visualization and interface design, how to test visualizations for efficacy, how to interact with clients and produce useful scientific software. Ideally, One team member should be interested in 3D visualization design, one in tablet programming.

- **Theory:** No prior experience with Chemistry is required. However, you will have to be willing to learn the basics of molecular structure (which is useful knowledge!). Must not be afraid of a challenge!
- **Implementation:** Implementation will be of average difficulty for an Honours project. The emphasis will be on an interactive design process to obtain a really useful and user-friendly product. New, innovative ideas are welcomed.

**Facilities needed:**

Samsung tablet, standard pc/laptop.

**Supervision:**

You will have weekly meetings with Michelle, together with remote Co-supervision by Göran Widmalm (University of Stockholm, Sweden).

**Number of Students:**

2 (or 3)

## 13. Project: Lifetimes of Migration

### Proposer:

Geoff Nitschke

### Abbreviation:

LOM

### Brief Description:

The project will use agent based simulations to study the impact of lifetime duration on the evolution of migration behaviours in populations. In nature, animals with short life spans tend to have their migration behaviour genetically encoded (e.g. insects) whereas animals with relatively long life spans (e.g. mammals) must learn the migration behaviours during their lifetimes.

The project's objective is to use an agent-based simulation to ascertain the conditions (environmental, learning and evolutionary) under which migration behaviour becomes genetically encoded versus having to be learned during one's lifetime. To investigate this two main approaches will be used; 1) A swarm intelligence approach (Particle Swarm Optimization) combined with lifetime learning; 2) A hybrid evolutionary and learning algorithm. The task is for an agent population to adapt collective migration behaviours so as they learn to periodically migrate across any given virtual world with changing seasons, moving to where resources are most plentiful.

### Computer Science Content:

Agent Based Systems, particle swarm optimisation, hybrid evolutionary and learning algorithms.

### Specific Learning Outcomes:

Experiment design and execution, system development and implementation, theoretical and statistical analysis of results, drawing of conclusions based on analyses as well as observations of empirical data.

### Skills Required by Team as a Whole:

Some knowledge of, and experience with swarm intelligence, hybrid evolutionary and learning algorithms, and agent based systems is desirable. It is suggested that one team member work on combining lifetime learning with particle swarm optimisation and the hybrid evolutionary and lifetime learning method. A third member could work on either only a lifetime learning or evolutionary approach.

- **Theory:** Hybrid evolutionary and learning algorithms, swarm intelligence, agent based simulations.
- **Implementation:** Agent based system – can be written in a variety of languages.

### Facilities needed:

Honours lab computers.

### Supervision:

Weekly meetings, updates, and occasional demos.

### Number of Students:

2 or 3 (most suited for 2).

## 14. Vector Variations

**Proposer:**

Edwin Blake

**Abbreviation:**

mobivect

**Brief Description:**

*Vector art creator for entry-level smartphones*

This project concerns different ways of tackling the problem of producing vector drawings on a smartphone. It is related to our “A mobile visual design application for entry-level smartphones” (mobvisap) project but focuses exclusively on different ways of tackling the important issue of creating vector illustrations. Our recent research show how young people from resource-constrained settings are increasingly using their mobile phones to create and share visual creations – combining elements from their mobile cameras, free in-phone editors, type, and clip art libraries. These young people are finely negotiating the limited capabilities of mobile phones for the purposes of self-expression.

The approaches can be quite independent parts of the project and include:

1. drawing directly on HTML5 canvas with a command line type interface
2. taking a photo of a paper drawing and tracing over it
3. Drawing directly on an interactive phone app

Option 1 can also be supported by drawing tools. The advantage with this option is that it is a web-based approach, which removes platform dependencies.

For Option 2 you will design and build an application for an entry-level smart phone that allows users to transform mobile phone photographs of drawn elements into editable vector objects.

Option 3 (and perhaps 2) will require a pointing device linked to the phone (such as a Bluetooth mouse).

