Formal analysis of user interfaces: state of the art and future prospects

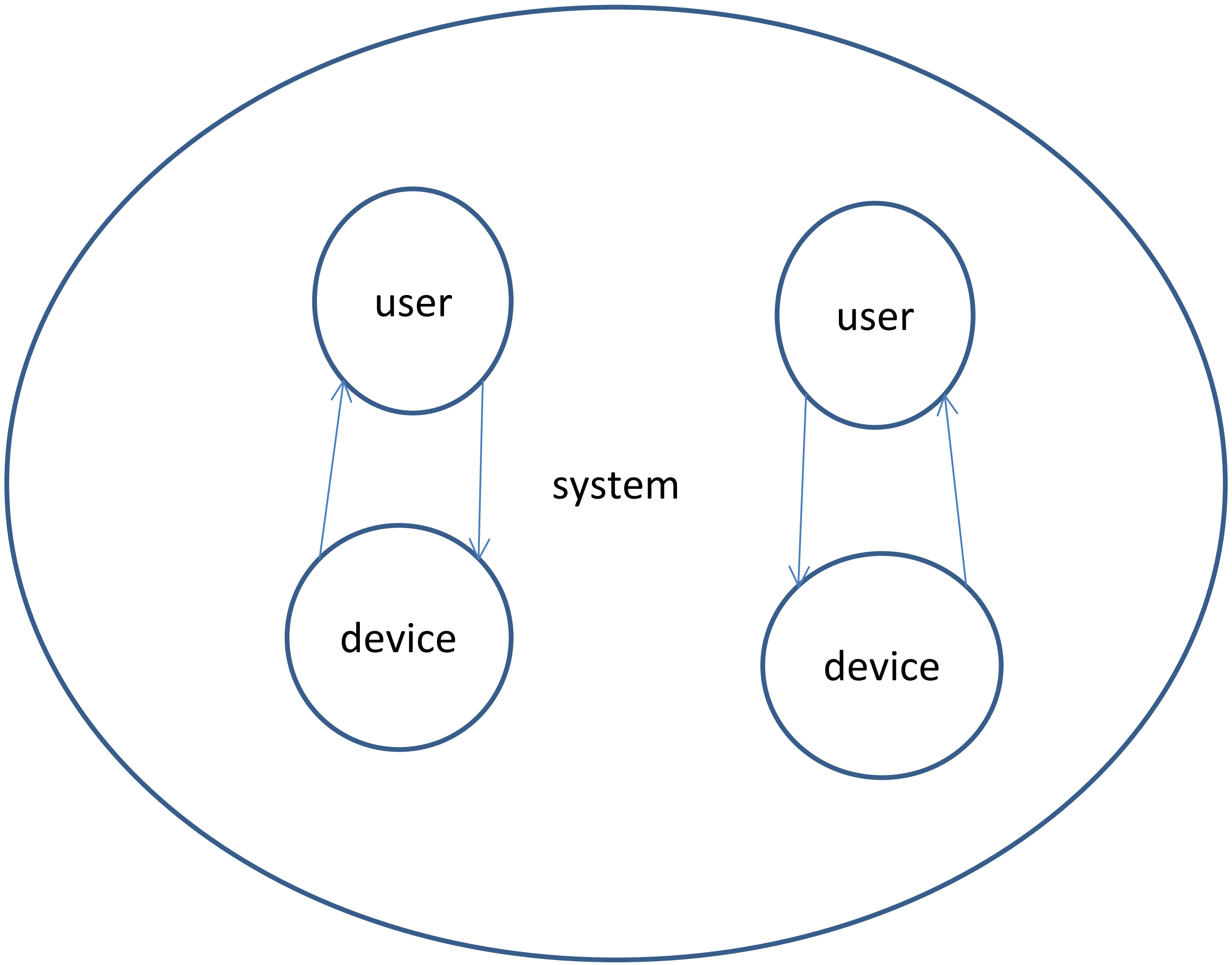
Michael Harrison

Newcastle University, UK

State of the art

- Descriptive role of formalism
  - Device
  - User
  - Interaction
  - The whole system
- Analysis approaches
  - Clarity of expression
  - Generation of prototypes or animations
  - Automated analysis (theorem proving, model checking)
- Relevance of results
  - direct vs. indirect (i.e. property of the dialogue vs property of the model)
  - quantitative vs. qualitative
  - cognitive plausibility?

