MONASH University

MONASH
INFORMATION
TECHNOLOGY

Human-centric Software Engineering for Next Generation Cloud- and Edge-based Applications

ARC Laureate Professor John Grundy

GROUP OF EIGHT AUSTRALIA

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.

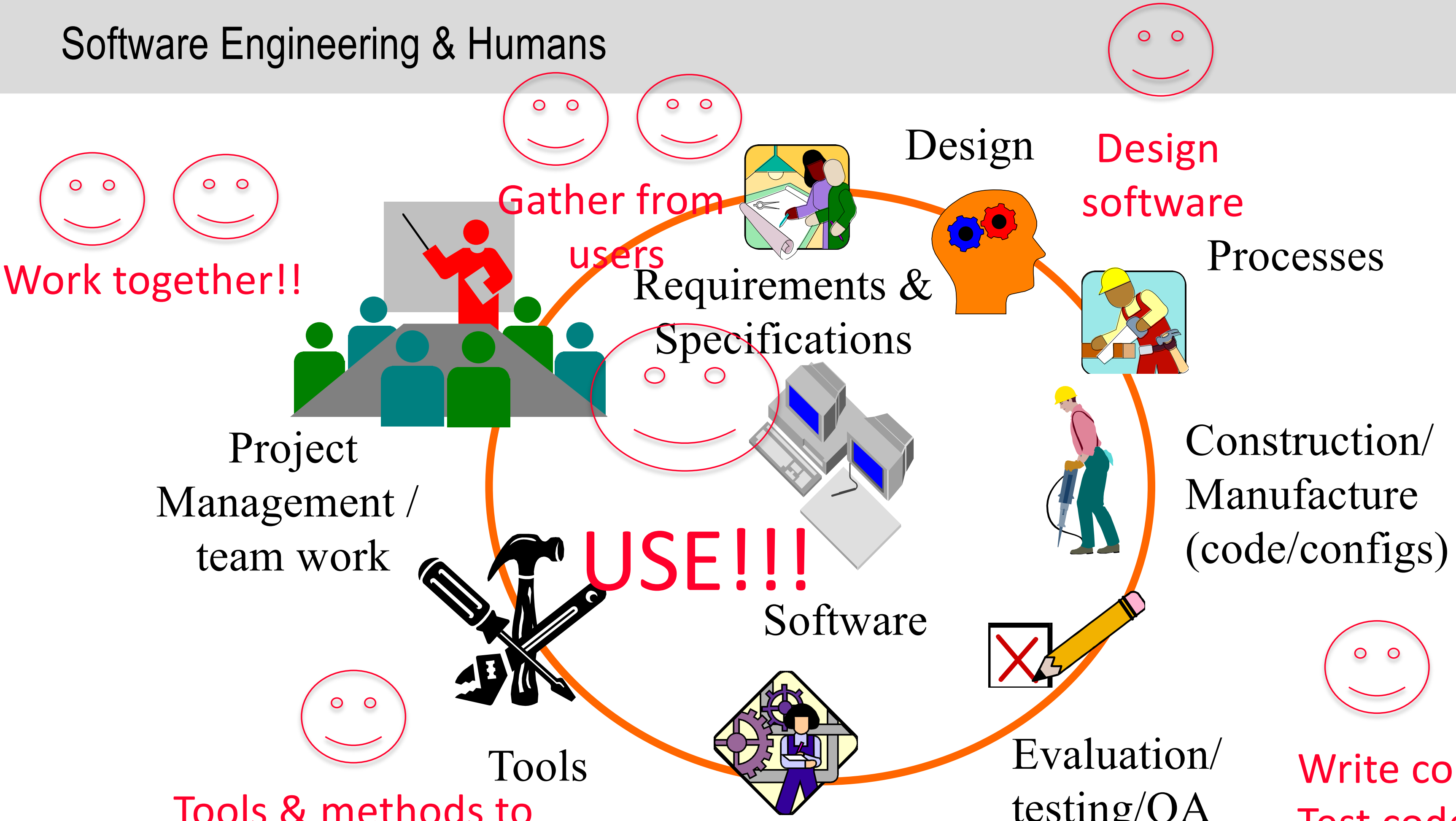
I acknowledge the people of the Kulin nations, the traditional owners of the land on which I am meeting with you from today.

Outline

- Software Engineering & humans
- Examples from our work
  - Human-centric, domain-specific visual models for non-technical experts to specify and generate apps and data analysis applications
  - Personality impact on aspects of software development
  - Incorporating end user emotions into software requirements engineering for eHealth apps
  - Fog-based workflow performance analysis
  - Visualising smart city data
  - Deploying computation and managing caching for next-generation edge apps
  - Human-centric privacy requirements in smart buildings
- Outstanding challenges, issues
- Future directions

- Software Engineering & humans
- Examples from our work
- Outstanding challenges, issues
- Future directions

Software Engineering & Humans



Work together!!

Gather from users

Design

Design software

Requirements & Specifications

Processes

Project Management / team work

Construction/ Manufacture (code/configs)

USE!!!

Software

Tools

Tools & methods to support

Methods

Evaluation/ testing/QA

Write code, Test code

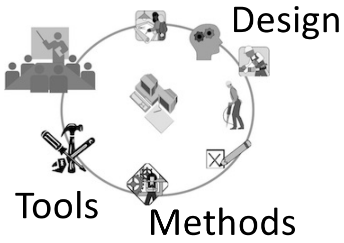
Problems if we don't include human perspective…

- Gender bias – UIs, seat belts, health app
- Ethnic bias – over-recommend minorities for search, don't recognize faces
- 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
- Physical challenge bias – guesture, sound, sight, voice inappropriate
- Cognitive challenge bias – raise anxiety, poor fit to mental model
- Enjoyment bias – boring, unengaging, distracting
- Emotional bias – stressful, anxiety-inducing, frightening
- Personality bias – workflow, lack of engagement, disconnected

All Can Apply to TEAM and USERS!!!

- Software Engineering & humans
- Examples from our work
- Outstanding challenges, issues
- Future directions

Human-centric, domain-specific visual models (DSVLs)

