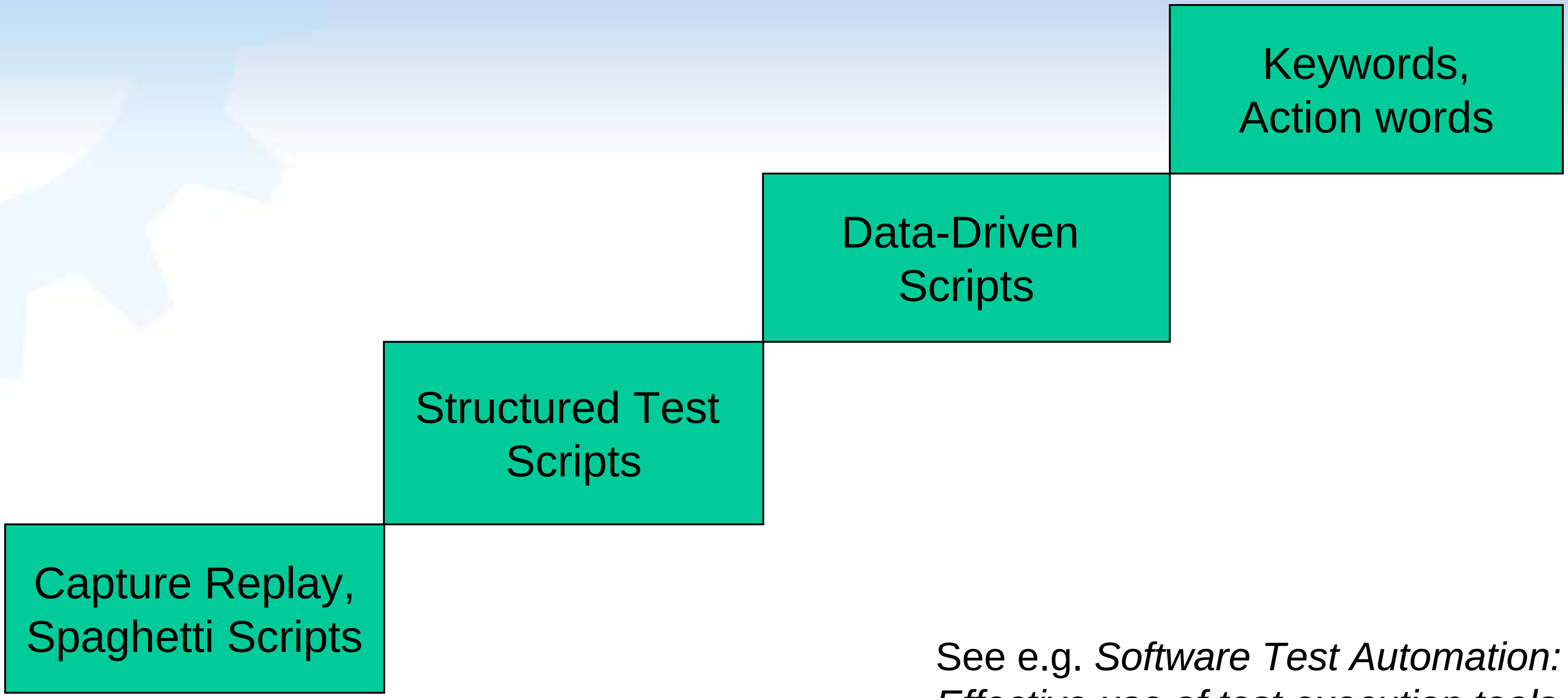
Towards Deploying Model-Based Testing with a Domain-Specific Modeling Approach