Further common requirements for the options listed above include:

- A tool which can effectively “cut out” illustrations and add transparent background, for the purposes of overlaying these in other compositions.
- A selection of editing tools for vector fine-tuning: isolating and editing curves or lines; dragging anchors and using control handles, rotate and scale
- Exporting these vector elements into library sets

This is part of a larger Nokia/Microsoft research initiative for visual design on mobile, exceptional work may be taken further or considered for inclusion in publication.

**Computer Science Content:**

This project falls in the ICT4D sphere where the challenges are both technical and HCI related. Technically the challenge is to make responsive apps for things like vector drawing and manipulation on a phone. There are also HCI issues related to design and usability of the mobile interface.

**Specific Learning Outcomes:**

mobile programming and performance measurement, visual design. In this project user experiments are optional and heuristic (expert) evaluations may suffice..

**Skills Required by Team as a Whole:**

This project requires a diverse team with both mobile programming and HCI skills.

- **Theory:** Android, or HTML and javascript, Programming; Visualization and 2D graphics; HCI; Experience in vector drawing and graphic design is a plus.
- **Implementation:** This project will include mobile programming and some graphic design concepts that might be unfamiliar. Assistance will be available.
- **Other:** Similar software might exist for more sophisticated hardware, the challenge is to simplify these and present an appropriate interface for a lower-end mobile device and mobile-primary end-user

**Facilities needed:**

Android mobile phones, access to users

**Supervision:**

You will join in activities of the interdisciplinary ICT4D centre. You will have access to mobile programming expertise and will be co-supervised by Dr Marion Walton (Media Studies) and Anja Venter (PhD student and experienced graphic designer). Please see <http://www.cs.uct.ac.za/~edwin/honsProj.html> for more details.

**Number of Students:**

2 or 3

## 15. A mobile visual design application for entry-level smartphones

**Proposer:**

Edwin Blake

**Abbreviation:**

mobvisap

**Brief Description:**

Design and build an application for a mobile device which allows small and micro entrepreneurs to create printed and digital visual materials, and generate branding (e.g., Logos) for a social media presence (Facebook Page, Multimedia Messaging (MMS), BlackBerry Messenger (BBM), WhatsApp).

Creative disciplines are fundamentally involved with digital media in the world today - creative software, social networks and mobile interfaces pervade formal and informal creative practices globally: enabling bedrooms or garages to become micro- recording studios, fashion houses or advertising agencies, to name a few.

Our recent research shows how aspiring creative people overcome constraints to produce, market and network their creative media on mobile phones: a fringe network of free applications, low data costs, Bluetooth sharing, and real world communities which are seldom taken into account in the design of creative applications.

This project will encompass the development of a micro-branding application for lower-end touch screen phones within the bounds of these resource constraints. This project will be split between team members and will include building

- basic image and text compositing tools
- a logo builder, including a vector drawing tool
- include design templates (e.g., Facebook cover page, profile pic, mobile ad)
- export to various image sizes for social media sharing and if possible, printing

This is part of a larger Nokia/Microsoft research initiative for visual design on mobile, exceptional work may be taken further or considered for inclusion in publication

**Computer Science Content:**

This project falls in the ICT4D sphere where the challenges are both technical and HCI related. Technically the challenge is to make responsive apps for things like vector drawing and image manipulation on a phone. There are also HCI issues related to design and usability of the mobile interface.

**Specific Learning Outcomes:**

mobile programming and performance measurement, user experiment design, visual design.

**Skills Required by Team as a Whole:**

This project requires a diverse team with both mobile programming and HCI skills.

- **Theory:** Android Programming, Visualization and 2D graphics, HCI, Experience in vector drawing and graphic design is a plus

- **Implementation:** This project will include mobile programming and some graphic design concepts that might be unfamiliar. Assistance will be available.
- **Other:** Similar software might exist for more sophisticated hardware, the challenge is to simplify these and present an appropriate interface for a lower-end mobile device and mobile-primary end-user

**Facilities needed:**

Android mobile phones, access to users

**Supervision:**

You will join in activities of the interdisciplinary ICT4D centre. You will have access to mobile programming expertise and will be co-supervised by Dr Marion Walton (Media Studies) and Anja Venter (PhD student and experienced graphic designer). Please see <http://www.cs.uct.ac.za/~edwin/honsProj.html> for more details.