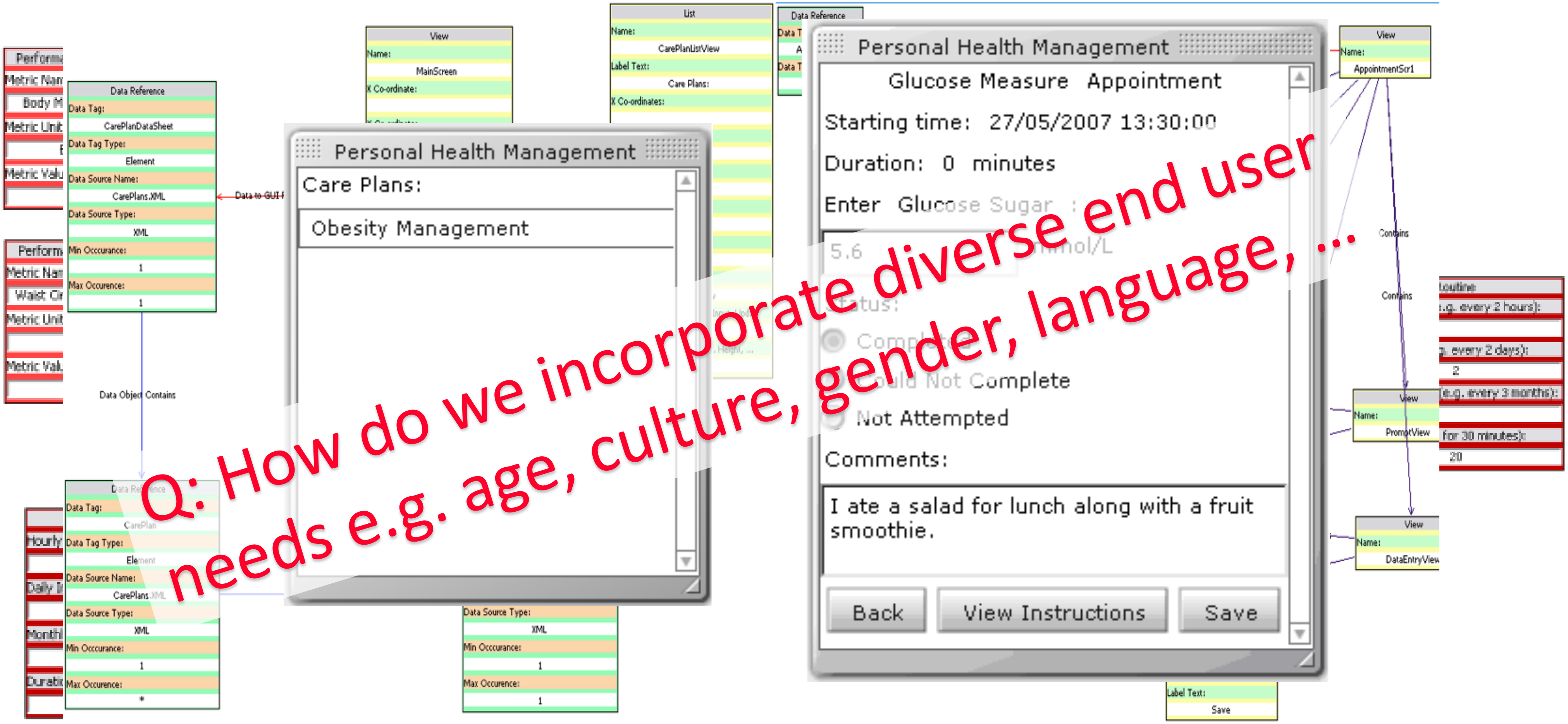


- Idea: complex models hard to work with for developers
  - And non-develpers!!
- Represent using more ”human-centric” way – visual metaphors, visual constructs – “like what sketch on a napkin in a café…” ☺
- We have a (very) large body of work on this:
  - DSVL Platforms – MViews, JViews, Pounamu, Marama, Horus, …
  - Software Engineering uses – Design tool generators, software architecture, performance engineering, user interfaces, requirements, testing, software visualisations, traceability, …
  - “End-user” Application modelling and generation – Statistical Design Language, Report Generation Language, Mobile Health App generation, Business processes, Music, Games, Visual Wikis, Data analytics, …

Example #1: Mobile Health app generation

- Scenario: want to model, generate range of eHealth apps
- Mobile phone-based personal health care planning applications
- Two meta-models with associated DVSLs: Visual Health Care Planning Language, Visual Care Application Model
- Model generic care plan with a visual DSVL tool
- Configure generic care plan for individual
- Model mobile app UI for individual from tailored care plan with a visual DSVL tool
- Generate Flash, Windows Mobile, iPhone app code

VHCPL



Personal Health Management

Care Plans:

Obesity Management

Personal Health Management

Glucose Measure Appointment

Starting time: 27/05/2007 13:30:00

Duration: 0 minutes

Enter Glucose Sugar : 5.6 mmol/L

Status:

○ Completed

○ Could Not Complete

○ Not Attempted

Comments:

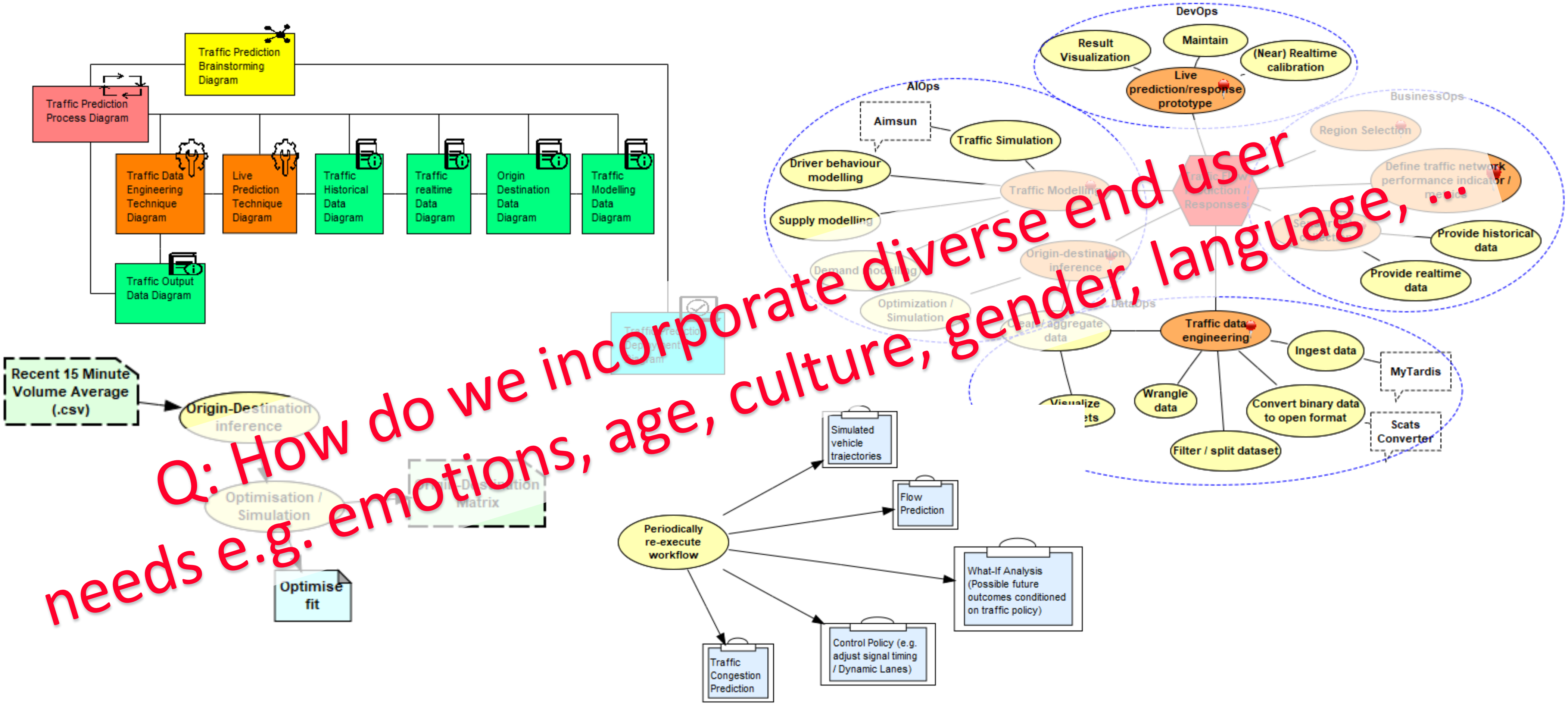
I ate a salad for lunch along with a fruit smoothie.

Q: How do we incorporate diverse end user needs e.g. age, culture, gender, language, …

Example #2: BiDaML

- Scenario: developing new data analytics solution
- Traditionally: domain experts can’t talk to data scientists can’t talk to software engineers can’t talk to end users…
- Alternative: a common set high->low level modelling visual languages
- Visual models include brainstorming diagrams, task diagrams, technique diagrams, data diagrams, deployment diagrams…
- Applied with various companies e.g.