29th-31st August, 2006 @ TAIC PART, Windsor, UK

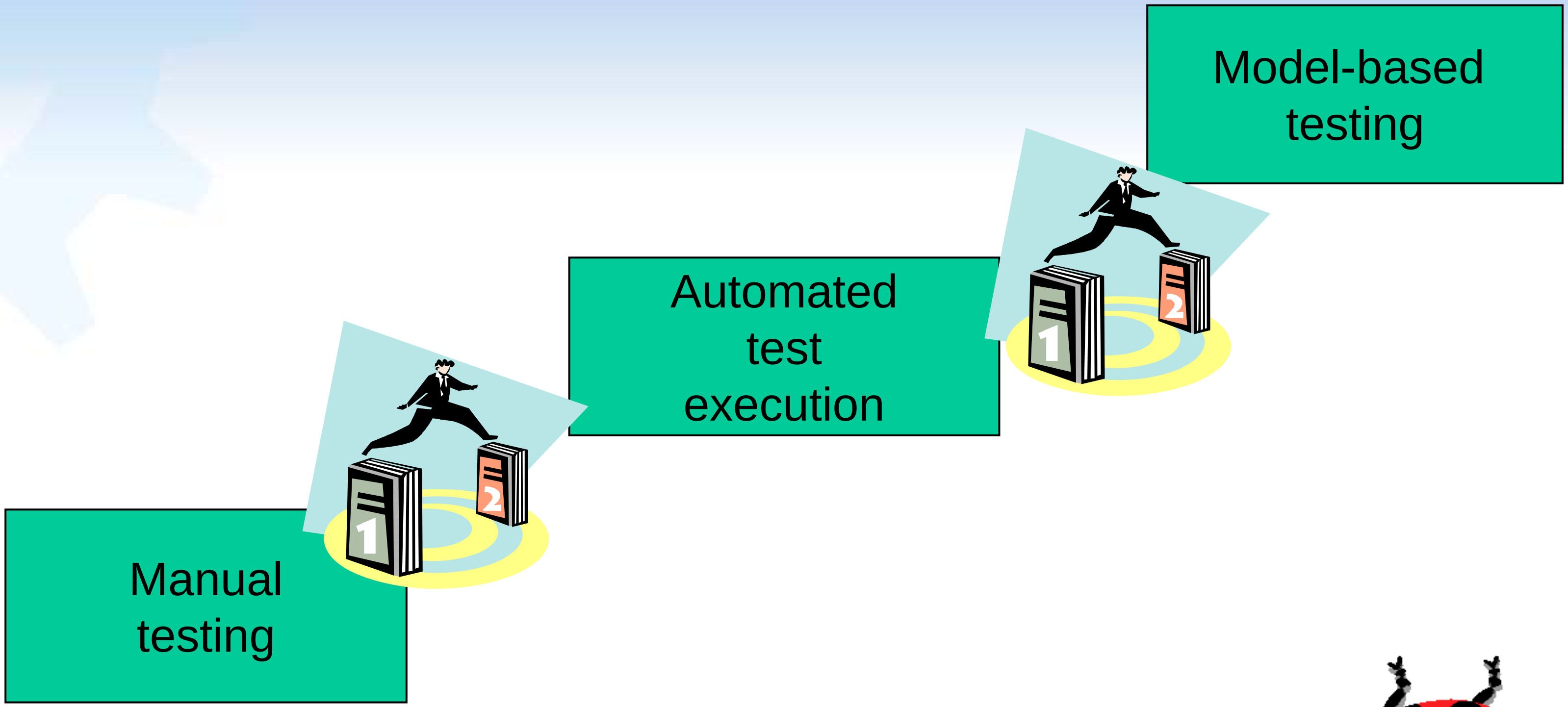
Mika Katara, TUT
Antti Kervinen, TUT
Mika Maunumaa, TUT
Tuula Pääkkönen, Nokia Technology Platforms
Mikko Satama, TUT

Background: Generations of System Level Test Automation



See e.g. *Software Test Automation: Effective use of test execution tools*
By Mark Fewster and Dorothy Graham, Addison Wesley, 1999.

… and the Future?