Device, user, interaction



Modelling the device

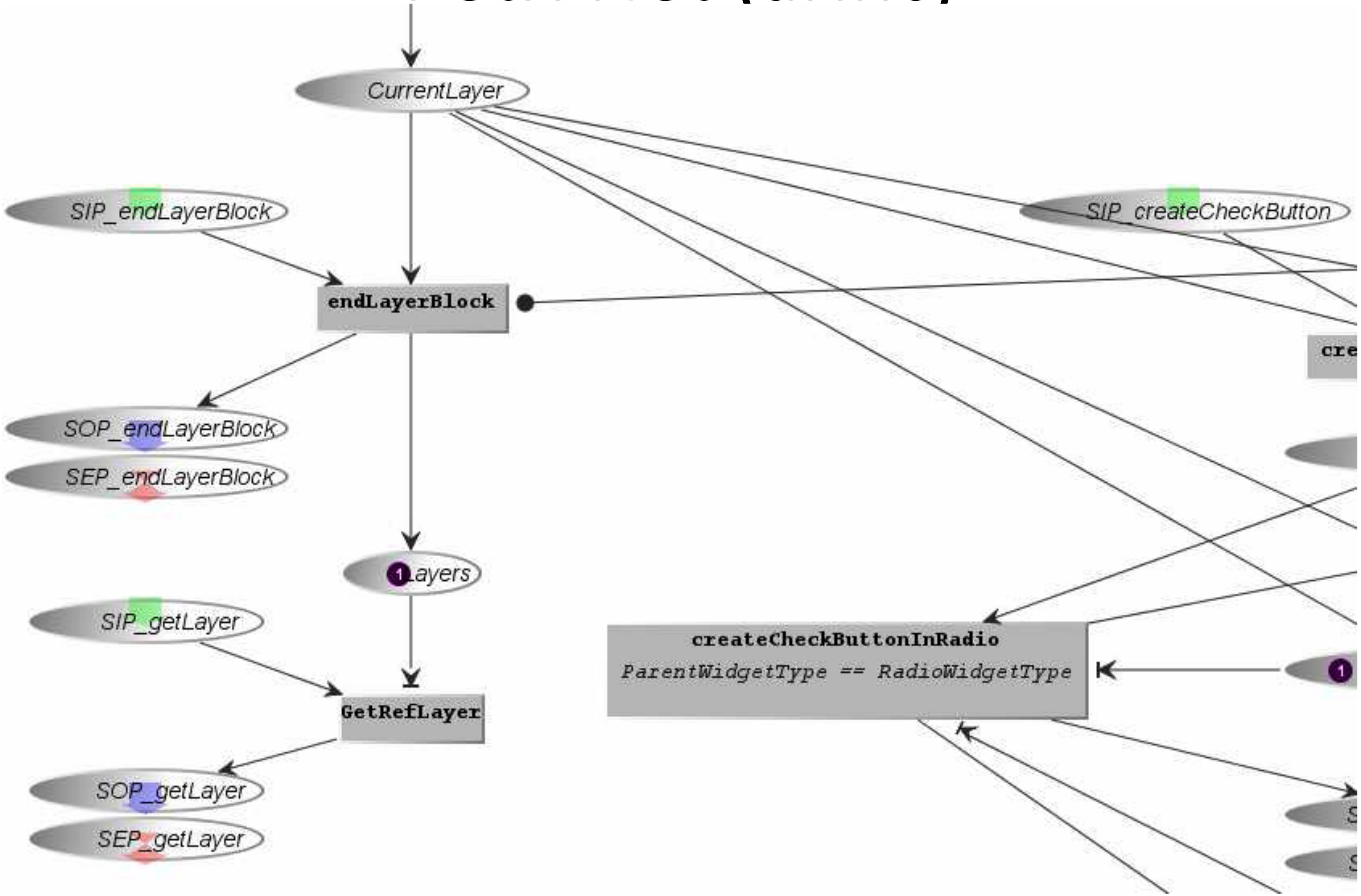
- We are able to model devices (behaviour)
  - Petri nets
    - Formalising standards (Barboni et al., 2007)
  - Graph and finite state models
    - Medical instruments (Thimbleby&Gow, 2007)
    - Markov models (Cairns & Thimbleby)
    - VEG (Berstel et al)
    - SMV (Dwyer)
  - Interactors as a structuring construct
    - Using LOTOS (Paternò, 1999)
    - Using variety of other formalisms (Duke&Harrison; ofan, Degani)
- More focus on model checking and therefore behaviour less focus on structure and nature of the display

ARINC based interface

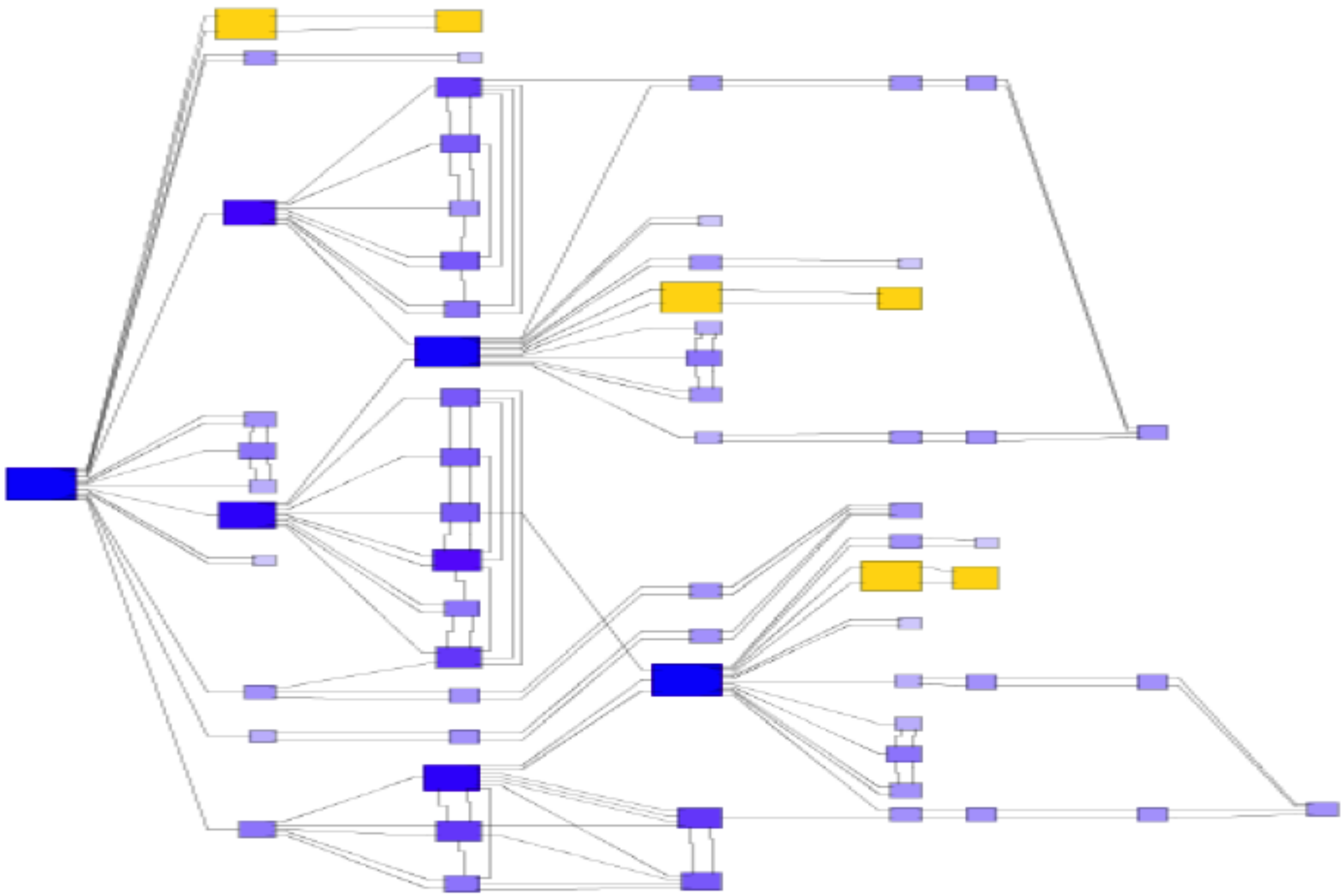
- Aviation widget standard
- Based on a variant of Petri nets