BiDaML example – VicRoads data traffic flow analysis

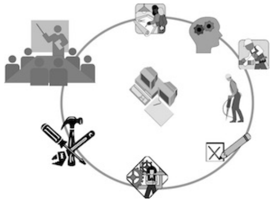


Traffic Prediction Brainstorming Diagram
Traffic Prediction Process Diagram
Traffic Data Engineering Technique Diagram
Live Prediction Technique Diagram
Traffic Historical Data Diagram
Traffic realtime Data Diagram
Origin Destination Data Diagram
Traffic Modelling Data Diagram
Traffic Output Data Diagram
Recent 15 Minute Volume Average (.csv)
Origin-Destination inference
Optimisation / Simulation
Origin-Destination Matrix
Optimise fit
Traffic Prediction Deployment Diagram
DevOps
Result Visualization
Maintain
(Near) Realtime calibration
Live prediction/response prototype
AIOps
Aimsun
Traffic Simulation
Driver behaviour modelling
Traffic Modelling
Supply modelling
Demand modelling
Origin-destination inference
Optimization / Simulation
Traffic Flow Prediction / Responses
BusinessOps
Region Selection
Define traffic network performance indicator / metrics
Provide historical data
Provide realtime data
DataOps
Clean / aggregate data
Traffic data engineering
Ingest data
MyTardis
Visualize datasets
Wrangle data
Convert binary data to open format
Scats Converter
Filter / split dataset
Simulated vehicle trajectories
Flow Prediction
Periodically re-execute workflow
What-If Analysis (Possible future outcomes conditioned on traffic policy)
Control Policy (e.g. adjust signal timing / Dynamic Lanes)
Traffic Congestion Prediction

Q: How do we incorporate diverse end user needs e.g. emotions, age, culture, gender, language, …

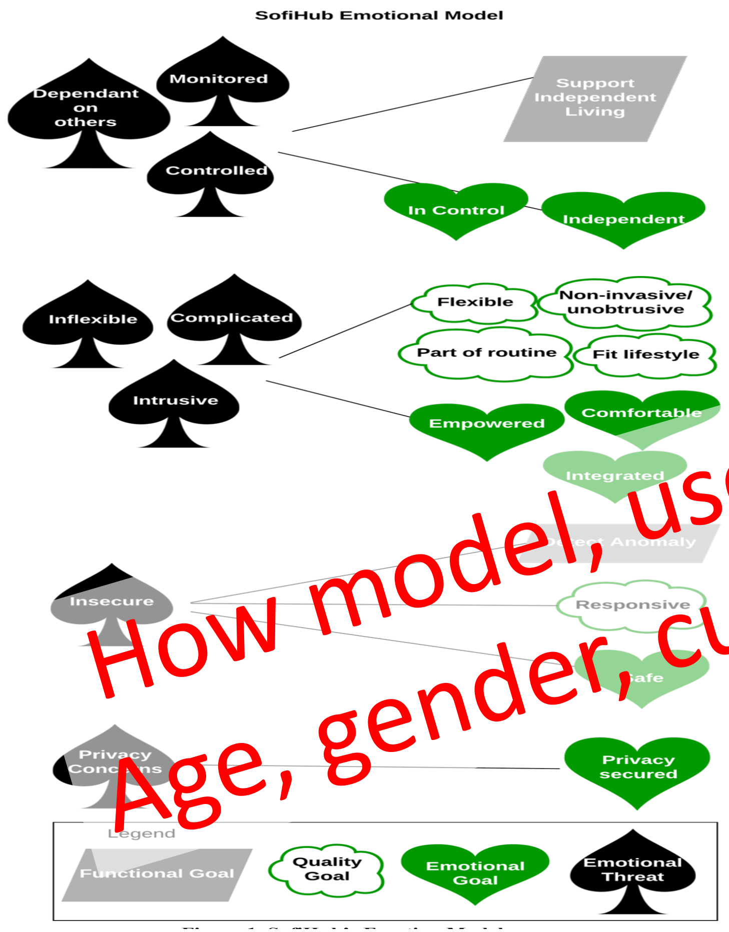
Incorporating end user emotions into software requirements engineering

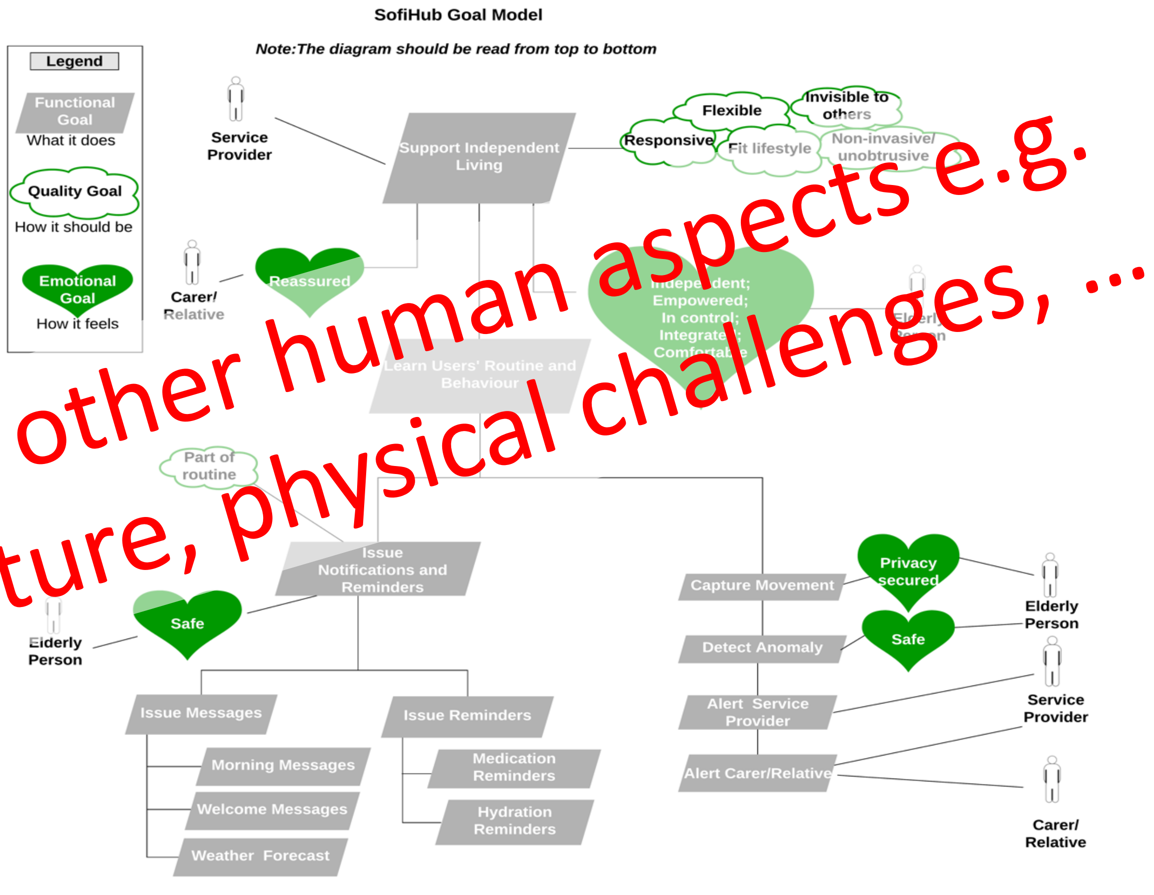
Requirements



- People use software
- Software is designed to help people perform tasks, solve problems
- But – people react to software / tasks / situations in various ways
- One (under-researched) way is emotional reactions to software usage
- Incorporating emotions / emotional reactions into software requirements, design, evaluation
- Applying to eHealth systems: smart homes, dementia training apps, chatbot design

Example: requirements for the Smart Home

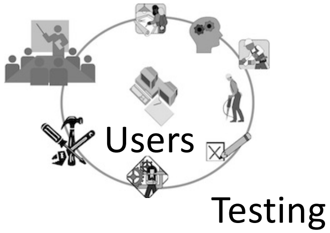




How model, use other human aspects e.g.