TEMA Project: Academic & Industrial Collaboration

Tampere University of Technology/Institute of Software Systems
TEKES, the Finnish Funding Agency for Technology and Innovation
Nokia
Conformiq Software
F-Secure
Plenware Group
Mercury Interactive

A general goal of the project:
Industrial deployment of MBT in GUI testing of Symbian S60 smart phones
”Nokia alone has cumulatively shipped 50 million S60 enabled devices by end of February 2006” (Source: www.s60.com)

Disclaimer: these slides represent the views of the presenter, not necessarily the views of the above or any other parties

Pros and Cons of Model-Based Testing

Pros:

- Higher level of **abstraction** helps to concentrate on the right things – details are hidden
  - Better chances when fighting against the increasing complexity
- Models (small ones) can be **visualized** easier than code – better comprehension
- Automatic test generation – better **coverage**
  - **Maintenance** should be easier also – not much studied subject

Cons:

- Based on reported industrial experiences **deployment** can be very challenging
- ”Bubbles don’t crash” – easier to **ignore** real problems

Regarding the deployment, in our experience, two big questions are

- ”Where Do We Get the Test Models”
  - Learning?
  - Reverse engineering?
- ”How Does this Change the Way we Work?”
  - Who is going to do the modeling?
  - What skills are needed?

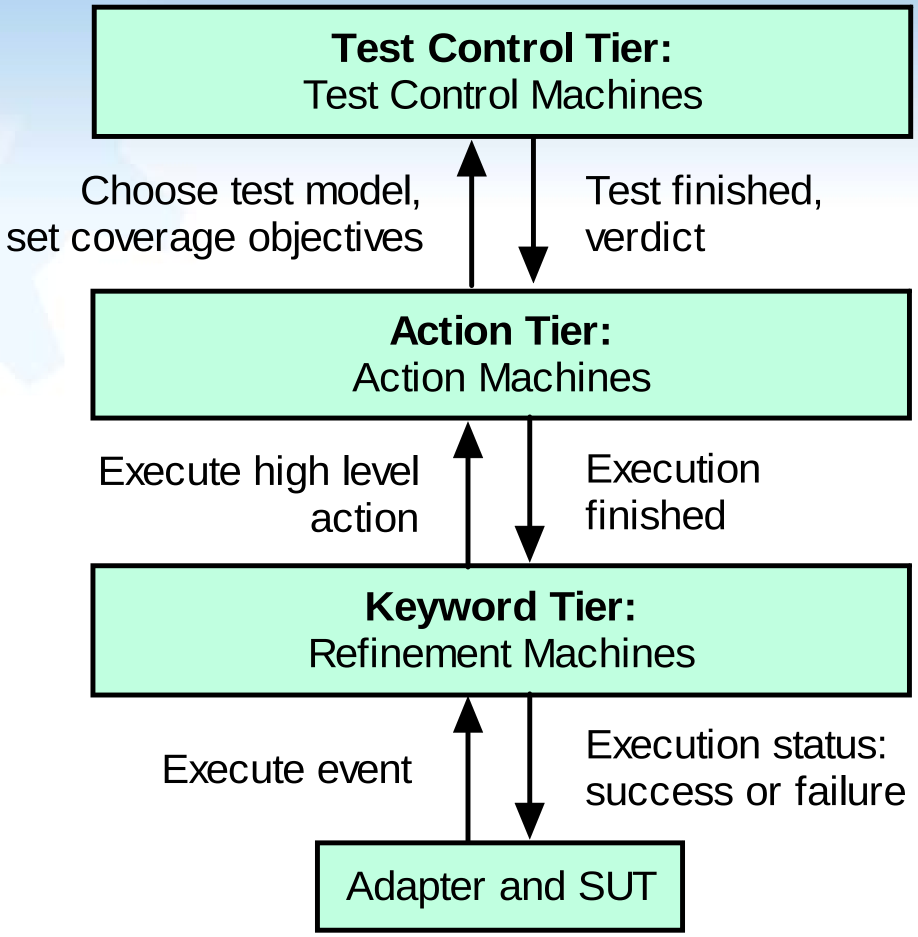
Our proposal:

1. Make test modeling as easy as possible
2. Generate some parts of the models automatically by capturing GUI events
3. Adapt your organization

Domain-Specific Test Modeling of Product Families

- Domain-specific modeling pays off when the domain is stable
- In a product family context there are some parts that stay the same and some things that change
- Our context:
  - Symbian OS, an operating system for smart phones
  - The look & feel stays the same across the family of phones using S60 GUI platform on top of Symbian OS
  - In our approach, the smart phones are tested through a GUI using commercial test automation software
  - For this purpose, we have defined a domain-specific test modeling language based on action words and keywords