**Number of Students:**

2 or 3 (4 possible depending on the scope tackled)

## 16. Information system for protected areas in South Africa

**Proposer:**

Maria Keet

**Abbreviation:**

PA-DB

**Brief Description:**

Biologists and ecologists collect a lot of data that needs to be stored and analysed so that the frontiers of their state of the art can be pushed forward. In this case, a lot of data in different formats and types has been collected and digitised about protected areas in South Africa. This includes, amount others, SANPARKS data about visitors and infrastructural aspects, interviews with about 100 people involved in management of both public and private parks and their social interaction, permit data about translocation of animals, and more. This has to be integrated and consolidated into a relational database that takes into account current trends in ecology data management standards and will have a neat user-useable front end that enables the biologists to analyse the data without having to formulate cumbersome large queries in SQL. Ultimately, from the conservationist viewpoint, the system will help analysing the data, which will lead to a better understanding of conservation biology with its policies and practices in South Africa.

**Computer Science Content:**

Databases, Information Systems, possibly also ontology-based data management

**Specific Learning Outcomes:**

Data integration, international data standards, system development

**Skills Required by Team as a Whole:**

- **Theory:** databases, possibly some NLP and/or OBDM
- **Implementation:** creation of an RDBMS with 'intelligent' front-end to the database

**Facilities needed:**

Standard networked PC, tool development environment, RDBMS.

**Supervision:**

The proposer is the supervisor, and the project will involve collaboration with Prof. Cumming and his students from the Percy FitzPatrick Institute of African Ornithology. There will be regular project meetings and meetings with members of the FitzPatrick Institute. Data will be made available, as well as technical background literature.

**Number of Students:**

2



## 17. Enriching UML class diagrams with the taxonomy of part-whole relations

**Proposer:**

Maria Keet

**Abbreviation:**

PW-UML

**Brief Description:**

Designing good conceptual data models is not easy, but advances are being made in specifying guidelines informed by ontology and logic foundations. Concerning UML Class Diagrams, and in particular the shared and composite aggregation association (i.e., part-whole relations), multiple problems have been identified over the years due to their underspecification in the UML standard. The main issues have been resolved mainly thanks to the latest instalment of a taxonomy of part-whole relations. However, this has not been designed for UML conceptual data modelling regarding neither how to include it graphically nor its proposed decision diagram or any other selection method to choose the most appropriate part-whole relation. Its non-availability in a modelling tool leads modellers to continue making part-whole/aggregation association modelling mistakes. You will investigate the possibilities to extend UML Class Diagrams and implement a tool that helps the modeller to make use of the taxonomy of part-whole relations during the modelling process, integrated in an existing UML CASE tool, using a decision diagram and a foundational ontology. This is then to be evaluated on comprehension and usability.

A selection of some background information:

1. Keet, C.M., Fernández-Reyes, F.C., Morales-González, A. Representing mereotopological relations in OWL ontologies with OntoPartS. 9th Extended Semantic Web Conference (ESWC'12), Simperl et al. (eds.), 27-31 May 2012, Heraklion, Crete, Greece. Springer, Lecture Notes in Computer Science LNCS 7295, 240-254.  
<http://www.meteck.org/files/OntoPartSESWC12.pdf>
2. Keet, C.M. and Artale, A. Representing and Reasoning over a Taxonomy of Part-Whole Relations. Applied Ontology -- Special Issue on Ontological Foundations for Conceptual Modelling, IOS Press, 2008, 3(1-2): 91-110. [http://www.meteck.org/files/AO07\\_pw\\_AK.pdf](http://www.meteck.org/files/AO07_pw_AK.pdf)
3. Keet, C.M. Part-whole relations in Object-Role Models. 2nd International Workshop on Object-Role Modelling (ORM'06), Montpellier, France, Nov 2-3, 2006. In: OTM Workshops 2005. Meersman, R., Tari, Z., Herrero., P. et al. (Eds.), Lecture Notes in Computer Science LNCS 4278. Berlin: Springer-Verlag, 2006. pp1116-1127.  
<http://www.meteck.org/files/ORM06parts.pdf>

**Computer Science Content:**

Software engineering research, with theory on conceptual data modelling and tool development and experimental evaluation.