- Focus on semantic clarity

Petri Net (arinc)



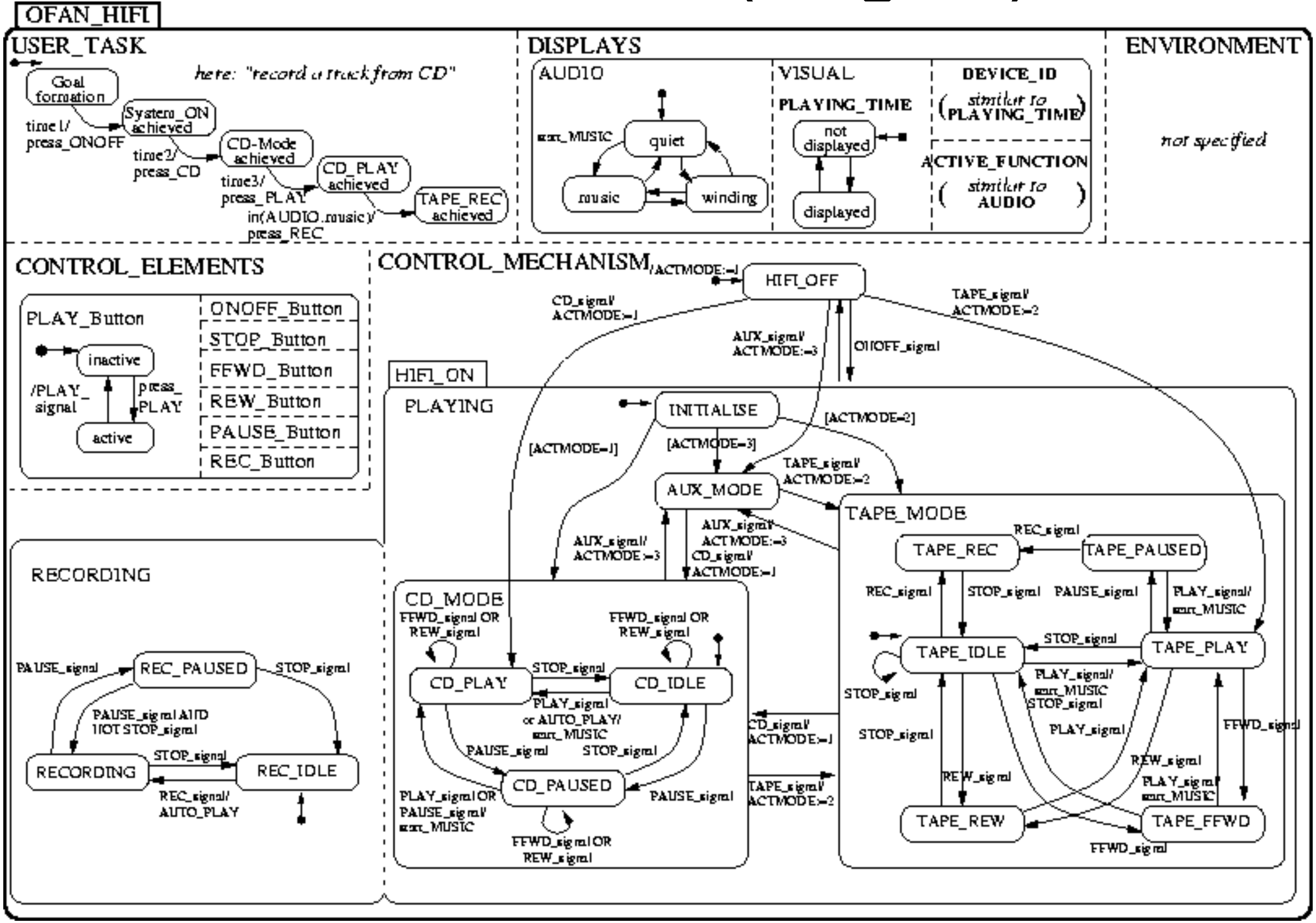
Graph model of infusion pump



Graph properties

- Path lengths – diameter of the model
- Concepts such as between-ness and centrality

Ofan model (Degani)



MAL interactors

- interactor main
  attributes
  [vis] auto, on, front, ac: boolean
  [vis] airintake: boolean # true: fresh / false: recirc
  automem, acmem, airintakemem: boolean
  [vis] settemp: Temp
  [vis] airflow: AirFlow
  airflowmem: AirFlow
  [vis] fanspeed: FanSpeed
  actions
  autokey off modekey fanspeedup fanspeeddown
  tempup tempdown frontkey ackey airintakekey
  axioms
  [autokey] auto' & on' & !front' & keep(airintake,settemp)
  [off] !auto'& !on' & fanspeed'=0 & !ac' &
  keep(airintake,settemp,front,airflow)