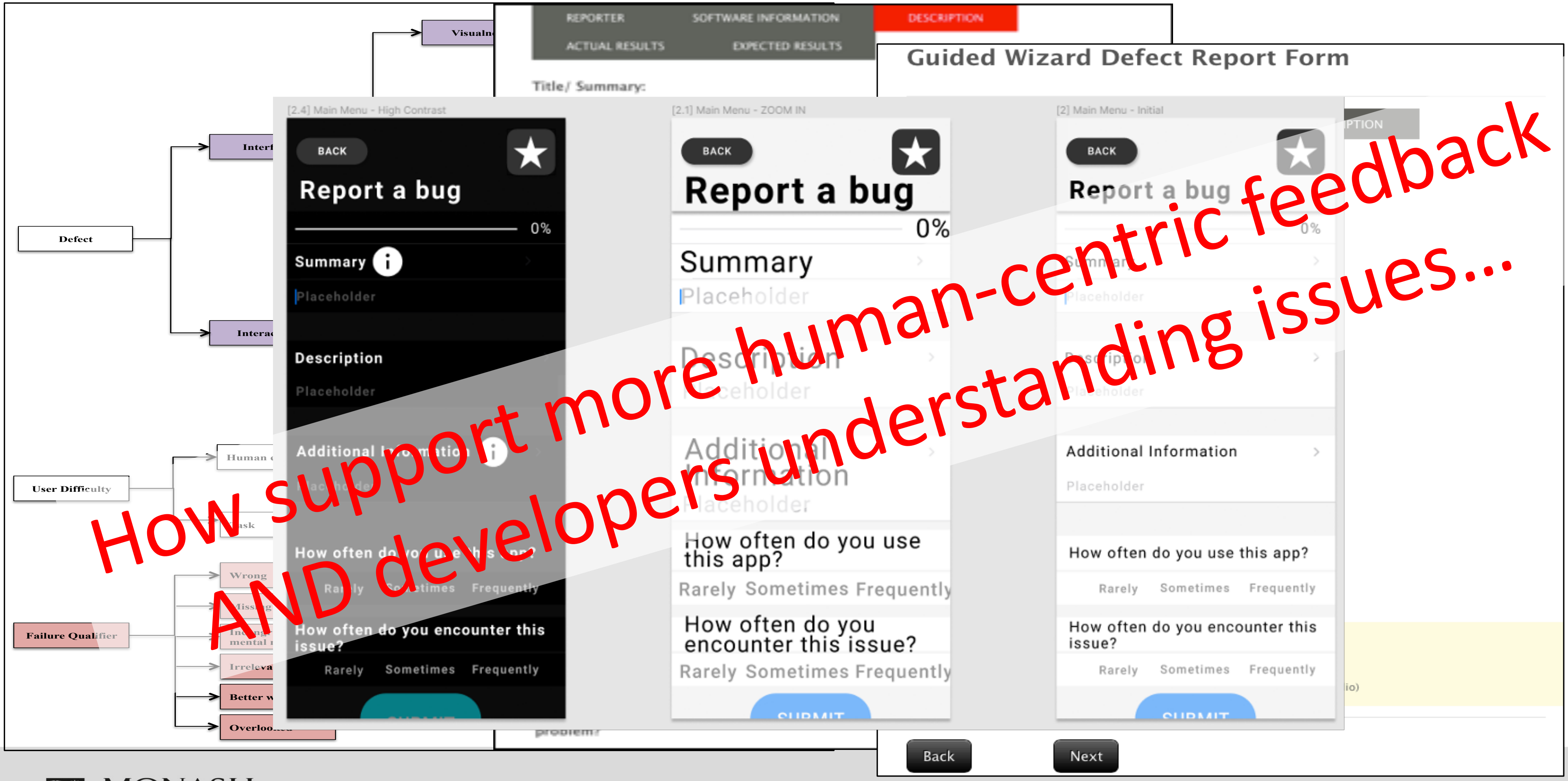
Age, gender, culture, physical challenges, …

Reporting usability defects



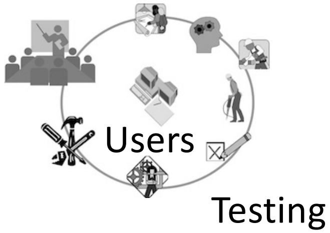
- Software typically has a bunch of “defects”
- Functional and non-functional
- One under-researched non-functional area are usability defects
  - Problems with how users interact with the software
- How do we currently find, report, fix these?
- How can we improve the reporting?
- Better understand current reporting needs: survey, repository mining, observation
- New usability defect taxonomy to better characterise usability defects
- New usability defect reporting tool

Usability Defect Taxonomy & Reporting



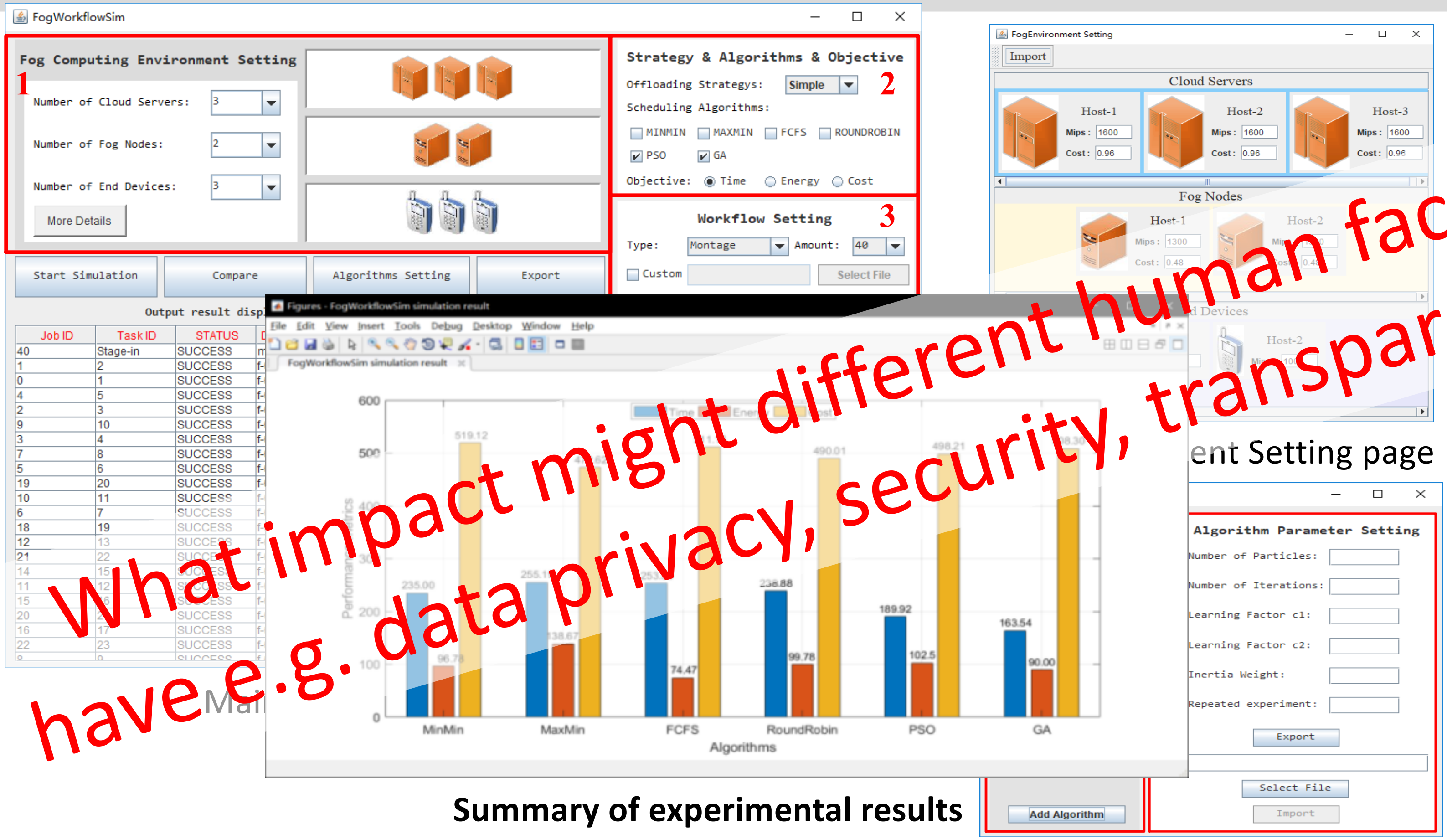
How support more human-centric feedback AND developers understanding issues…

