Human-centric (Self-Adaptive) Software Engineering

ARC Laureate Professor John Grundy

SEAMS Keynote

May 2023

https://www.monash.edu/it/humanise-lab

MONASH University

Australian Government
Australian Research Council

HUMANISE

Acknowledgement of Country

As we gather for this meeting physically dispersed and virtually constructed let us take a moment to reflect the meaning of place and doing so recognise the various traditional lands on which we do our business today.

We acknowledge the Elders – past, present and emerging of all the land we work and live on and their Ancestral Spirits with gratitude and respect.

I acknowledge the people of the Kulin nations, the traditional owners of the land on which I am meeting with you from today.

Outline

End User Diverse Humans and software

Motivating Example (lacking adaptation to diverse users…)

Our early adaptive UIs work

Our early adaptive collaboration work

Our more recent adaptive UIs work

Our more recent end user adaptive work

Future directions

One size fits all in Software Engineering

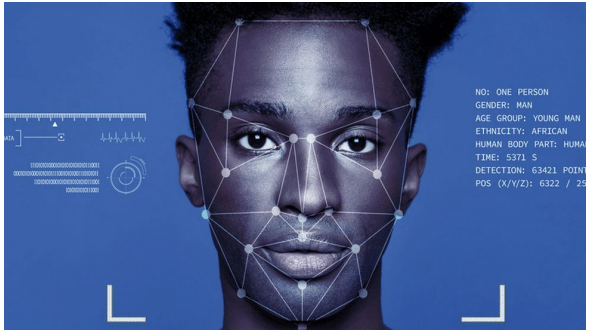
Gender bias – UIs, seat belts, health apps

Ethnic bias – over-recommend minorities for search, don't recognize faces, mis-classify

Culture bias – inappropriate words, phrases, colours, icons, workflow

Language bias – over-technical, wrong dialect, impersonal

Age bias – too complex, too simple, inappropriate words, symbols, workflow



More one-size-fits-all…

Physical challenge bias – guesture, sound, sight, voice inappropriate

Cognitive challenge bias – raise anxiety, poor fit to mental model, doesn't support neuro-atypical

Enjoyment bias – boring, unengaging, distracting

Emotional bias – stressful, anxiety-inducing, frightening

Personality bias – workflow, lack of engagement, disconnected

And… many others :-(

All Can Apply to TEAM and USERS!!!

Developers as (diverse) humans…

NOT the focus of this talk - but we have a bunch of projects on too :-)

BUT - developers usually VERY different to their stakeholders and software end users:

-high education level ; high use of jargon ; mostly male

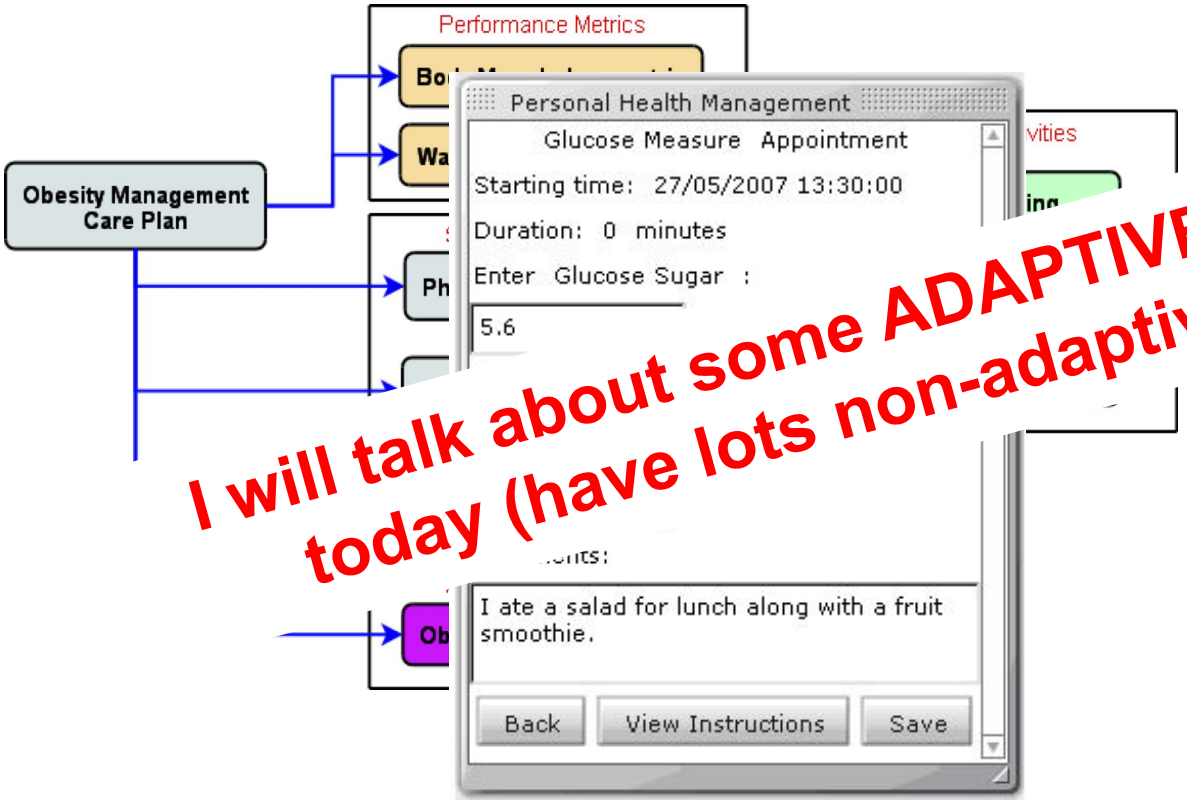
-mostly highly English-proficient; mostly 20s and 30s; high socio-economic group

-interestingly much higher proportion are neuroatypical than general population…

Developers struggle to understand end users different to themselves

DEVELOPERS BUILD ONE-SIZE-FITS-ALL - THEIR SIZE :-) - SOLUTIONS…

Motivating Example…



Model-driven, end user approach

Clinician mod… care plan… for app

I will talk about some ADAPTIVE solutions today (have lots non-adaptive too…)

Fails to take account of ageing patient, gender, culture, language proficiency, terminology, accessibility issues, ...