Dwyer et al

- Not to find the best abstractions for model checking analysis
- Rather seek abstractions that enable the analysis of naturally occurring specifications of GUI behaviour
- Interactive development of the model that
  - Uses the application's control flow based on Swing's event driven framework
  - Uses the windows, widgets on a window and text and colour associated with widgets on the windows
- Interaction ordering properties
  - When an error message is displayed the only available user action is acknowledgement via the OK button
  - Exploring *modal dialogues* which restrict the next user interaction to the enabled actions in that dialogue, all other actions are disabled
- Focus on *visible* actions and *enabled* actions and track containment relationships

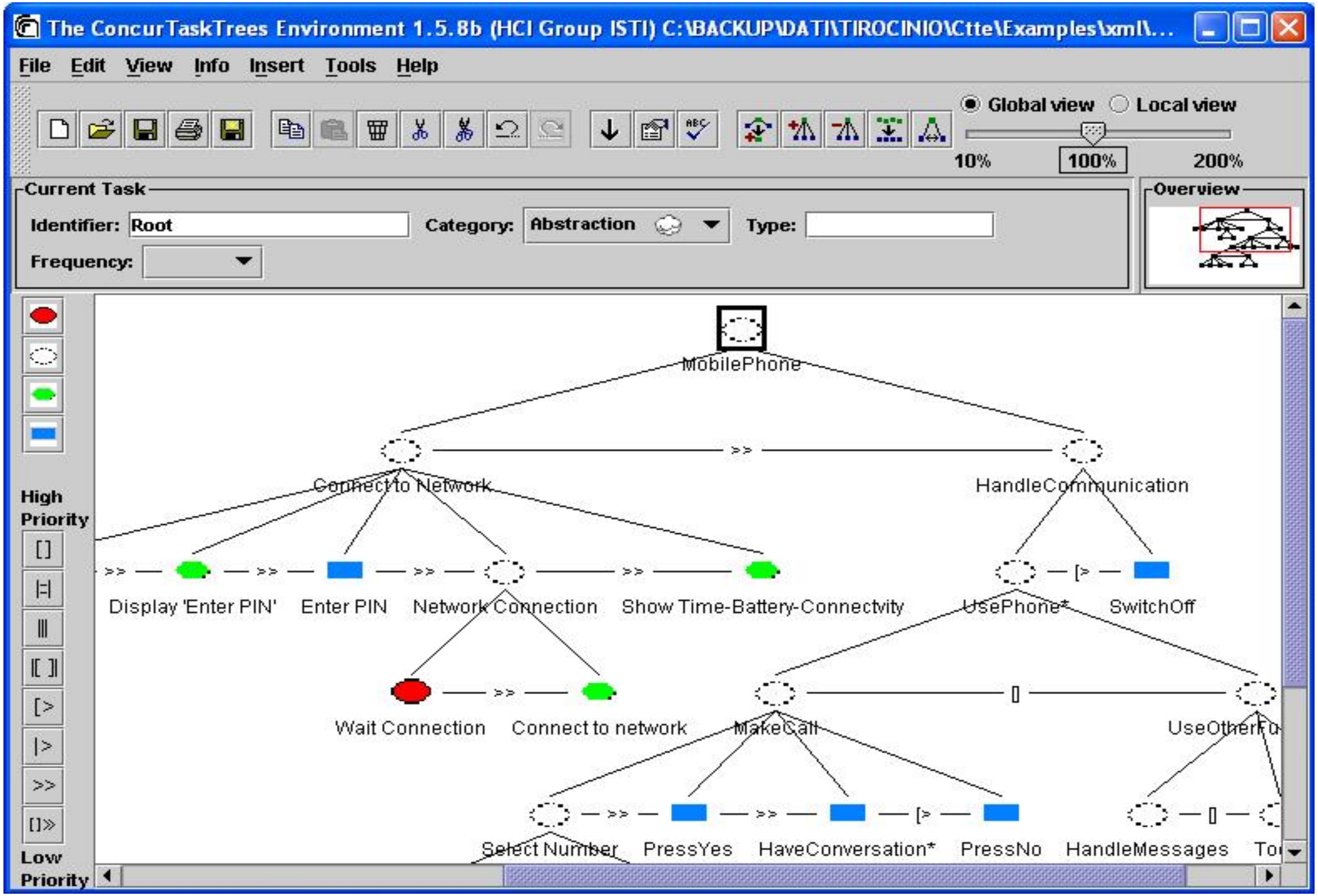
Modelling the user

- Modelling tasks
  - LOTOS -> CTT
  - Bob Fields using murphi as a basis for model checking (Fields, 2001)
- Modelling cognitive process
  - Syndetic modelling (Duke et al., 1998)
  - Curzon and Blandford work
  - (substantial literature on models of cognitive process in general)

Tasks

- what goals is the user meaning to achieve?
- what does the user need to know to achieve these goals?
- how does the user achieve these goals?
- what happens when goals aren't achieved?
- task representation is about the "observable behaviour" of users rather than their mental state.
- what are the plans, actions?
- what does the user have to perceive or interpret?