Fog Application Performance



- Need to deploy large scale sensor applications on edge/fog
- We have particular interest in workflow systems on cloud / edge / fog platforms
- Earlier work did extensive analysis on cloud…
- …but how does fog deployment differ?
- E.g. workflow in scientific app for running a smart lab infrastructure, industry 4.0 infrastructure…

FogWorkflowSim



FogWorkflowSim

1 Fog Computing Environment Setting

Number of Cloud Servers: 3

Number of Fog Nodes: 2

Number of End Devices: 3

More Details

2 Strategy & Algorithms & Objective

Offloading Strategys: Simple

Scheduling Algorithms:

□ MINMIN □ MAXMIN □ FCFS □ ROUNDROBIN

☑ PSO ☑ GA

Objective: ● Time ○ Energy ○ Cost

3 Workflow Setting

Type: Montage Amount: 40

□ Custom Select File

Start Simulation | Compare | Algorithms Setting | Export

Output result disp

Job ID	Task ID	STATUS
40	Stage-in	SUCCESS
1	2	SUCCESS
0	1	SUCCESS
4	5	SUCCESS
2	3	SUCCESS
9	10	SUCCESS
3	4	SUCCESS
7	8	SUCCESS
5	6	SUCCESS
19	20	SUCCESS
10	11	SUCCESS
6	7	SUCCESS
18	19	SUCCESS
12	13	SUCCESS
21	22	SUCCESS
14	15	SUCCESS
11	12	SUCCESS
15	16	SUCCESS
20	[illegible]	SUCCESS
16	17	SUCCESS
22	23	SUCCESS

Figures - FogWorkflowSim simulation result

FogEnvironment Setting

Cloud Servers

Host-1	Host-2	Host-3
Mips: 1600	Mips: 1600	Mips: 1600
Cost: 0.96	Cost: 0.96	Cost: 0.96

Fog Nodes

Host-1	Host-2
Mips: 1300	Mips: 1300
Cost: 0.48	Cost: 0.48

End Devices

Host-2

Algorithm Parameter Setting

Number of Particles:

Number of Iterations:

Learning Factor c1:

Learning Factor c2:

Inertia Weight:

Repeated experiment:

What impact might different human factors have e.g. data privacy, security, transparency…

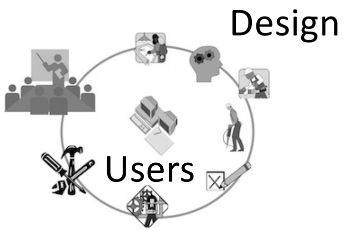
Mai

ent Setting page

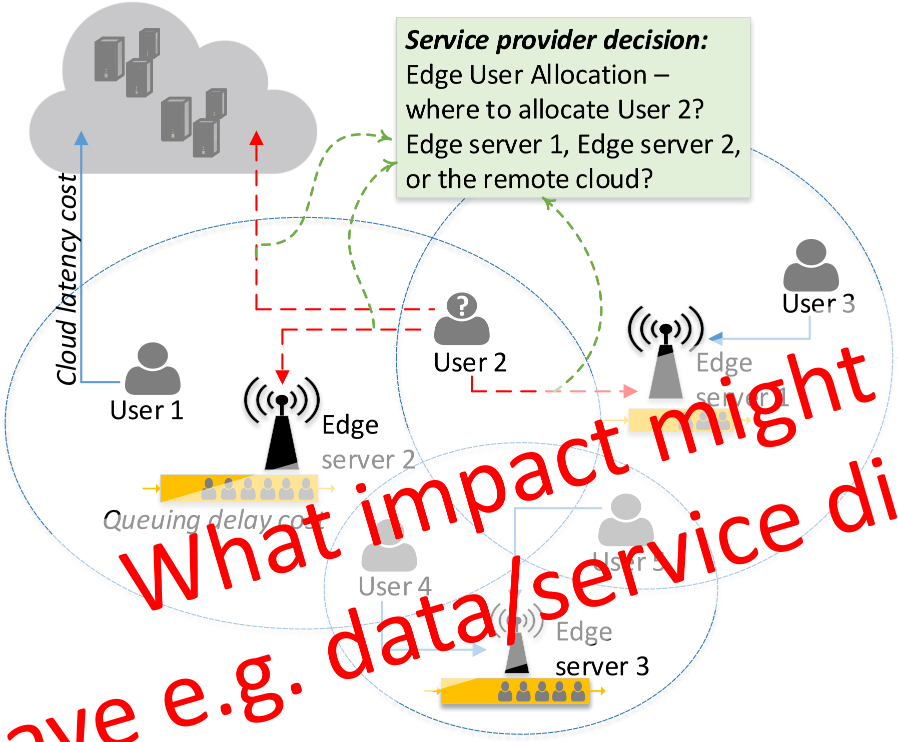
Summary of experimental results

Algorithm Setting page

Deploying large edge-based applications



- How do we optimally distribute compute & data on large edge-based applications?
- How do we distribute users, based on human aspects & functional requirements?
- How do we cache data to optimize performance, again based on human aspects and functional requirements?
- How do we adapt at run-time as movement, changing functions, new devices/edge servers etc. change?

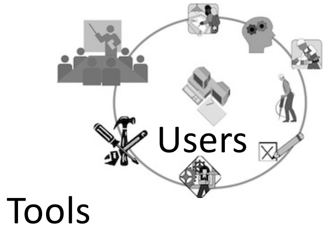


Service provider decision:
Edge User Allocation –
where to allocate User 2?
Edge server 1, Edge server 2,
or the remote cloud?

- Set of cloud services, edge servers, edge devices
- Set of services, data sets
- Set of users
- Overlapping edge servers, data, services, users…
- How allocate services, data, users to optimize esp over 5G connections…

What impact might different human factors have e.g. data/service distribution, privacy, security, ..