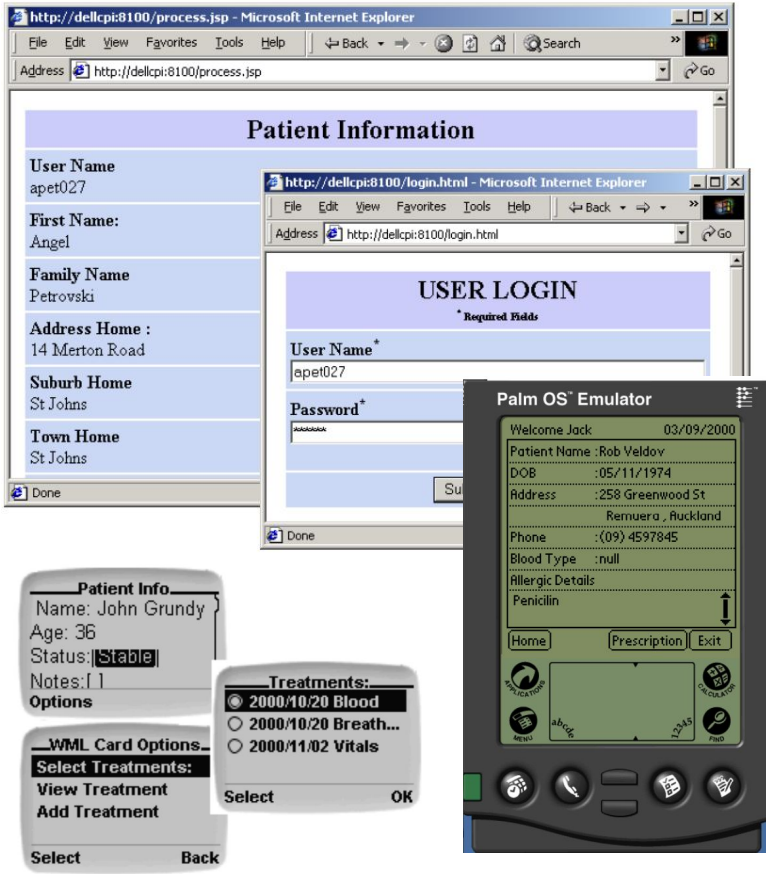
Problem 1 - people use different devices

We became interested in the issue of supporting people accessing web pages across multiple devices around 1999-2000 when mobile devices with screens first appeared (in NZ anyway)

We built several web and mobile interfaces to access same system

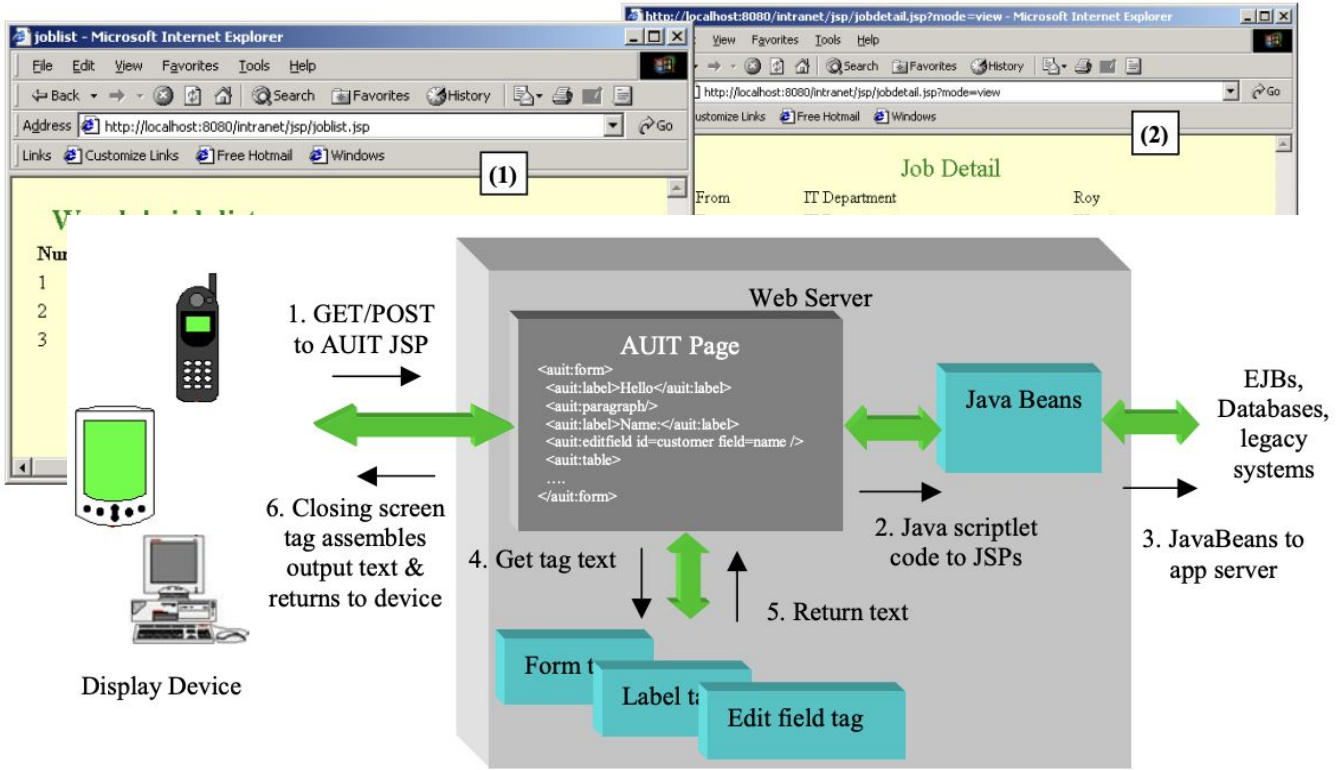
Lots of similarities - but lots of differences too…

Question - can we run-time adapt a single application to serve UIs for diverse devices - addresses end user issues of device type, screen size, mobility, …



Solution - multi-device adaptation @ run-time

Adaptive User Interface Technology (AUIT)
Augmented Java Server Pages with various tags
Run time pre-processing to generate device-specific UI
Served to different devices (HTTP, WAP, etc)
Can do size, layout, colour, complexity adaptations…



Problem 2 - people want to collaborate

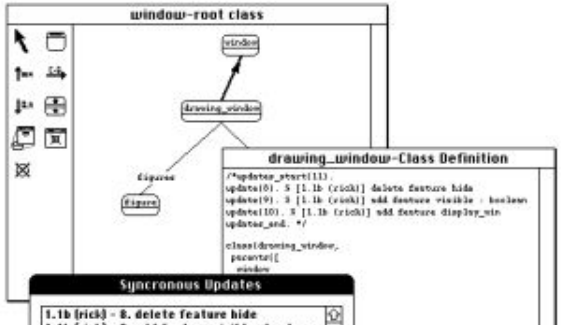
We had been building modelling tools for many years (since 1990…!!)