CTTE – ConcurTaskTrees Environment



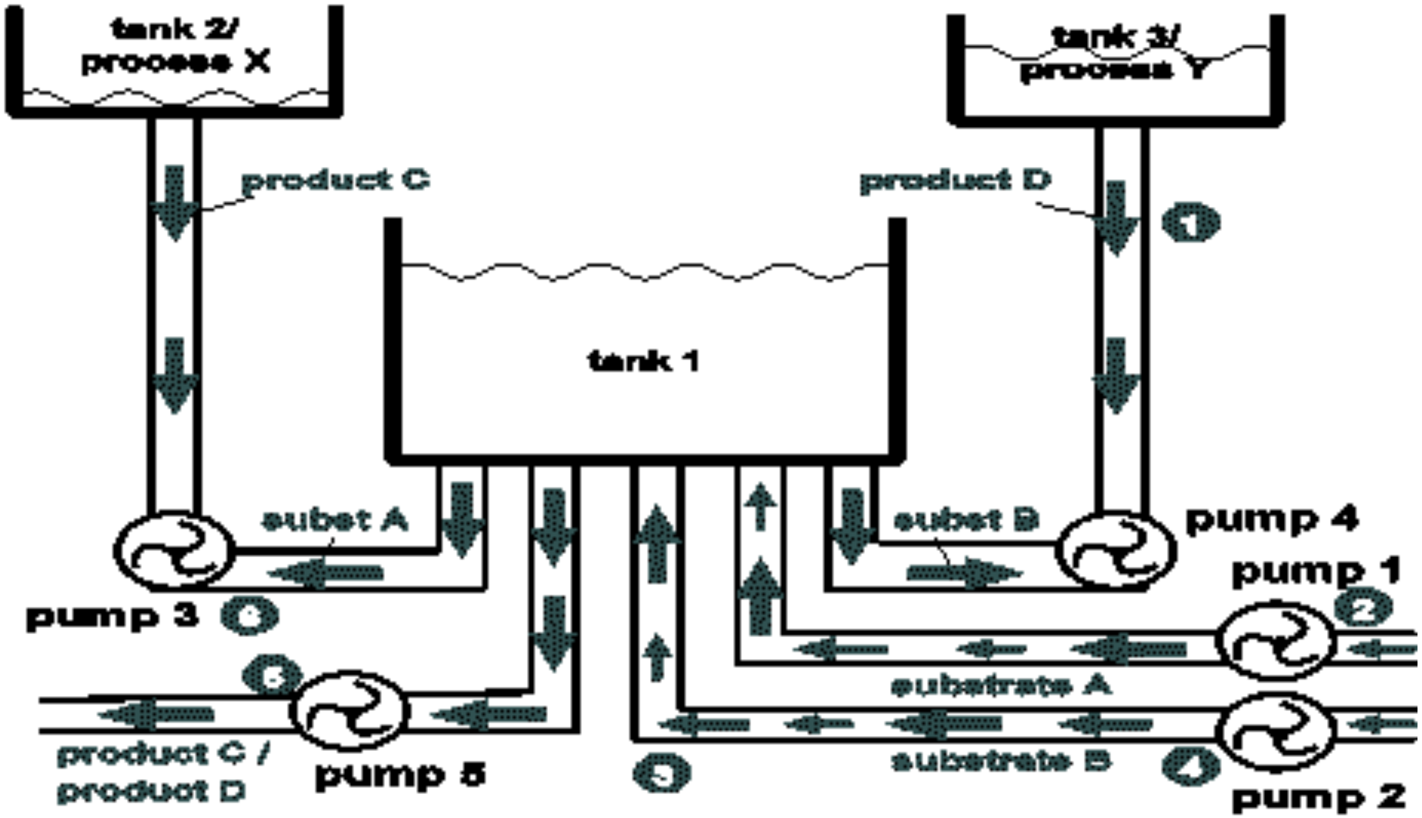
SAL Modelling of cognition (Ruksenas and Curzon)

- TRANSITION
  [](i:GoalRange): GoalCommit:
  gcommit[i] = ready ∧
  NOT(gcomm ∨ rcomm) ∧ finished = notf ∧
  goals[i].grd(in, mem, env) →
  gcommit′ [i] = committed;
  gcomm′ = TRUE
  []
  [](i:ReactRange): ReactCommit:
  rcommit[i] = ready ∧
  NOT(gcomm ∨ rcomm) ∧ finished = notf ∧
  react[i].grd(in, mem, env) →
  rcommit′ [i] = committed; rcomm′ = TRUE

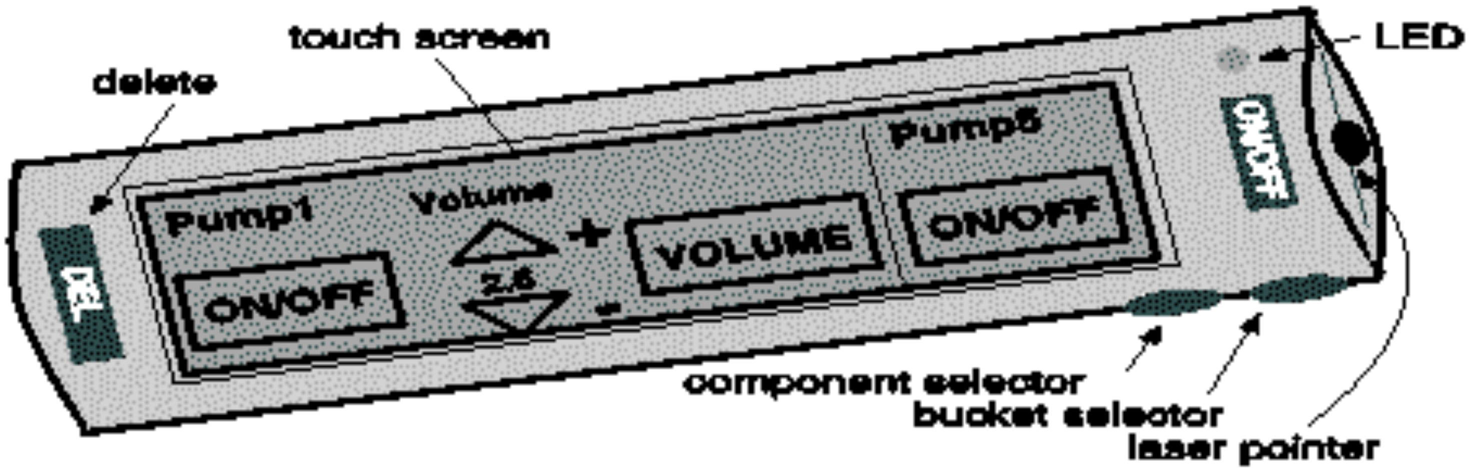
Modelling constraints on interaction

- Exploring paths in which actions are governed by constraints, explores more possible behaviours including workarounds
- Adding resources to specifications