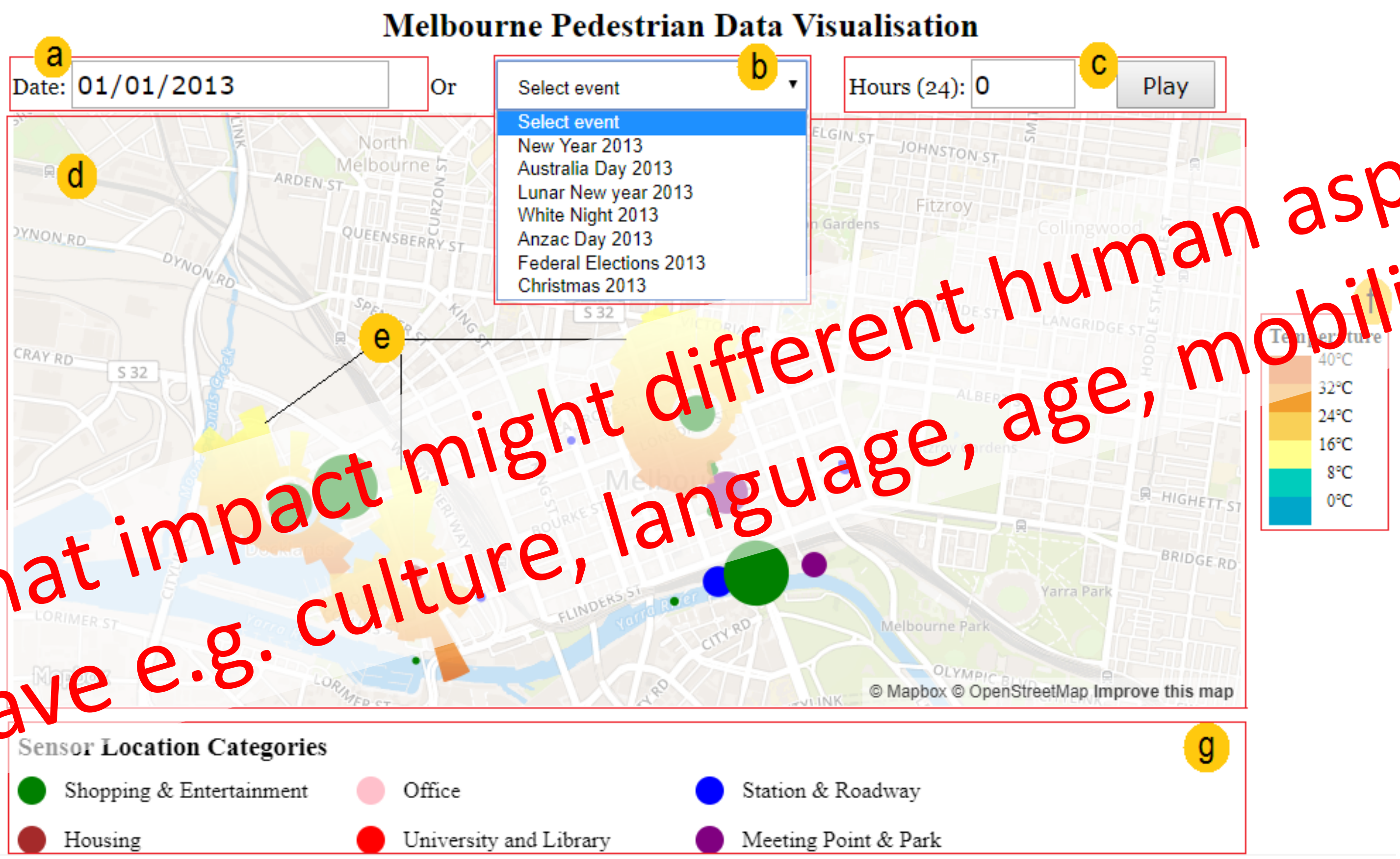
Visualising smart city data in human-centric ways



- Smart cities generate heaps data
- Sources include cloud, edge services but also humans
- Integration with traditional system data adds even more…

- What data will help operators, planners to make better decisions?
- What data is useful for citizens?
- How do we manage large scale distribution, privacy, security, scalability…?

PedaViz



Melbourne Pedestrian Data Visualisation

Date: 01/01/2013 Or Select event

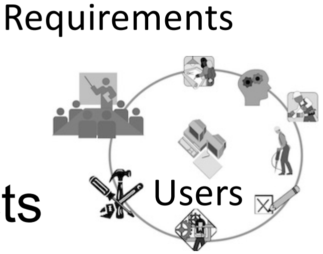
Hours (24): 0 Play

Sensor Location Categories

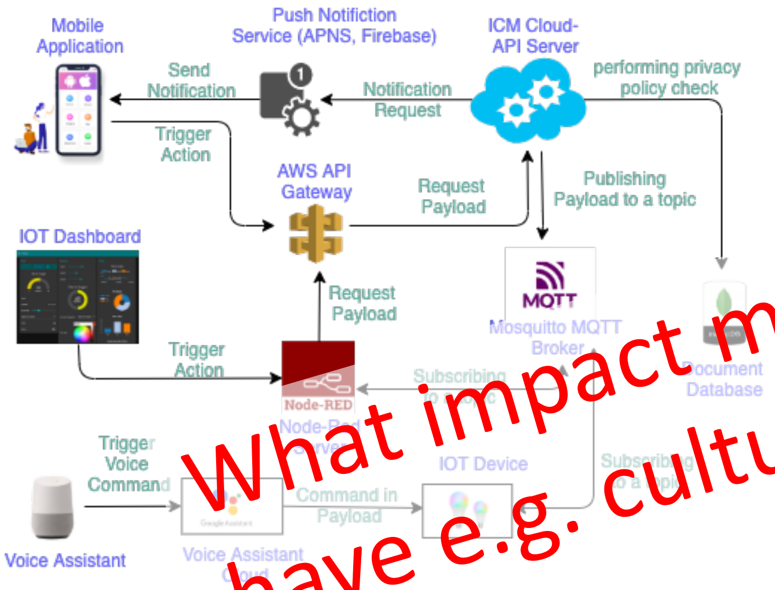
● Shopping & Entertainment	● Office	● Station & Roadway
● Housing	● University and Library	● Meeting Point & Park

What impact might different human aspects have e.g. culture, language, age, mobility, …

Privacy requirements for smart buildings



- Smart buildings have wide range of edge devices and servers
- Have a wide range of end users with wide range of human aspects
- Want to support informed privacy consent
- Developed new model, architecture and prototype
- Want to simulate with large number of (diverse) users



Mobile Application

Push Notifiction Service (APNS, Firebase)