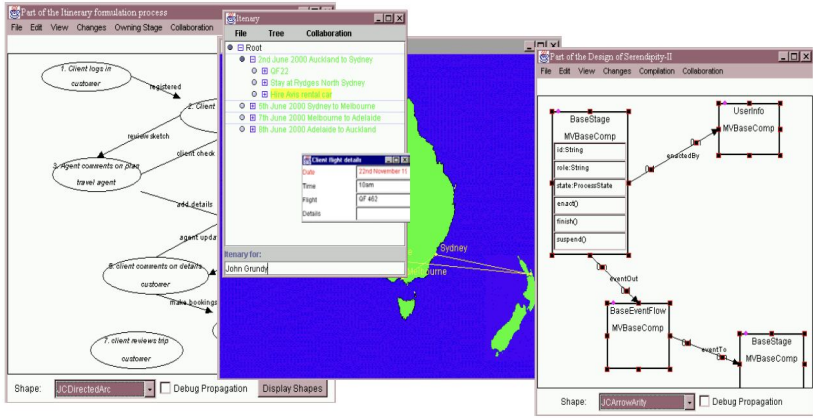
People want to collaboratively model

We custom-built collaborative modelling support for each tool

People also want to e.g. collaborative travel plan, find and discuss information, document edit, present, …

Can we use the same set of collaborative work components across multiple, diverse applications?

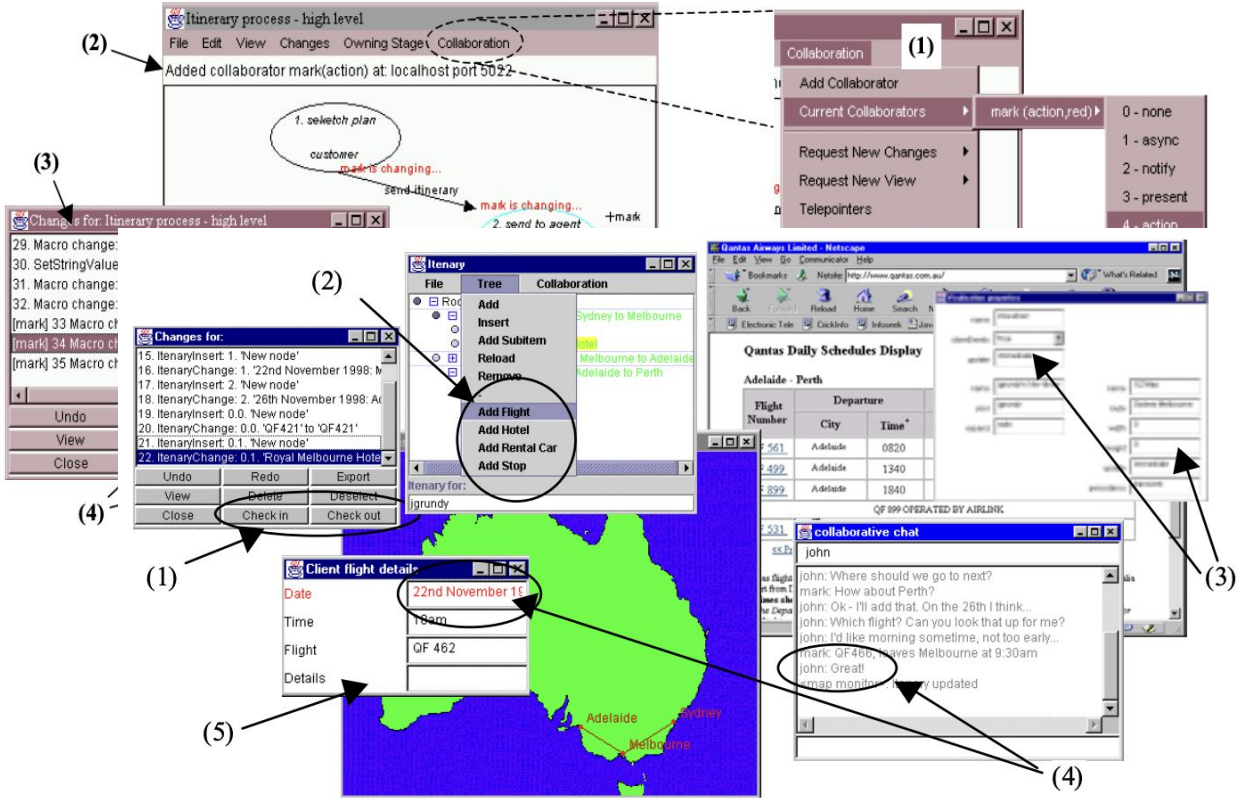




(a) Workflow tool (b) Travel planning tool (c) CASE tool

Solution 2 - (run time) plug-in adaptive components

- Developed a set of plug-ins can be run-time added to tools, web apps
- Gives range of collaborative work (and adaptive UI) facilities
- Dynamically adds various collaboration capabilities @ run time
- Some of these are user-configurable at run-time



https://www.monash.edu/it/humanise-lab

Problem 3 - hard to do run-time plug-ins

Nice, but…

Plug-ins have to be implemented to “know” about application UI and framework adaptation points