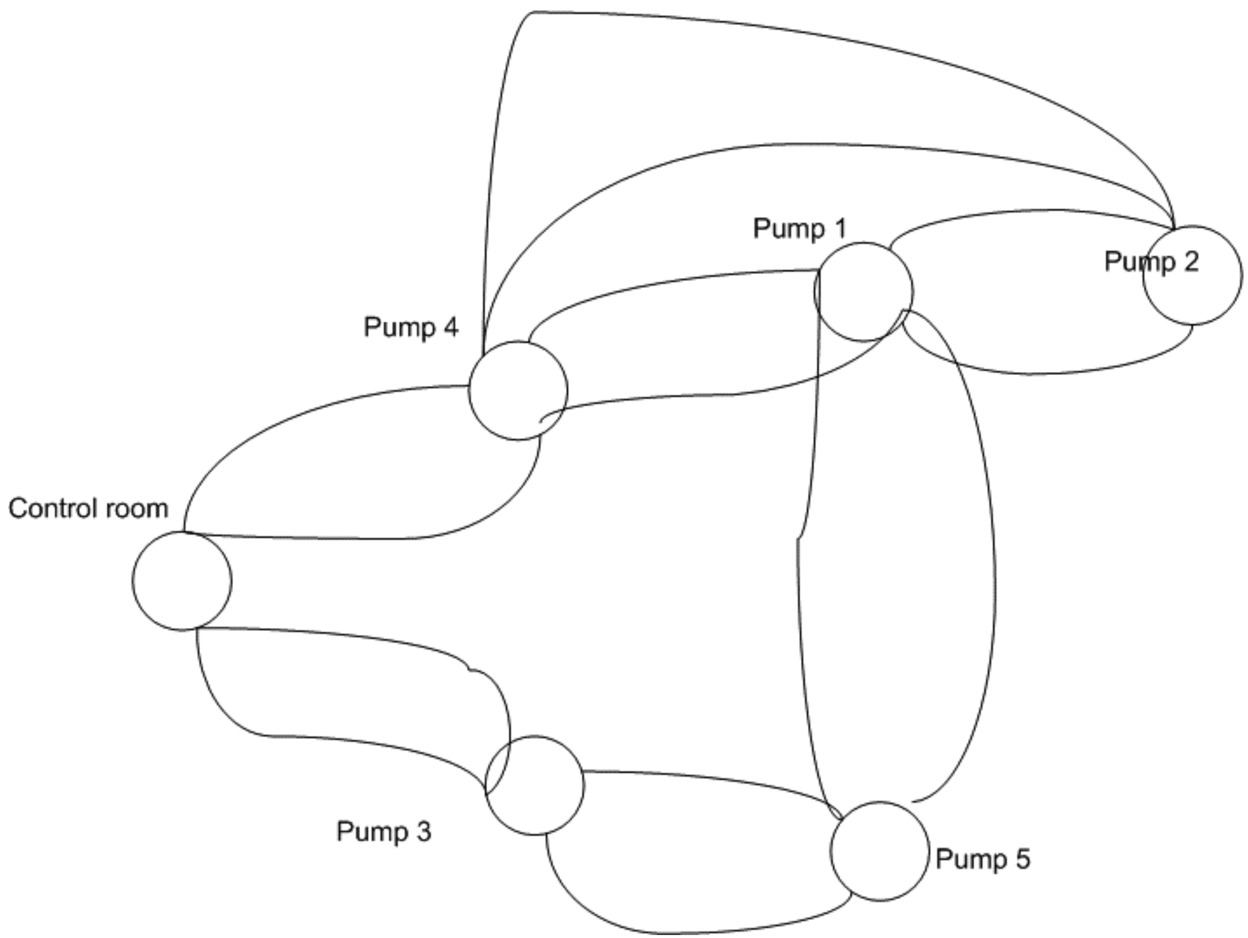
Process control environment



Controlled by a mobile device



Plant organisation



Plausible behaviour subject to resourcing

- Making the right information available to the right people at the right time
- designer must consider a range of issues:
  - support for user strategies,
  - making the most of available screen space,
  - avoiding information overload,
  - reconciling competing information requirements when the system supports a number of different activities.
- avoiding for example the “keyhole problem”
  - users required to traverse a large virtual perceptual field to gather together information resources which are accessible in different parts of the system

Resources

- status/visible information
  - for example the display indicates that a message is waiting
- action possibility
  - indicating that an action is available including indication that the possibility for action exists
- action effect information
  - letting the user know what the likely effect of an action will be
  - “Press ok to save”
- plan information
  - aiding the sequencing of user actions.
  - “You are in step 3 of 5”
- goal information
  - helping the user to formulate and keep track of multiple goals.
  - “there are new messages”

Resources

- Some of these resources are visible in the device, some may be dependent on operator knowledge
- resource[pumptype]: simple, volume or directional.
- resource[pmpstate]: on, off
- resource[pmpdirection]: backward, forward
- resourcevol: volume defined by the pump
- resourcescevol, resourcedstvol volume of material in the source and destination tanks
- position: where is the operator
- schedule: order of using pumps: 2, 4, 5

Defining the relation

```
::  (resource[pumptype]==simple) &&
        (resource[pmpstate]==off) &&
        (resourcescevol==empty) ->
              { complete=true; next++; assert(false)}
```