ICM Cloud-API Server

Send Notification

Notification Request

performing privacy policy check

Trigger Action

AWS API Gateway

Request Payload

Publishing Payload to a topic

IOT Dashboard

Request Payload

Mosquitto MQTT Broker

Trigger Action

Node-RED Server

Subscribing to a topic

Document Database

Trigger Voice Command

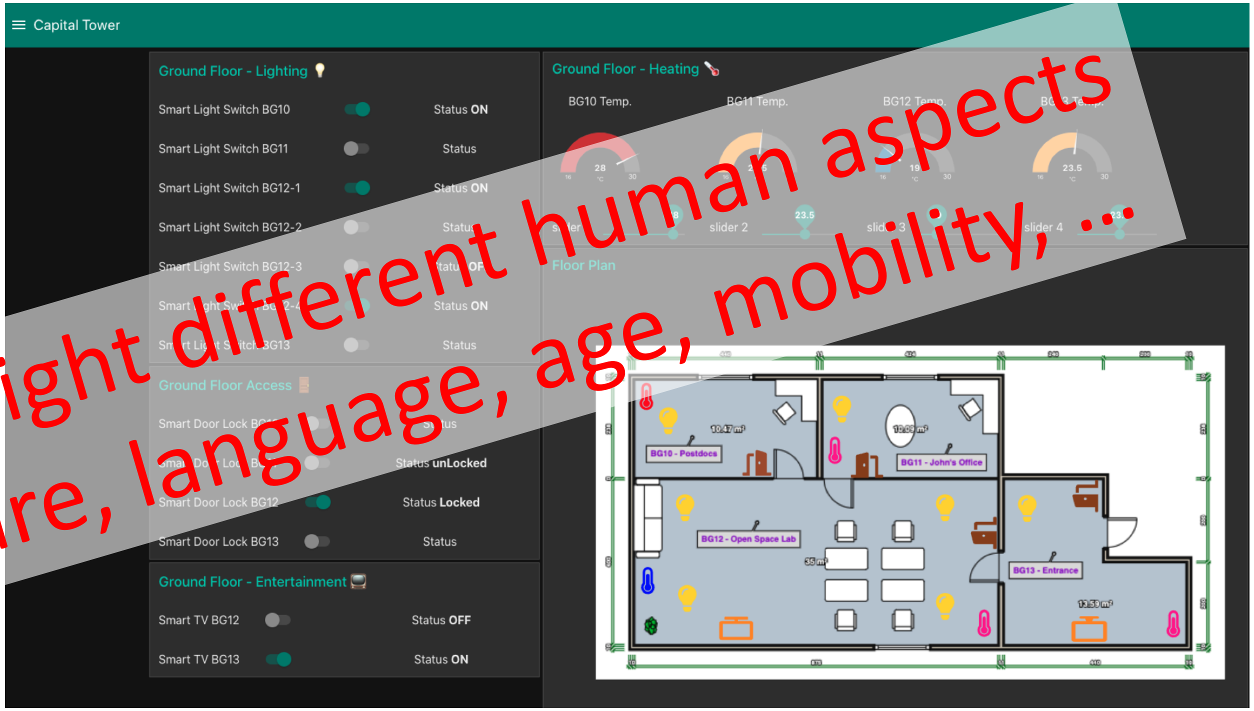
Command in Payload

IOT Device

Subscribing to a topic

Voice Assistant

Voice Assistant Cloud



Capital Tower

Ground Floor - Lighting

Smart Light Switch BG10 Status ON

Smart Light Switch BG11 Status

Smart Light Switch BG12-1 Status ON

Smart Light Switch BG12-2 Status

Smart Light Switch BG12-3 Status OFF

Smart Light Switch BG12-4 Status ON

Smart Light Switch BG13 Status

Ground Floor Access

Smart Door Lock BG10 Status

Smart Door Lock BG11 Status unLocked

Smart Door Lock BG12 Status Locked

Smart Door Lock BG13 Status

Ground Floor - Entertainment

Smart TV BG12 Status OFF

Smart TV BG13 Status ON

Ground Floor - Heating

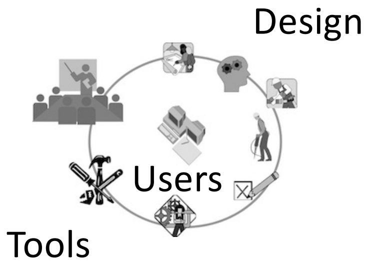
BG10 Temp. BG11 Temp. BG12 Temp. BG13 Temp.

slider 1 slider 2 slider 3 slider 4

Floor Plan

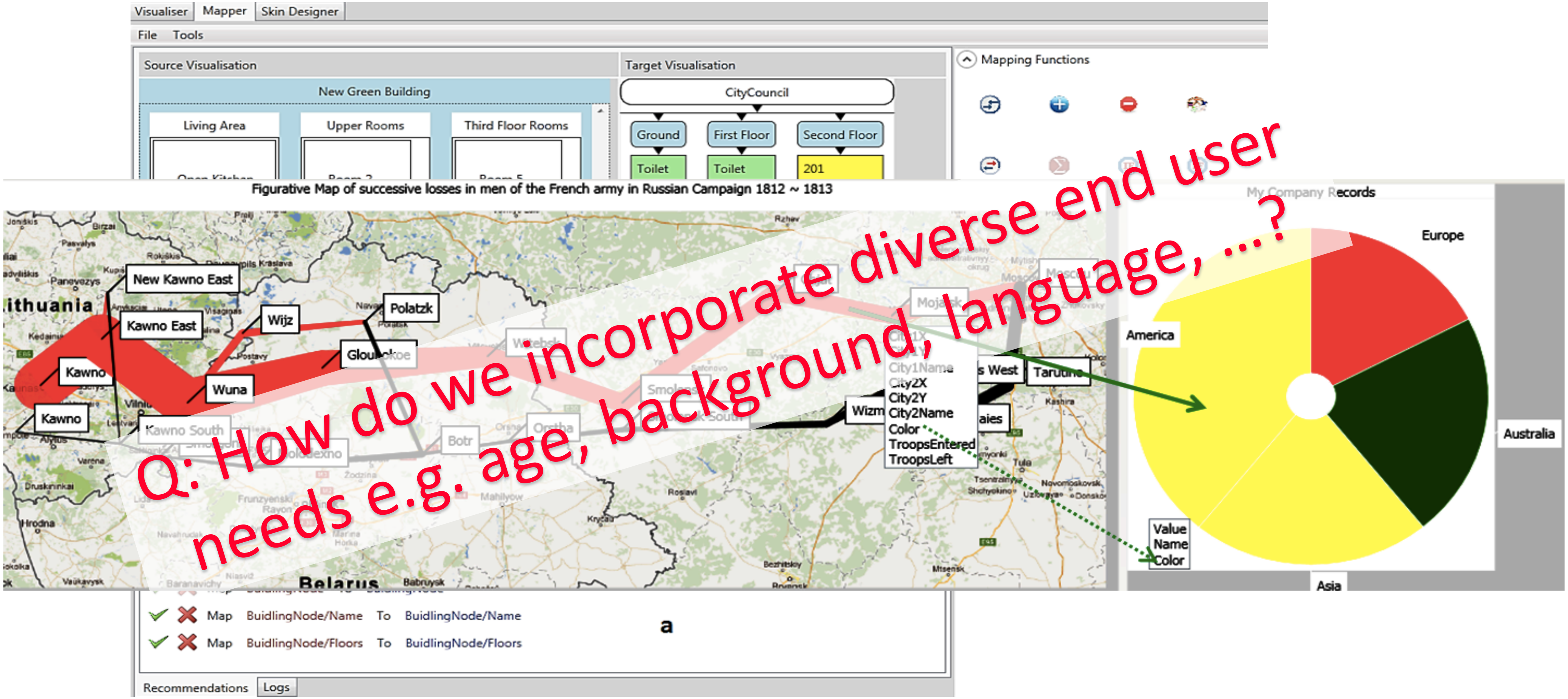
What impact might different human aspects have e.g. culture, language, age, mobility, …

End-user development of solutions – lets get rid of software engineers ☺



- Scenario: complex XML or EDI message format; want to translate into a different format; then process e.g. data wrangling, harmonization ☺
- Traditionally: write QVT/ATL/XSLT/code to do
- Alternative: model transformation visually and generate these transformation implementations
- Meta-model = source/target and mappings
- Visual models might include forms, trees, concrete data visualisations
- Model-driven Engineering = generate XSLT, ATL, Code (C++, Java),…
- We have developed various approaches to this…

CONVErT – by-example based data mapping/integration/visualisation



✓ ✗ Map BuidlingNode/Name To BuidlingNode/Name

✓ ✗ Map BuidlingNode/Floors To BuidlingNode/Floors

a

Q: How do we incorporate diverse end user needs e.g. age, background, language, …?