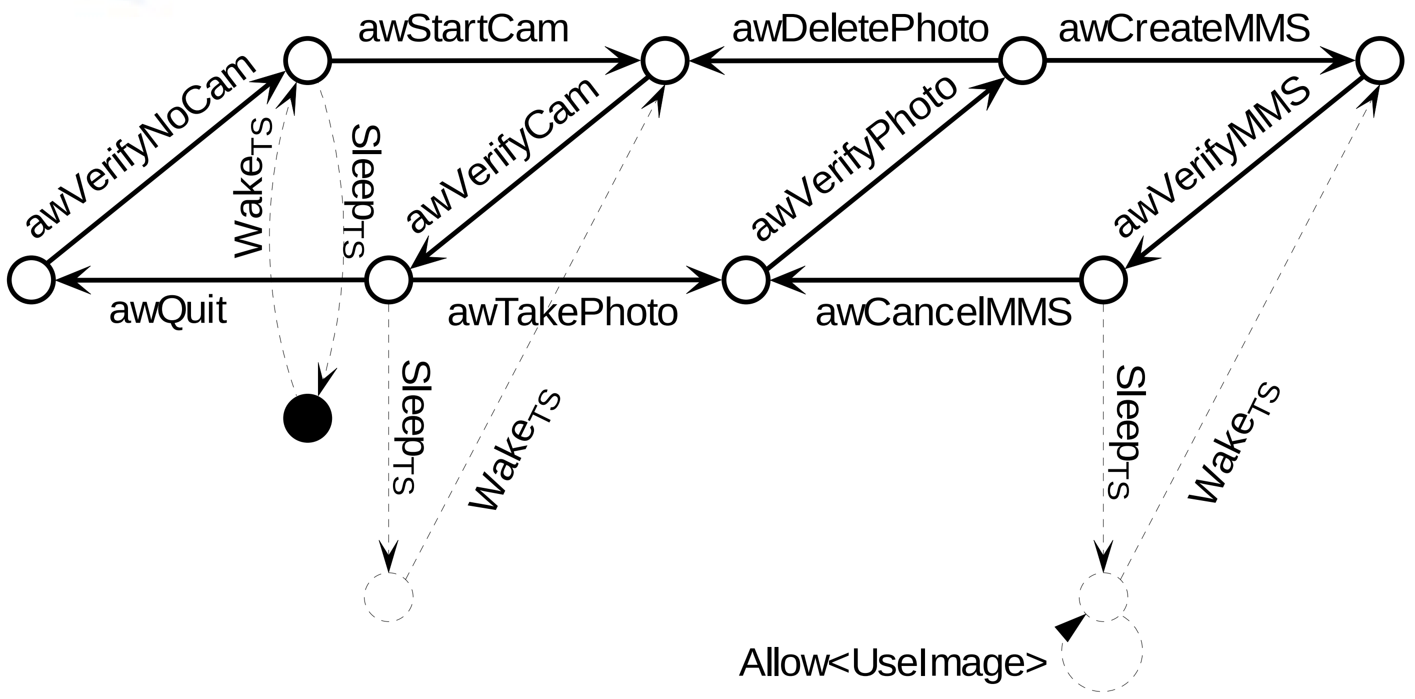
TEMA 3-Tier Test Model Architecture



Antti Kervinen, Mika Maunumaa, and Mika Katara : "Controlling Testing using Three-Tier Model Architecture", Proc. Second Workshop on Model Based Testing (MBT 2006), Vienna, Austria, March 2006.