Applications have to be implemented to support these extensions

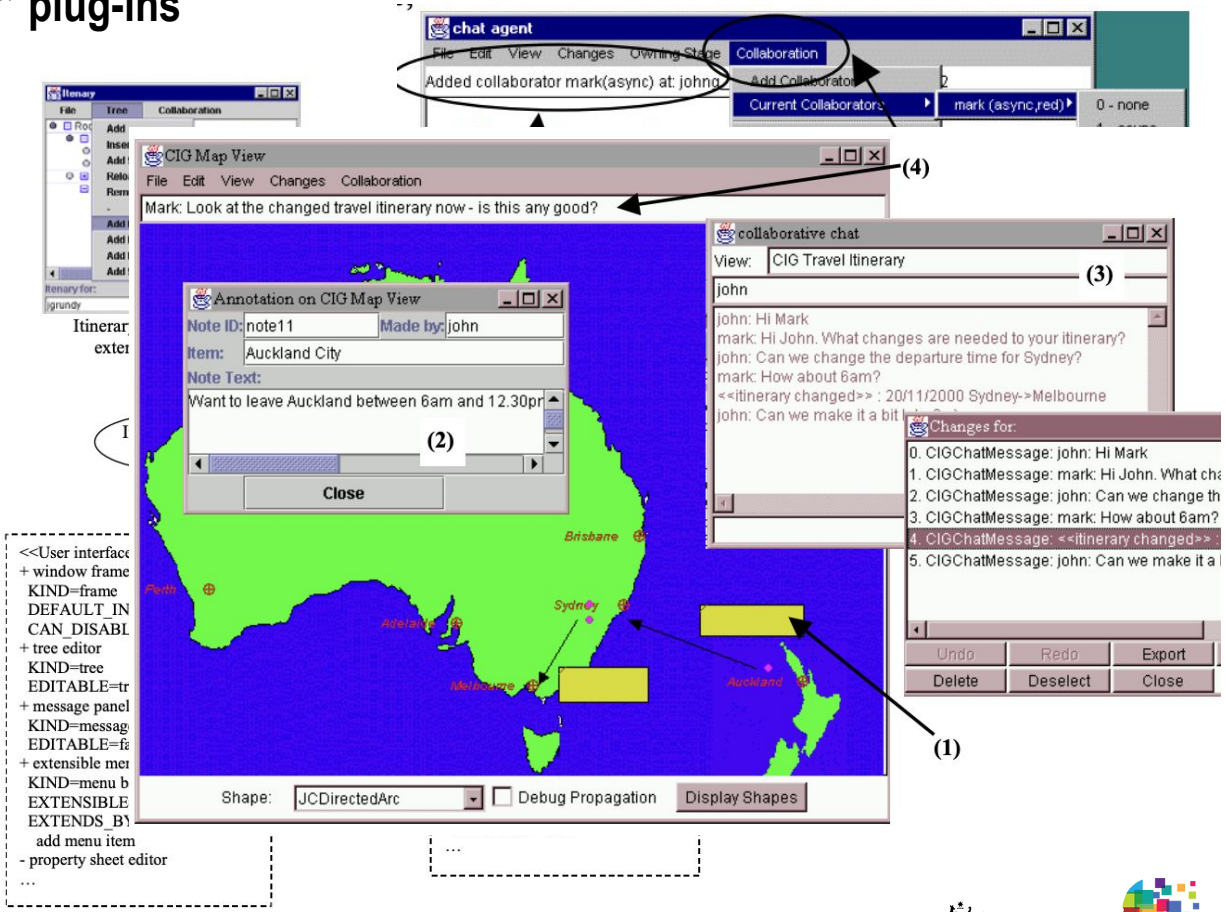
Applications have to “know” about possible plug-ins

Plug-ins need a lot of knowledge about application data management, communications, components, UI elements…

Can we abstract away from this to support “emergent” adaptive applications?

Solution 3 - “aspect-oriented” plug-ins

- “Aspects” advertising provides/required capabilities
- All application components advertise
- All UI/collaborative work plug-ins look for & use
- Fully run-time configurable



https://www.monash.edu/it/humanise-lab

Problem 4 - people want “human-centred” modelling tools

Continuing our modelling tool work, we wanted to support (generically) a variety of “human centric” modelling tool features

-sketch-based input of diagrams

-collaborative editing, model sharing, version control

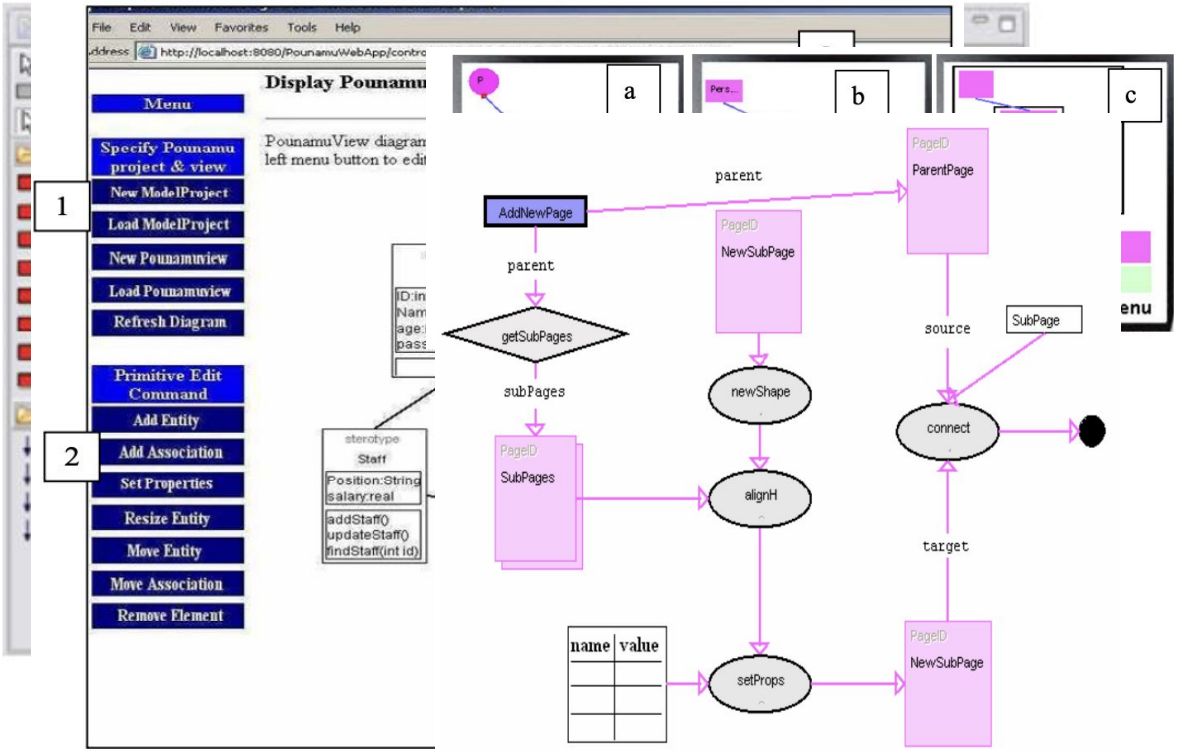
-multiple device - desktop, web, mobile - editing