**Specific Learning Outcomes:**

Extending UML Class Diagram language (metamodel, profile), software development, experimental evaluation of augmented UML modelling.

### Skills Required by Team as a Whole:

- **Theory:** UML, modelling (optionally including knowledge of a foundational ontology), algorithm development, how to conduct an experimental evaluation of software
- **Implementation:** mainly C++ or Java, code repositories, usage of freely available UML tool or Icom tool (and its XML serialisations). The software itself is not difficult to design.

### Facilities needed:

PC with C++ or Java development environment and a open source UML tool. Note: if ORM is preferred over of UML, then MS Visio + NORMA (one source).

### Supervision:

There will be regular project meetings. Background material will be provided. If the Icom tool is chosen to extend, then there will be contact also with its lead programmer Prof. Pablo Fillottrani from the Universidad Nacional del Sur, Argentina, within the SA-ARG bilateral project.

### Number of Students:

2

## 18. Visualization of Radio Frequency Interference data for MeerKAT/SKA

### Proposer:

Michelle Kuttel

### Abbreviation:

RFI\_VIS

### Brief Description:

The MeerKAT radio telescope is currently being built on the site of the planned SKA radio telescope. All radio telescopes suffer from Radio Frequency Interference (RFI) from radio signals generated by terrestrial or man-made sources, such as networks, satellites, aeroplanes, lightning, cameras, cell phones and laptops. It is important to limit RFI at the MeerKAT site, as RFI obscures astronomic signals. There are many mitigation strategies in place for managing the RFI environment, including legislation against emitters around the core site, RFI monitoring, RFI detection and RFI excision. Christopher Schollar, an M.Sc. student now working for the SKA, has developed a prototype RFI monitoring system to record of RFI at the MeerKAT site. MeerKAT staff access this data via two simple online visualisations. However, there is considerable potential to improve these visualizations (to allow for easier identification of RFI) as well as implement alternate algorithms to identify or detect RFI, including RFI cycles.

### Computer Science Content:

Visualization, novel algorithms, web programming.

### Specific Learning Outcomes:

You will learn about the SKA and RFI, principles and practice of visualization and interface design, how to test visualizations for efficacy, how to interact with clients and produce useful scientific software.

### Skills Required by Team as a Whole:

Both students can focus on alternate visualizations, or one could focus on visualization and one on detection/identification of RFI.

- **Theory:** *Absolutely no prior experience with SKA or Astronomy is required. However, an interest in Astronomy and the SKA is assumed: you will have to be willing to learn the basics of RFI detection.*
- **Implementation:** *Implementation will be of average difficulty for an Honours project. The emphasis will be on an interative design process to obtain a really useful and user-friendly product. New, innovative ideas are welcomed.*

### Facilities needed:

Standard CS facilities

### Supervision:

You will have weekly meetings with Michelle, together with co-supervision by Christopher Schollar (who developed the system) as well as Sarah Blyth(UCT Astronomy) and Anja Schroeder, as required.

**Number of Students:**

2 –preferred ( 3 possible)

## 19. Fast and accurate triangle mesh simplification for large heritage models

### Proposer:

Patrick Marais

### Abbreviation:

SIMPLIFY

### Brief Description:

The Zamani project ([www.zamaniproject.org](http://www.zamaniproject.org)) at UCT uses laser scanning to capture huge 3D models of buildings and terrain for digital preservation and analysis. In order to reduce the size of the models, they would like to simplify them, so that they retain important structural information while retaining their original appearance. There are two methods that will be utilised: simplification with normal maps, and feature preserving simplification. Normal maps approximate missing detail by means of shader-based techniques and can produce compelling results with low triangle counts. Feature-preserving approaches tend to reduce the number of triangles required to represent less complex regions, and only keep high detail where it is really required in the model. Once simplification is complete, the visual fidelity of the result must be assessed. This can be done by means of a visualisation front-end which shows the distribution or error caused by simplification. Various visual fidelity metrics can be used to quantify the simplification error too. A major challenge for simplification will be dealing with models that are very large and cannot fit into memory: this will require an 'out-of-core' solution.