Analysis

- Use of tools to analyse
  - Possible interface behaviours
    - considering users (varying levels of detail)
    - not considering users
  - Structural (Topological?) properties of the models
    - Graph properties
- Analysing
  - Heuristics (typically qualitative)
    - Need to translate to checkable properties
  - Metrics (typically qualitative)
    - Need to establish metrics' relevance
- Relevance of Simulation

The role of a property

- Capture the way that the device is used
- Example
  - Every action should have a visible effect
    - It is not enough that an action has a visible effect, it is important that the effect is visible to the user in the context of a typical activity
  - Every action’s effect should be predictable by the user and this effect cannot be changed by the environment without the user recognising its implications

Systematic analysis

- Plausibility analysis
  - AG(auto -> on)
- Application of property patterns
  - Feedback
  - Behavioural consistency
  - Undo
  - Reversibility
  - Reachability
  - Eccentricity
  - Completeness

Feedback

$AG(airflow = x \rightarrow AX_{modekey}(airflow \neq x))$

note use of shorthand $AX_{modekey}p$ becomes AX(modekey->p) in IVY tool

$AG(fanspeed = x \rightarrow AX_{fanspeedup}(fanspeed \neq x))$

– Fails when fan speed at maximum (10) and the button does not change speed

Property Pattern: ***Feedback***
Intent : To verify that a given action provides feedback.
Formulation : $AG(pred(s) \wedge c =_* x \rightarrow AX_a\, (c \neq_* x))$ Under the defined condition (*pred*), the action (*a*) will always cause a change in some perceivable a ttribute (in *c*).

Future work

- Scaling the analysis
- Dealing with ubiquitous systems

Scaling

- Scale and Genericity
  - Off the shelf models that can be instantiated to particular requirements
  - Batteries of properties that can be used as part of an integrated analysis
  - Addressing different types of systems and user related concerns
- Finding interesting/relevant examples
  - Finding good examples for analysis is hard
  - Repository of analyses of real systems to establish confidence
  - Connecting with industry
- Usable tool support
  - "You cannot promote usability with unusable methods"
  - Developing tool support for all stages of analysis
- Lack of a common language?
  - Do we need one?
  - "Speak the users language"

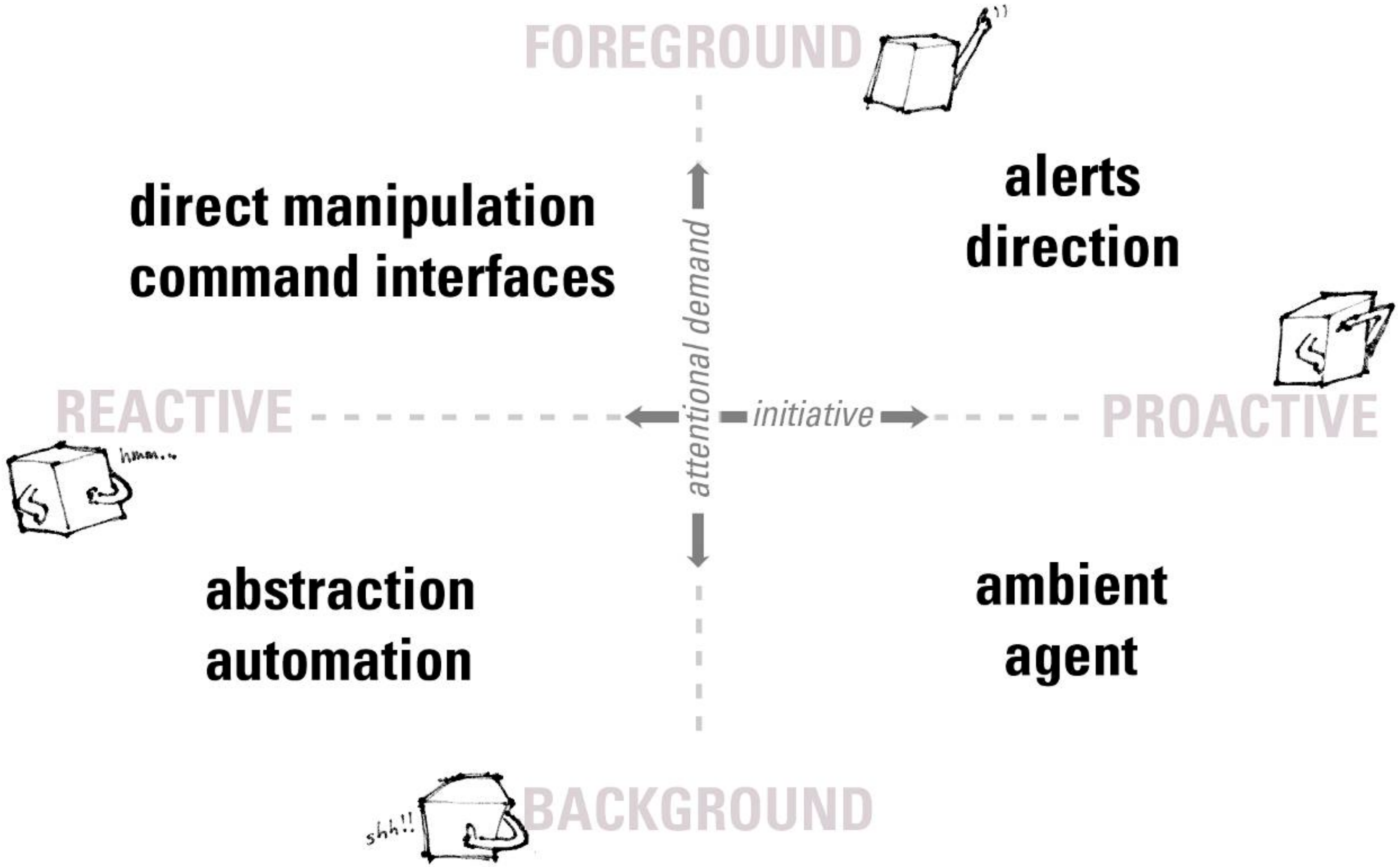
Big Challenges

- Technology transfer
  - There seems to be interest, but we need…
  - Scalable methods
  - Tool support
  - Integration with existing practices
- Build common ground
  - Repositories (Models / Properties / Analyses)
  - Common language?
  - Benchmark systems?
- Extend scope of analysis
  - to better deal with highly dynamic systems
  - to better deal with user experience different user related concerns
  - To reverse engineer and create models for implemented systems automatically

Future work

- Scaling the analysis
- Dealing with ubiquitous systems
  - Modelling the system as a whole
  - Exploring quantitative properties
  - What is experience?