-per-user modelling constraints

Wanted to do this via plug-in, run-time configurable way…

Solution 4 - plug-in sketch, web/mobile, collaborate, reconfigure constraints components…

- Plug-in sketching components
- Plug-in web, mobile editing components
- Plug-in collaborative editing components
- Plug-in sharing components
- Plug-in UI adaptation, architecture adaptation
- Extensible constraints



Problem 5 - users want their own e.g. custom data visualisations

Developers are different to (most) of their end users

Developers take a long time to make needed changes (if at all)

End users are domain experts

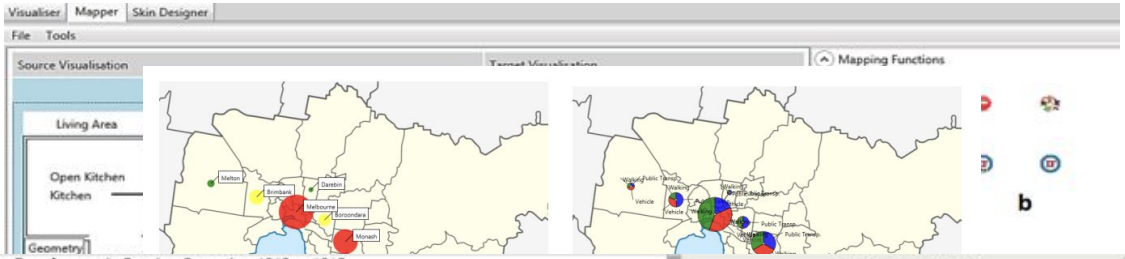
Example: visualising complex data

Most end user tools have various limitations/constraints

Solution 5 - let them build, generate their own

CONVeRT end user development tool

("no code" env)



https://www.monash.edu/it/humanise-lab

Problem 6 - Users have different usability challenges

Users are different

Users have different accessibility, inclusivity, device, location, cultural, language, … needs

Once size fits all software doesn't help…

Device-level adaptations e.g. font size, zooming, colour maps too crude to assist a lot of the time

Ideally want screen element-level adaptations

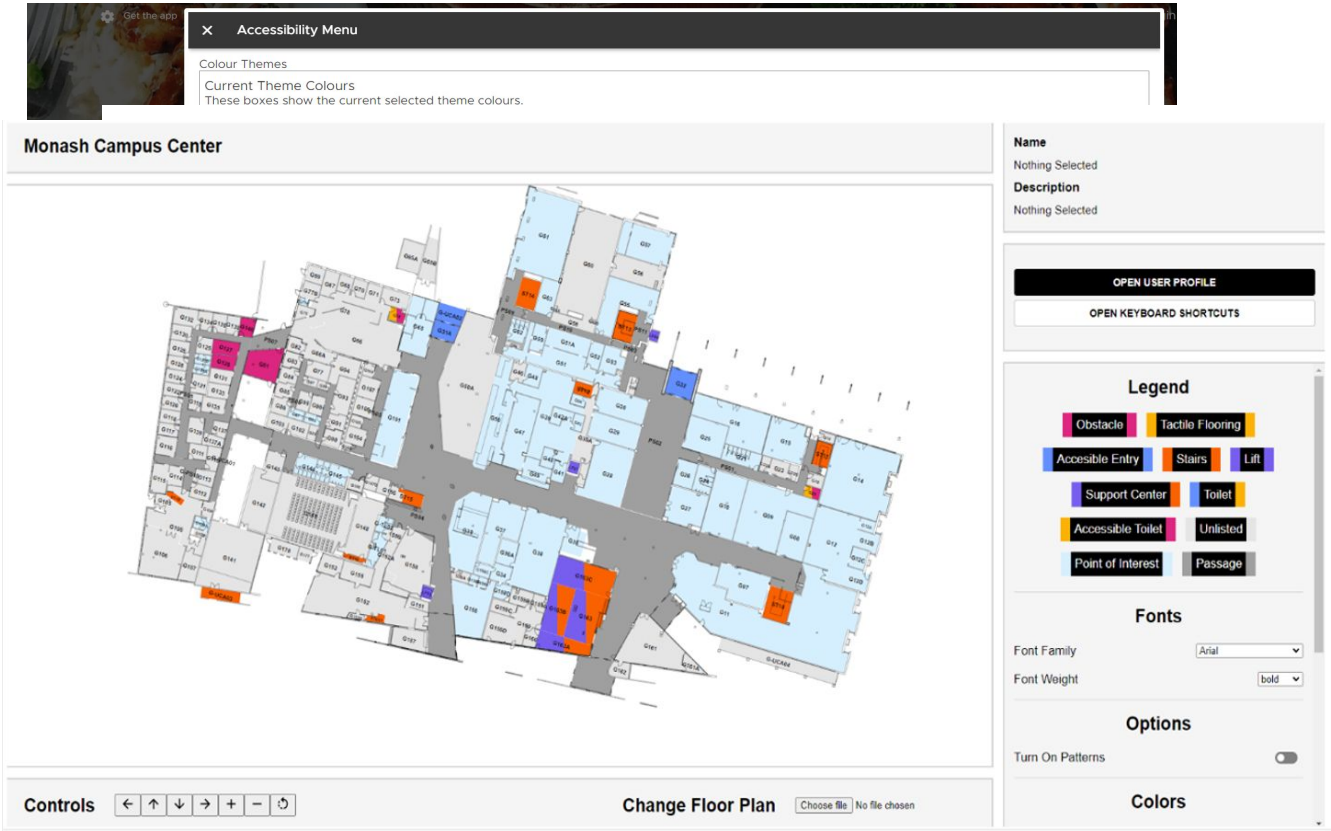
Ideally want run-time, user-controlled, AI-assisted adaptation/adaptive software…

Solution 6 - give them adaptive UIs

We built bunch of Adaptive React and Flutter components

Like the earlier AUIT ones for devices - but focused on accessibility adaptations