### Computer Science Content:

Data structures and algorithms for 3D mesh representations - specifically those which are memory efficient; computer graphics and visualisation; computer vision/image processing.

### Specific Learning Outcomes:

Understanding of 3D model manipulation, display and processing; basic understanding of visualisation principles; general algorithmic optimisation for memory efficiency and speed.

### Skills Required by Team as a Whole:

The project will consist of 3 components: 2 which implement different simplification schemes in an out-of-core manner and a third which can be used to sensibly visualise and quantify the error caused by each of the simplification schemes. Consequently, the two algorithmic projects will require good coding and optimization skills and the visual project will require knowledge of rendering APIs and a willingness to learn visualisation techniques.

- **Theory:** (Avg to hard) 3D meshes and geometry; simple geometric operations on meshes; visualisation; 3D rendering.

- **Implementation:** (Avg to Hard)

The techniques used will require extensive reading and are not trivial to implement. Some techniques are better implemented using libraries. You might consider CGAL for geometric operations and VTK for visualisation. These are both Open Source/available for academic use. Out-of-core optimizations can also be complex, depending on how closely you manage disk access.

You will need to read up on popular mesh simplification and methods as well as methods to construct normal maps. Papers and suggested readings will be provided by your supervisor.

**Facilities needed:**

Standard computer lab equipment should suffice although more server space may be required for the model files, which are large. The visualisation component will require access to a computer with a decent graphics card.

**Supervision:**

This projects will require interaction with Prof Ruther and his group at the UCT Geomatics Dept. The students and supervisor will meet weekly initially and then bi-weekly to ensure progress is adequate.

**Number of Students:**

3. This projects can be reduced to 2 people if required.

## 20. Tick-borne Diseases Database

**Proposer:**

Sonia Berman

**Abbreviation:**

TickDB

**Brief Description:**

Ticks and tick-borne diseases have a major impact on both human health and animals, including livestock, with corresponding losses across all sectors. While it has been fairly well investigated in Europe and the USA, this is much less the case for Africa, which has a different climate, animals etc. Data is being collected, digitised, and analysed about ticks and tick-borne diseases, the host of the ticks, their location, and so on. Currently, this data resides in various repositories, such as an MS Access database, excel sheets, and maps, which have to be integrated into one repository for effective and efficient data analysis by biologists. The aim of this project is to integrate it into one database that is usable by scientists (e.g. in biology) as a minimum, and then to extend it with further analysis through natural language processing and/or (spatial) data visualization and analysis.

**Computer Science Content:**

Databases

**Specific Learning Outcomes:**

Database management, data integration, data visualization, NLP

**Skills Required by Team as a Whole:**

- **Theory:** This is not a theoretical project.
- **Implementation:** Working with relational databases, preferably with a web-based front-end, and data visualization or NLP

**Facilities needed:**

No special equipment required.

**Supervision:**

Weekly meetings with supervisor. The project will involve collaboration with Prof. Cummings and his students from the Percy Fitzpatrick Institute of African Ornithology. Data will be made available by Prof Cummings.

**Number of Students:**

2

## 21. Data Structures Tutor

**Proposer:**

Audrey Mbogho

**Abbreviation:**

Tutor

**Brief Description:**

Manipulation of trees and graphs forms a significant part of the second year data structures course. Some students find it hard to adjust to this type of material after getting used to programming in first year. What could help students is practicing performing these manipulations, for instance AVL rotations repeatedly until they learn them. The course as it currently runs provides few opportunities this to happen since practical work consists entirely of programming assignments. The aim of this project is to build software for manipulating various data structures in a game-like platform so that rewards can be given for successful completion of steps in order to motivate repeated use of the software. Intelligence should be built into the software so that it recognises when a student struggles with a particular concept and gives him/her more drills of the type that teaches that concept. It could also offer hints where needed.