Implicit Interaction (ubiquitous systems)



Range: Exploring Implicit Interaction through Electronic Whiteboard Design: Ju Lee and Klemmer CSCW: 08

Particular class of ubiquitous systems

- Designed to enhance physical environments through
  - Sensors
  - Public displays
  - Access to personal mobile devices if they are available
- Examples
  - Airports
  - Hospitals
  - Shopping centres

These systems can be:

- Designed to support experience
- Why?
  - User immersed within the environment
  - Can create a sense of place in an otherwise forbidding space
  - Can provide information that is relevant to the circumstance, anticipating the concerns of the user
- Hence experience is a primary focus for ubiquitous system research

Experience Properties (1)

- Visitors (office building, stadium):
  - How to get to a specific office, or seat within the stadium.
  - Tailored timely and relevant information
  - Avoid confusion or frustration
  - Avoid uncertainty and feeling of being lost
- Evacuating an office building, or stadium:
  - clear and calm instructions at each stage of their exit to ensure they take optimal routes that offer fastest, shortest, safest way out.

Experience Properties (2)

- Airline passengers:
  - Tailored information about queue to join for check-in, baggage screening, passport control etc.
  - Reduce waiting, improve sense of the airport as a place rather than a forbidding space.
- Hospital out-patients:
  - Directions at each stage of the appointment
  - Potential for rescheduling the different stages
- Newly registering college students:
  - directing to get a library card, pay fees

Challenge

- To develop functional models of a class of ubiquitous system that can be used for both qualitative and quantitative reasoning
- To use these models as “evidence” of the suitability of a particular system design before deployment in terms of the experience requirements seen to be relevant
- A problem with these systems is they are highly dependent on the texture and context of the actual physical environment

Quantitative model

- Example properties:
  - How long to wait before display relates to the user who has just entered?
  - Will a specific algorithm lead to congestion?
- These issues relate to the experience of people within the system
- Analysis for design
  - How to configure the display, how to define appropriate paths

The model

- Preliminary, rather abstract, exploration in relation to ubiquitous systems and HCI
- Used PEPA
  - Because of potential for qualitative and quantitative analysis
  - More detailed functional analysis using SPIN
- Analysis
  - Stochastic model checking
  - ODE-analysis
  - Checked results of ODE analysis using simulation

Our concerns

- Scalability of model
- Properties that can be addressed
- Required abstractions
- Limitations and potentialities
- We just started to explore this, but first results are promising. Larger buildings have been modelled and analysed as well.
- Work is ongoing

Questions

- What does usability mean where interaction is implicit and crowd factors are important?
- Can crowd usability be analysed using quantitative and qualitative models?
- Can these models provide the backing to the deployment of particular ubiquitous systems?
- Can techniques such as described here be scaled to realistic systems?
- Can these techniques be made accessible to usability engineers?

References (system modelling)

- Barboni, E., Navarre, D., Palanque, P., Basnyat., S. 2007. A Formal Description Technique for Interactive Cockpit Applications Compliant with ARINC Specification 661. SIES 2007: 250-257.
- Thimbleby, H. and Gow, J. 2008. Applying Graph Theory to Interaction Design. EIS 2007: 501-519.
- M.B. Dwyer, O. Tkachuk, and W. Visser. Analyzing interaction orderings with model checking. In Proceedings of ASE'04, pages 244–261, 2004.
- Campos, J.C. and Harrison, M.D. 2008. Systematic analysis of control panel interfaces using formal tools. DSV-IS 2008: 72-85.
- Berstel, J., Crespi-Reghizzi, S., Roussel, G., San Pietro, P. 2005. A scalable formal method for design and automatic checking of user interfaces. ACM Trans. Softw. Eng. Methodol. 14(2): 124-167A. Degani (2004) Taming HAL Palgrave MacMillan 2001

References (users, tasks, resources)

- Paternò, F. 1999 Model-Based Design and Evaluation of Interactive Applications. Springer-Verlag.
- Fields, R. 2001. Analysis of Erroneous Actions in the Design of Critical Systems. DPhil thesis. University of York.
- Duke, D.J., Barnard, P.J., Duce, D.A., May, J. 1998. Syndetic Modelling. Human-Computer Interaction, 13(4:337-393.
- Blandford, A., Barnard, P.J., Harrison, M.D. 1995. Using interaction framework to guide the design of interactive systems. International journal of human-computer studies, 43(1):101-130.
- Wright, P.C., Fields, R.E. and Harrison, M.D. (2000) Analyzing Human-Computer Interaction as Distributed Cognition: the resources model. *Human Computer Interaction.* 15(1) pp. 1-42.
- Doherty, G., Campos, J.C. & Harrison, M.D. (2008) Resources for Situated Actions. In T.C. Nicholas Graham and Philippe Palanque (editors) Interactive Systems: Design, Specification and Verification (DSVIS 2008) Springer Lecture Notes in Computer Science 5136 pp. 194-207. Link to UPPAAL model.

References (smart environments)

- Harrison, M.D., Massink, M. and Latella, D. (2009) Engineering Crowd Interaction within Smart Environments, editors, G. Calvary, T.C.N. Graham and P. Gray, Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, ACM Press, pp. 117-122.