Adaptive SVG for floor plans



Problem 7 - users have different perceptions of software problems

End users are different

“Defects are in the eye of the beholder”

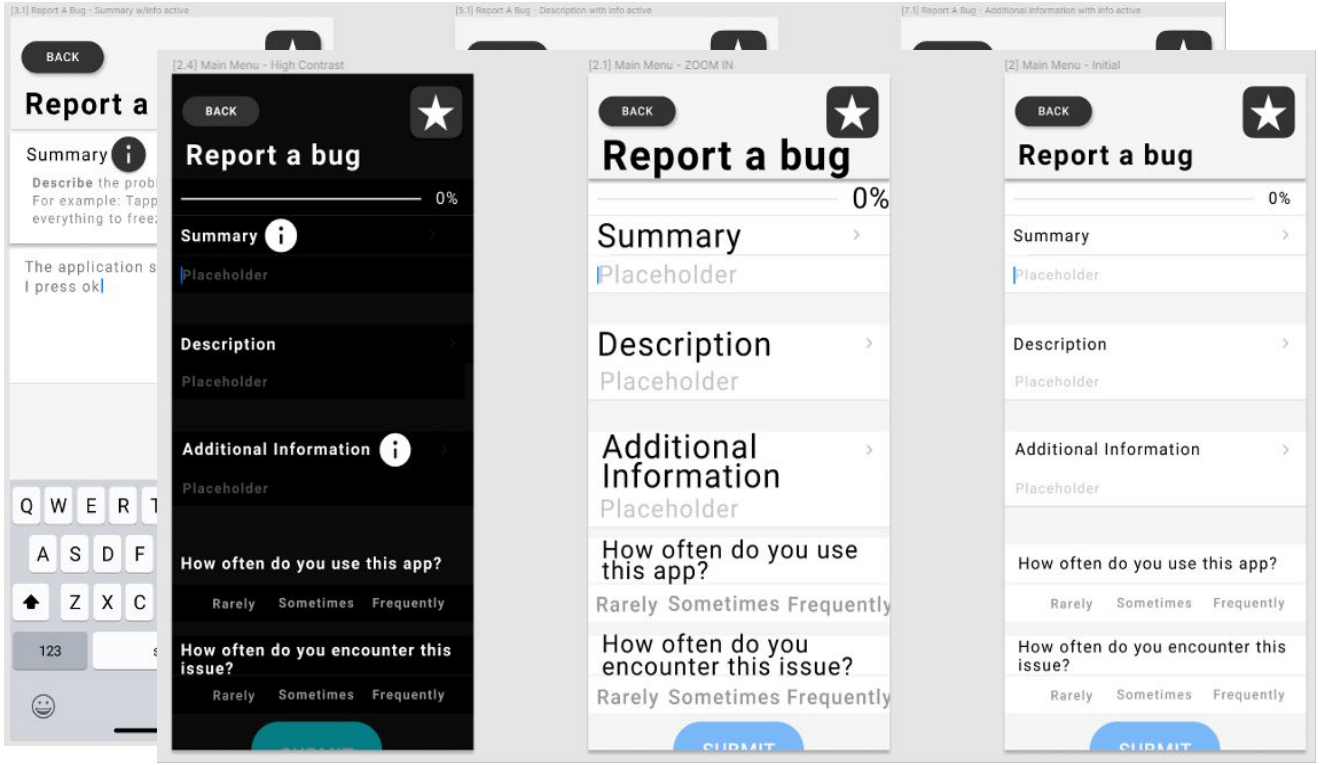
Some “defects” severely impact some end users, not a defect at all for others

Developers different to many of their end users ; find it hard to appreciate, understand, fix some such defects

Defect reporting tools themselves are… not very human-centric :-)

Solution 7 - give them “human-centred” defect reporting

- “Human-Centric defect reporting”
- Different users report defects THEY perceive
- Guided reporting form
- Adaptive reporting form
- (its human-centred reporting itself!)