**Computer Science Content:**

Intelligent tutoring, Educational technology, Human computer interaction

**Specific Learning Outcomes:**

Software development and evaluation

**Skills Required by Team as a Whole:**

Programming

- **Theory:** *Moderate*
- **Implementation:** *Challenging*

**Facilities needed:**

Lab computers should suffice

**Supervision:**

Weekly meetings with supervisor

**Number of Students:**

3



## 22. Pseudo-natural language rendering (verbalization) of OWL 2 ontologies

**Proposer:**

Maria Keet

**Abbreviation:**

VerbOWL

**Brief Description:**

Domain ontology development invariably involves interaction with subject domain experts, who are normally not logicians or computer scientists, hence, are unfamiliar with logics such as OWL and one cannot expect them to become a logician. In addition, with the increase of ontology-driven information systems and software localization and internationalization, some way to easily access the knowledge represented in the ontology becomes a necessity (e.g., for the Indigenous Knowledge Base project [NIKMAS] of the Department of Science and Technology). One way to bridge this well-known impasse is to verbalize the ontology, i.e., to generate a (pseudo-)natural language rendering of the axioms. This has been done successfully mainly for English, whereas preliminary investigations into grammatically richer languages, such as Spanish, German, Lithuanian, and isiZulu, reveal that the common template-based approach is insufficient and at least a partial grammar engine (a set of rules) may be required to generate the verbalizations. As a bare minimum, you will investigate the issues of verbalizing logical theories and write an application (web-based or a Protégé plugin) that verbalizes an OWL ontology in a natural language of choice (except English). Depending on the chosen language and the amount of students in the group, there are few variations for extra features, such as a two-way information flow recognizing the axiom in the OWL file based on the phrase, incorporating a dictionary lookup, usage of the Lemon model for multilingual ontologies (or the outcome of project MultiKR), user evaluation on understandability of the generated sentences, and covering more or less language features.

**More information:**

1. Attempto Controlled English (ACE) at <http://attempto.ifi.uzh.ch/> and some challenges to do that for multiple non-English languages:
2. M. Jarrar, C.M. Keet, and P. Dongilli. Multilingual verbalization of ORM conceptual models and axiomatized ontologies. STARLab Technical Report, Vrije Universiteit Brussels, Belgium. [http://www.meteck.org/files/ORMmultiverb\\_JKD.pdf](http://www.meteck.org/files/ORMmultiverb_JKD.pdf)

Some additional material is available on verbalization in isiZulu, which will be made available if the topic is chosen.

**Computer Science Content:**

Artificial Intelligence, Computational Linguistics, Natural Language Generation

**Specific Learning Outcomes:**

Understanding of and skills in Semantic Web Technologies, internationalization of software, software development, (optionally) grammar engine

**Skills Required by Team as a Whole:**

- **Theory:** ontologies, Semantic Web, algorithm development, at least some familiarity with the chosen language and (preferably) the notion of a grammar

- **Implementation:** Java, optionally with a web-based front-end. This can become quite challenging, especially with respect to the grammar rules for the Nguni languages

**Facilities needed:**

PC with the usual software development environments and Semantic Web Technologies development tools.

**Supervision:**

There will be regular project meetings with the supervisor and, if other language-oriented projects are chosen, also group meetings. Background material will be provided. Depending on the chosen language, also relevant linguists may be involved.

**Number of Students:**

2-4

## 23. Virtual Panning for Lecture Recording

### Proposer:

Patrick Marais

### Abbreviation:

VIRPAN

### Brief Description:

To record lecturers as they move about the front of a class, one could use an expensive tracking (PTZ) camera. This project aims to investigate whether it is possible to replace this camera with 3 fixed cameras, and to build up a 'virtual panning' system from these 3 camera feeds. This would require the use of image analysis to synthesize a large virtual image and to crop this down to the lecturer centred video stream. To do this effectively, appropriate algorithms need to be developed to keep the presenter in the field of view, ensure smooth tracking motion, and to avoid obvious image artefacts where the camera images are joined. This project will be run in conjunction with the UCT Centre for Educational Technology (CET) and the solution devised may be implemented in larger lecture venues at UCT if it is sufficiently robust.