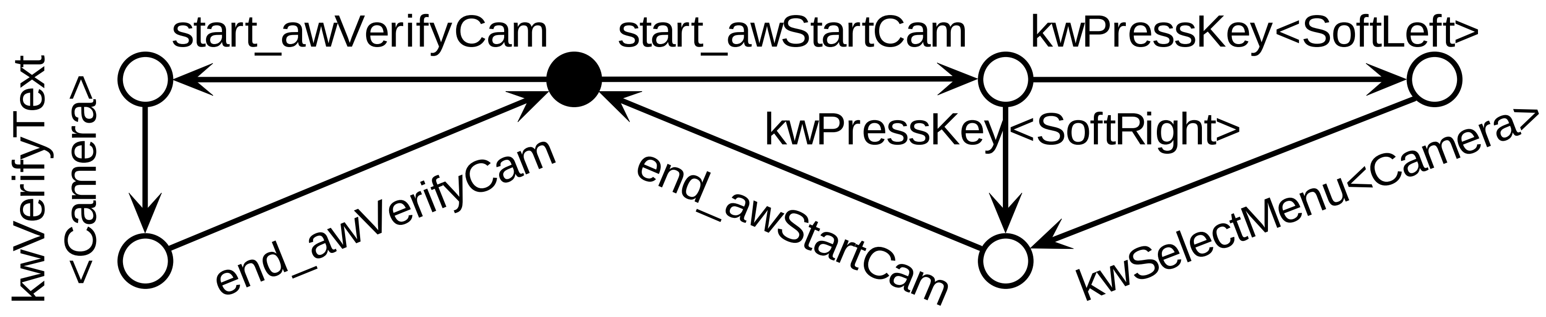
Example Action Machine

S60 Camera application, action word model



Example Refinement Machine

S60 Camera application, keyword model



Two Ways to Build Test Models

Top-Down

- First create action words, then keywords that implement them

Bottom-up

- Keywords are created first, then action words

The two ways complement each other, both are needed in different cases

Based on our experiments, models are created using a mixture of these techniques

Bottom-up

In this approach, the test model can be built from scratch using an event capturing tool called *Recorder*

The tool is capable of capturing GUI events much in the same way as conventional capture/replay GUI test tools

However, instead of producing hard-to-maintain test scripts, our tool produces a list of keywords corresponding to the sequence of GUI events.

Test modeling begins by starting *Recorder* and recording a test using the phone GUI on PC

After recording a test, we split the keyword sequence into action words with the Model Designer tool that reads in the generated keyword sequence and visualizes it as LTS (Labeled Transition System)

Top-down

The top-down approach can be used for maintaining the test models.

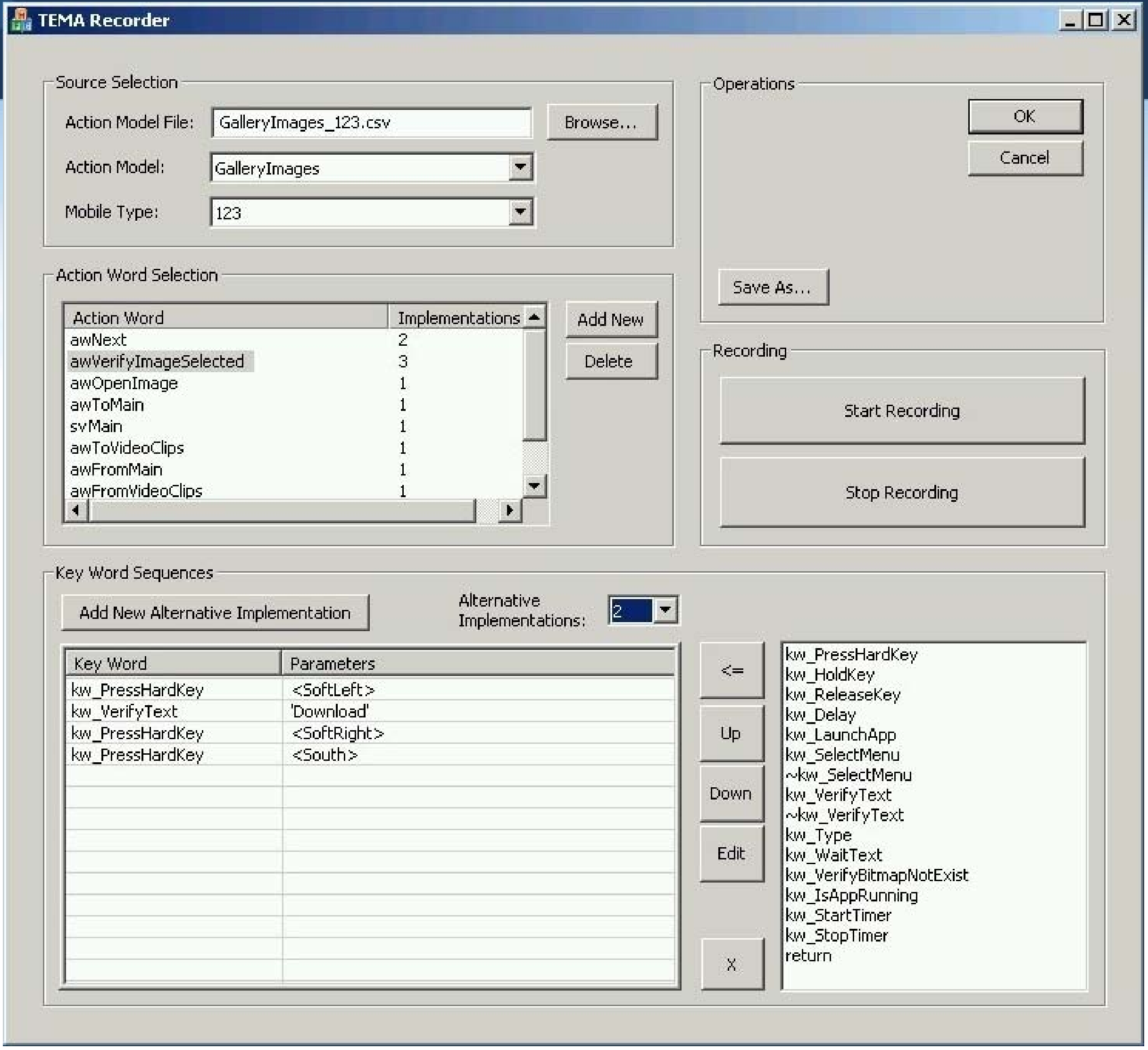
This approach starts from an action word model and Recorder is be used to record the keyword sequences