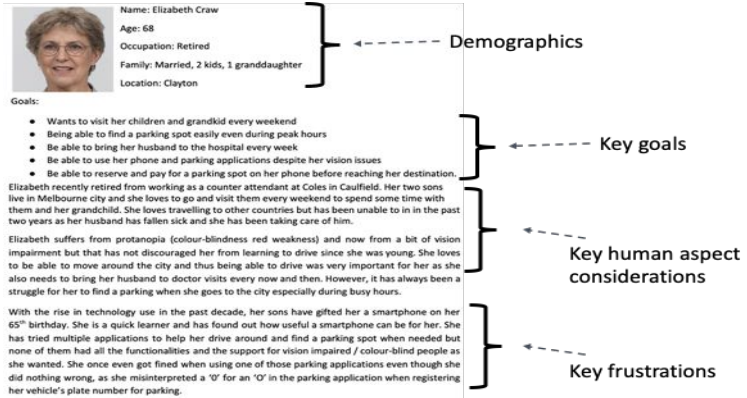
https://www.monash.edu/it/humanise-lab

Problem 8 - users want multiple-size-fits-them solutions

Different users have very different needs

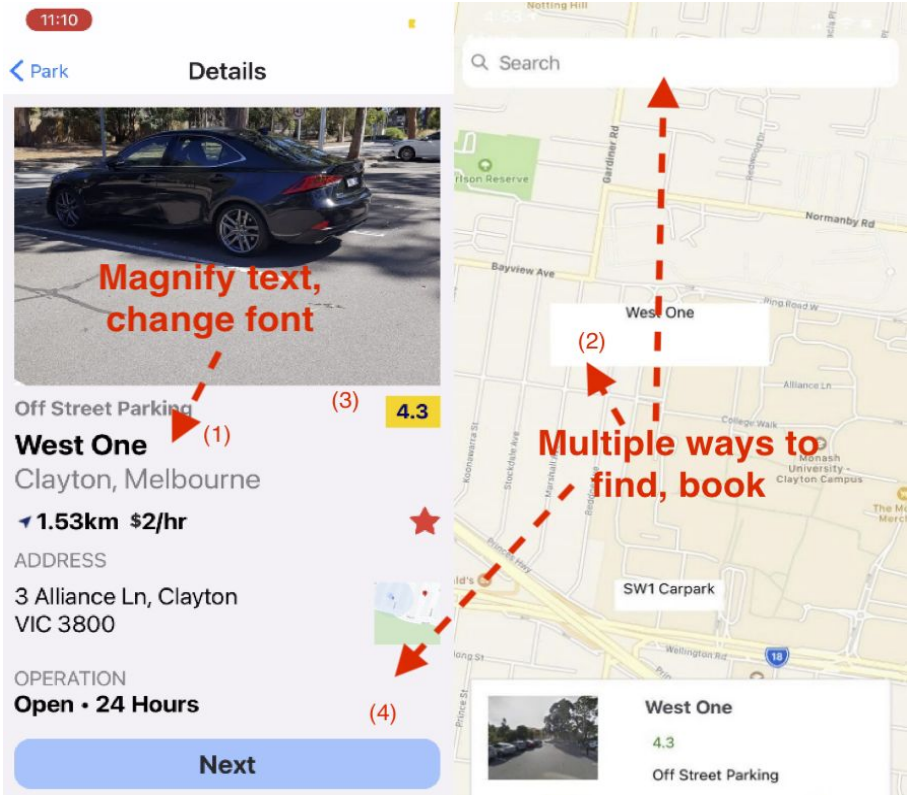
E.g. ‘smart parking’ - consider retired person vs person with multiple jobs vs person with young children for care drop off/pick up vs salesperson vs …

Almost all parking apps we reviewed support single way of find/pay for car parks…



Solution 8 - give them more “human-centred” developed solutions

- Model diverse users with rich personas
- Model super-set of user stories
- Model set of accessibility etc constraints
- Implement parking app that allows (i) multiple ways to find, book, pay, etc parks and (ii) accessibility, language, culture etc adaptations
- All at run-time, user profiles



Problem 9 - users want smart buildings BUT with informed consent re: data

Smart buildings becoming very common, popular

Wide range of ‘smart’ devices capturing all maner of data e.g. doors, lights, heating, AV, …

But - what data is captured? What done with it?

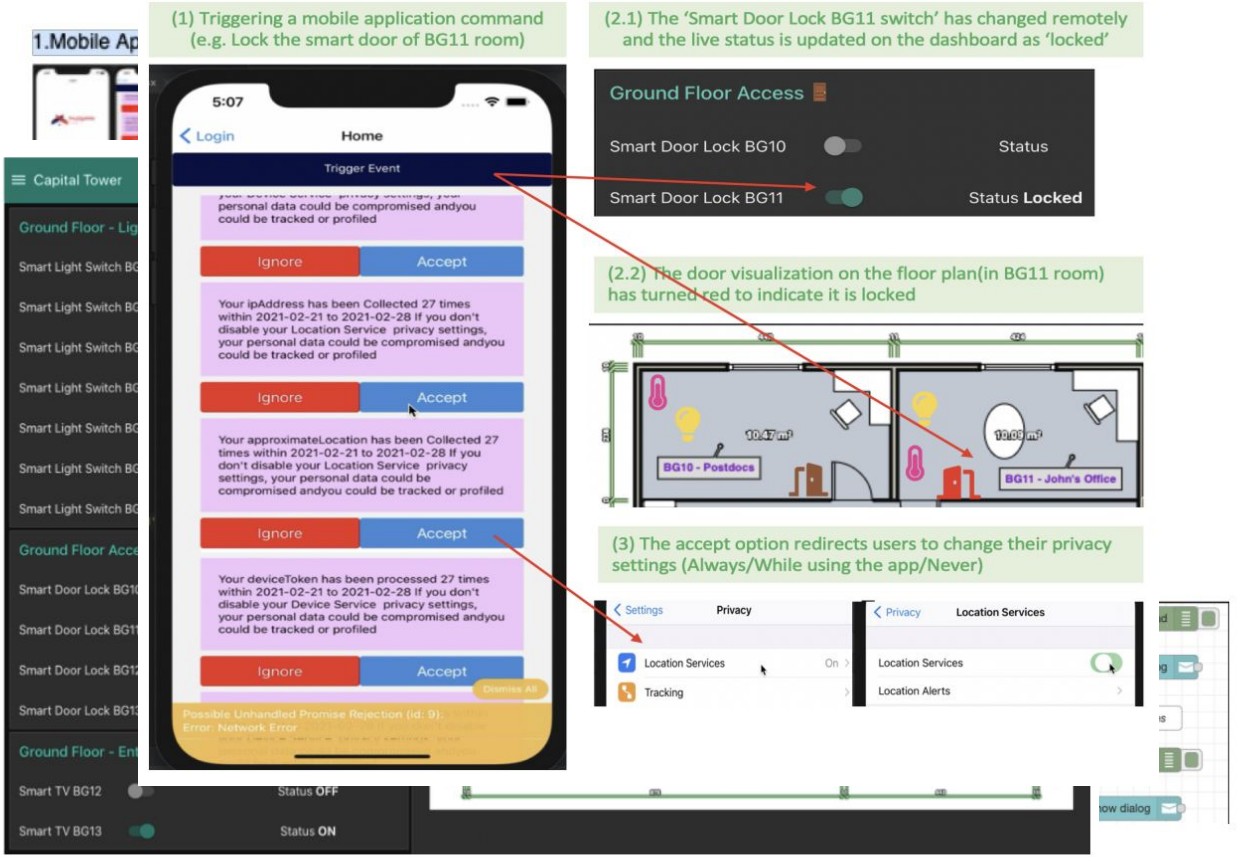
Users have little/no control over…

Solution 9 - let them specify

Model-driven informed consent system

Uses “nudges” to tell users when about to capture data, permission to capture, store, use…

My HumaniSE lab used as case study example… :-)



Problem 10 - human-centric application adapting / fixing is hard

Doing a lot of these run-time adaptations is hard… :-)

Want to

-add adaptivity to range of applications

-for range of diverse end user challenges

-automatically adapt to user profile/needs/preferences where can

-automatically ask if they need help

-use model-driven approaches to ease engineering

-get better feedback to developers

Solution 10 - Better RE/MDE/AI/ML to assist more human-centred, self-adaptive software…

Some of our current work:

-RE for ageing users via co-design, living labs - what do diff users need?