### Computer Science Content:

Image Processing, Computer Vision; data structures and algorithms

### Specific Learning Outcomes:

The students will learn how basic image segmentations and compute vision algorithms can be applied to streaming video. This will include an appreciation for performance issues that arise from real-time video processing

### Skills Required by Team as a Whole:

The team should have C++ programming skills (efficient algorithms are key) as well as some people doing the image processing course. It is suggested to use C++ and OpenCV for coreCV and image processing algorithms.

- **Theory:** Intermediate. Basic understanding of image registration, segmentation and image processing required. Patrick will advise on the most suitable approaches.
- **Implementation:** Intermediate. As far as possible, third party open source software like OpenCV should be used to do underlying image processing. However, efficiency will be key and benchmarking will determine what you need to implement.

### Facilities needed:

Appropriate hardware (cameras), data (source recordings) and test venues will be provided by the CET.

### Supervision:

Supervision will be mainly provided by Patrick with regular input and guidance from Stephen Marquard of the CET.

### Number of Students:

3 preferred, 2 maybe possible

## 24. Zamani Data Archive

**Proposer:**

Hussein

**Abbreviation:**

ZAMANI

**Brief Description:**

The Zamani Project, based at the UCT Geomatics Department, needs an online archive for its large spatial data collection of cultural heritage sites. The data comprises of highly detailed 3D models (from 50MB up to 2-5GB per building), photogrammetric and 360 degree panorama imagery, Geographic information Systems (GIS), sections & elevations, plans and videos. The Zamani Project has thus far documented about 40 sites in 12 African countries, with close to 100 3D models of individual structures. The volume of all African Heritage Sites documented so far is estimated to be in the order of 10-20TB!

There are two major aspects to the problem: archival storage management and end-user presentation. A management system is needed to organize the metadata as a layer over the current drive-based data-store. An end-user system should then allow easy retrieval via a Web interface, with appropriate search and discovery metaphors for location-oriented multi-format data. Datasets to be stored are collections of files associated with sites and the relationships among the files should be maintained in the archive and exposed to users during the discovery process. Metadata is partial and of varying quality – part of the solution will include automating the generation of metadata for the different heritage-oriented file formats used.

**Computer Science Content:**

Digital Libraries, heritage preservation, archiving

**Specific Learning Outcomes:**

Research Methodology; Experiment design and execution; Real-world software development

**Skills Required by Team as a Whole:**

**Theory:** Nothing specific.

**Implementation:** Knowledge of XML, HTML, JS and a scripting language – but most can be learnt “on the job”.

**Other:** Excitement about the preservation of Cultural Heritage!

**Facilities needed:**

Web server, Large data storage system, Data and Metadata – all will be provided

**Supervision:**

Hussein, regular contact with Digital Libraries research group and ICT4D centre for feedback and assistance, client in UCT Department of Geomatics

**Number of Students:**

2-3

# Student- proposed honour's project

## 25. Map based visual optimization of Hotel Booking Systems

**Proposer:**

Grace Jegede and Lynray Barends (student proposed project)

**Abbreviation:**

HOBOVIS

**Brief Description:**

**Map based visual optimization of Hotel Booking Systems**, this will include an improved visualization, recommender system, IR and or database management. Hotels registered to this system will have details of their available rooms visualised on the map based system

**Computer Science Content:**

Visualization, recommender Systems, databases, software engineering

**Specific Learning Outcomes:**

Improving visualizations, Database management,

**Skills Required by Team as a Whole:**

- **Theory:** Visualisation, large data-set management
- **Implementation:** redesign, data collection

**Facilities needed:**

Standard Hons machines.

**Supervision:**

Michelle Kuttel

**Number of Students:**

2 or 3