Model Designer is used for selecting the action word whose implementation needs to be defined

After selecting such an action word, Recorder is started to record the sequence the user inputs

The user can also choose to override an existing keyword sequence by recording a new one.

TEMA Recorder



TEMA Recorder

Source Selection

Action Model File: GalleryImages_123.csv

Action Model: GalleryImages

Mobile Type: 123

Action Word Selection

Action Word	Implementations
awNext	2
awVerifyImageSelected	3
awOpenImage	1
awToMain	1
svMain	1
awToVideoClips	1
awFromMain	1
awFromVideoClips	1

Key Word Sequences

Alternative Implementations: 2

Key Word	Parameters
kw_PressHardKey	<SoftLeft>
kw_VerifyText	'Download'
kw_PressHardKey	<SoftRight>
kw_PressHardKey	<South>

```
kw_PressHardKey
kw_HoldKey
kw_ReleaseKey
kw_Delay
kw_LaunchApp
kw_SelectMenu
~kw_SelectMenu
kw_VerifyText
~kw_VerifyText
kw_Type
kw_WaitText
kw_VerifyBitmapNotExist
kw_IsAppRunning
kw_StartTimer
kw_StopTimer
return
```

Adapting a Testing Organization

- Test Manager role:
  - define the entry and exit criteria to the test model execution
  - define coverage criteria and define which metrics are gathered
  - communicating the testing technology aspects
    - how model-based testing compares to conventional methods
    - advocating reasons for and against using
- Test Modeler role:
  - creates models based on product specifications
  - can be responsible of designing the execution of the model
  - can be responsible of setting up the environment accordingly
- Test Model Execution Specialist role:
  - follows up the test execution across the test model
  - ensure that the model is used according the agreed principles and data
  - reporting the results and faults onwards
  - document the test model usage and testware architecture to enable reuse

Conclusions

Status of the project: one year has passed, two years to go

We are currently implementing the necessary tools and conducting case studies

The biggest challenge still lies ahead: migrating from lab environment to real production

For further information contact:

Mika Katara

mika.katara@tut.fi

Visit our web site at http://practise.cs.tut.fi (TEMA project)