-Exploring MDE for adaptive UIs - ageing users, users with chronic disease

-Generating personas, using personas to automatically adapt UIs

-Detecting user difficulty with UI/application ; automatically adapting to try and help ; use problem solving dialogue with app to try and help

-Dialogue-driven adaptation of app (ChatGPT-style :-))

-Automated detection of “values violations” in apps to adaptively fix

-Automated internationalisation of apps - multi-language support

-Automated “inclusivity” adaptations of apps - UI and workflow

-Automated detection of e.g. “bad” ads in apps and automated block

-Automated detection of platform compatibility problems and patching

Some remaining challenges

How support the (very) wide range of diverse users - can it even be done in a single “adaptive” application

What about when addressing one set of user’s needs/requirements conflicts with another - need multiple apps after all??

How effectively model / generate (very) wide set of potential adaptations

Enabling AI-based adaptation - if have dialogue, how ‘know’ the underlying model that is built up (especially when it goes wrong)

Helping developers understand, appreciate feedback and address

Values-driven software engineering - fairness, privacy, honesty, transparency, etc - how ensure trustworthy software if can auto-adapt in ways end users (and developers!) can’t track, understand

Summary

End users are different (and often different to most developers!)

Adaptive applications can help ensure address diverse end user needs, challenges, preferences vs one-size-fits-all apps

Lots of human-centric adaptations are needed, some harder than others

Emergent requirements/unexpected end user needs/challenges are particularly hard to address

Co-design, Model-driven engineering, adaptive architectures/UIs useful

TBD how AI/ML-based adaptation support can help (or hinder) e.g. *“Make my UI meet my accessibility needs, add collaboration support, …”*

There are many unsolved challenges in this area to work on :-)

References

Pathmabandu, C., Grundy, J.C., Chhetri, M.B., Baig, Z. Privacy for IoT: Informed consent management in Smart Buildings, Future Generation of Computing Systems, vol. 145, August 2023, pp 367-383, Springer

Liu, P., Xia, Q., Liu, K., Guo, J., Wang, X., Liu, J., Grundy, J.C., Li, L. Towards Automated Android App Internationalisation - An Exploratory Study, Journal of Systems and Software, Volume 197, March 2023

Zhao, Y., Li, L., Liu, K., Grundy, J.C. Towards Automatically Repairing Compatibility Issues in Published Android Apps, 44th IEEE/ACM International Conference on Software Engineering, Pittsburgh, USA, 21-29 May 2022

Luy, C., Law, J., Ho, L., Matheson, R., Cai, T., Madugalla, A., Grundy, J.C., A Toolkit for Building Adaptive User Interfaces for Vision-impaired Users, 2021 IEEE Symposium on Visual Languages and Human-centric Computing (VLHCC2021), 10-13 October, St Louis, USA

Avazpour, I., Grundy, J.C., Vu, H. Generating Reusable Visual Notations using Model Transformation, International Journal of Software Engineering and Knowledge Engineering, Vol. 25, No. 2 (2015) pp. 1–29.

Grundy, J.C. and Hosking, J.G. Supporting generic sketching-based input of diagrams in a domain-specific visual language meta-tool, In Proceedings of the 2007 IEEE/ACM International Conference on Software Engineering (ICSE'07), Minneapolis, USA, May 2007, IEEE CS Press

Grundy, J.C., Hosking, J.G., Cao, S., Zhao, D., Zhu, N., Tempero, E. and Stoeckle, H. Experiences developing architectures for realising thin-client diagram editing tools, Software – Practice and Experience, vol. 37, no.12, Wiley, October 2007, pp. 1245-1283

Liu, N., Hosking, J.G. and Grundy, J.C., MaramaTatau: extending a domain specific visual language meta tool with a declarative constraint mechanism, In Proceedings of the 2007 IEEE Symposium on Visual Languages and Human-Centric Computing, USA, Sept 23-27 2007, IEEE CS Press.

Zhao, D., Grundy, J.C. and Hosking, J.G. Generating mobile device user interfaces for diagram-based modelling tools, In Proceedings of the 2006 Australasian User Interface Conference, Hobart, Australia, January 2006

Mehra, A., Grundy, J.C. and Hosking, J.G. A generic approach to supporting diagram differencing and merging for collaborative design, In Proceedings of the 2005 ACM/IEEE International Conference on Automated Software Engineering, Long Beach, California, Nov 7-11 2005, IEEE Press, pp. 204-213

Grundy, J.C. and Hosking, J.G. Engineering plug-in software components to support collaborative work, Software - Practice and Experience, Vol. 32, No. 10, August 2002, Wiley, 983-1013

Grundy, J.C. and Hosking, J.G. Developing Adaptable User Interfaces for Component-based Systems, Interacting with Computers, vol. 14, no. 3, March 2002, Elsevier, pp. 175-194

Grundy, J.C. and Jin, W. Experiences developing a thin-client, multi-device travel planning application, in Proceedings of 2002 New Zealand Conference on Computer-Human Interaction, July 12-13, Hamilton, New Zealand

Grundy, J.C., Wang, X. and Hosking, J.G. Building Multi-Device, Component-Based, Thin-Client Groupware: Issues and Experience, In Proceedings of the 2002 Australasian User Interface Conference, Melbourne, Australia

Petrovski, A. and Grundy, J. Web-enabling an integrated health informatics system, In Proceedings of the 7th Conference on Object-oriented Information Systems, 2001, Springer Lecture Notes in Computer Science

Grundy, J.C. Multi-perspective specification, design and implementation of components using aspects, International Journal of Software Engineering and Knowledge Engineering, Vol. 10, No. 6, December 2000, World Scientific

Grundy, J.C., Hosking, J.G., Mugridge, W.B., Apperley, M.D. A decentralised architecture for software process modelling and enactment, IEEE Internet Computing: Special Issue on Software Engineering via the Internet, Vol. 2, No. 5, September/October 1998, IEEE CS Press, pp. 53-62

Grundy, J.C. and Hosking, J.G. Human-Computer Interaction Issues for the Configuration of Component-based Software Systems, In Proceedings of OZCHI'99, Wagga Wagga, Australia, Nov 1999, pp. 37-43.

Grundy, J.C., Mugridge, W.B., Hosking, J.G., Amor, R.W. Support for Collaborative, Integrated Software Development, in Proceedings of the 7th Conference on Software Engineering Environments (SEE'95), IEEE CS Press, Netherlands, April 5-7, 1995, pp. 84-94