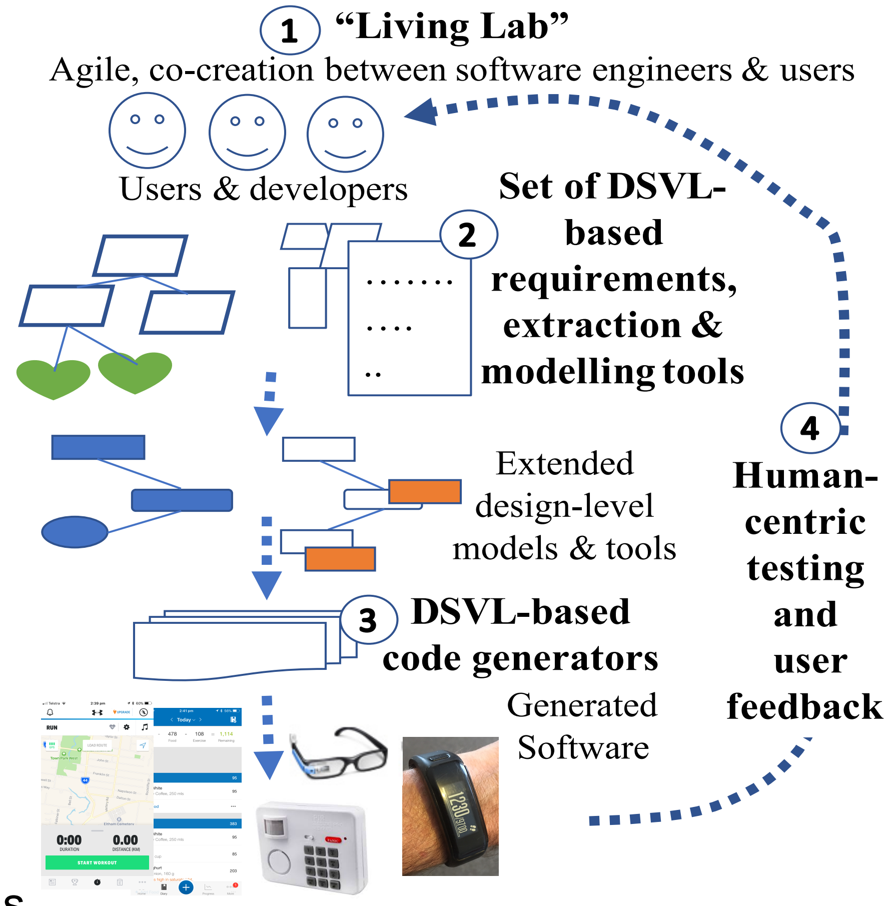
- Software Engineering & humans
- Examples from our work
- Outstanding challenges, issues
- Future directions

Challenges ; Outstanding issues

- Often software engineers don’t understand / appreciate human aspects of SE
- Neither it seems do MBIE (NZ) or ARC (Australia) grant Assessors…. ☹
  - So saying – perhaps my ARC Laureate and last Discovery grant are counter-examples ☺
- Designing and conducting experiments is hard, time-consuming
- Often need access to practitioners ; convincing them/their bosses can also be a challenge
- Many issues not yet well explored, but increasing interest in SE community
- I find them more challenging – but also in many ways more interesting – projects than the purely technical ones I do
- Recruiting (very good) students / post-docs to work on can be hard, but I’ve been pretty lucky to date…
- IMO – good research in these areas can make a major difference to practice

How we are tackling (some of) these issues…

- Human-centric
  - Living lab co-creation space idea
  - Personality, emotions, physical and mental challenges, gender, age, culture, language, …
  - Model these aspects of requirements, design solutions using Domain-Specific Visual Languages (DSVLs)
  - Reason about completeness of models for diverse end users of software applications
- Model-driven
  - Incorporate these human aspects into code generators
  - Auto-adapt produced applications to different end-user needs, implicitly (learned) and explicitly (configured)
  - Requirements-based testing of generated applications



1 **"Living Lab"**

Agile, co-creation between software engineers & users

Users & developers

2 **Set of DSVL-based requirements, extraction & modelling tools**

Extended design-level models & tools

3 **DSVL-based code generators**

Generated Software

4 **Human-centric testing and user feedback**

Summary

- Human aspects of Software Engineering are fascinating!!
- There is lots of scope for work here
- Can apply other discipline approaches, knowledge – Information Systems, Social Sciences, etc
- Ultimately humans PRODUCE software and humans USE software
- Incorporating human perspectives critical to improve software and its production
- Smart cities applications e.g. traffic analysis & control ; smart homes and buildings ; very large scale edge/fog applications a challenging domain to address these in – diverse end users & developers ; complex ; evolving

Questions…

References

- Grundy, J.C, Hosking, J.G., Amor, R., Mugridge, W.B., Li, M. Domain-specific visual languages for specifying and generating data mapping system, Journal of Visual Languages and Computing, vol. 15, no. 3-4, June-August 2004, Elsevier, pp 243-263
- Avazpour, I., Grundy, J.C., Grunske, L. Specifying Model Transformations by Direct Manipulation using Concrete Visual Notations and Interactive Recommendations, Journal of Visual Languages and Computing, Volume 28, June 2015, Elsevier,pp 195–211.
- Abizer Khambati, John Grundy, John Hosking, and Jim Warren, Model-driven Development of Mobile Personal Health Care Applications, In Proceedings of the 2008 IEEE/ACM International Conference on Automated Software Engineering, L'Aquilla, Italy, 15-19 September 2008, IEEE CS Press
- Kamalrudin, M., Grundy, J.C., Hosking, J.G., MaramaAIC: Tool Support for Consistency Management and Validation of Requirements, Automated Software Engineering, Springer, 2017, vol 24, no 1, pp. 1-45
- Sallah, N., Mendes, E., Grundy, J.C. Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments, Empirical Software Engineering, vol. 19, no. 3, Springer, 2014, pp. 714-752.
- Kanij, T., Merkel, R., Grundy, J.C. Performance Appraisal of Software Testers, Information and Software Technology, Elsevier, vol. 56, no. 5, May 2014, Pages 495–505
- Yusop, N.S.M., Grundy, J.C., Vasa, R. Reporting Usability Defects: A Systematic Literature Review, IEEE Transactions on Software Engineering, vol. 43, no. 9, 2017, pp. 848-867.
- Huynh, K., Benarivo, J., Xuan, C.D., Sharma, G.G., Kang, J., Grundy, J.C., Madugalla, A., Improving Human-Centric Software Defect Evaluation, Reporting, and Fixing, 2021 IEEE International Conference on Computers, Software, and Applications Conference (COMPSAC2021), July 12-16 2021, online.
- Ali, N.M., Hosking, J.G., Grundy, J.C., A Taxonomy and Mapping of Computer-based Critiquing Tools, IEEE Transactions on Software Engineering, vol. 39, no. 11, November 2013.
- Li, C., Yu, Y., Leckning, J., Xing, W., Fong, C., Grundy, J.C., Karolita, D., McIntosh, J., Obie, H. A human-centric approach to building a smarter and better parking application, 2021 IEEE International Conference on Computers, Software, and Applications Conference (COMPSAC2021), July 12-16 2021, online.
- Grundy, J.C. Abdelrazek, M., Kissoon, M., Vision: Improved development of mobile eHealth applications, IEEE/ACM International Conference on Mobile Software Engineering and Systems (MobileSoft 2018), 27-28 May 2018, Gothenberg, Sweden, ACM Press.
- Obie, H., Chua, C., Avazpour, I., Abdelrazek, M., Grundy, J.C. PedaViz: Visualising Hour-Level Pedestrian Activity, 11th International Symposium on Visual Information Communication and Interaction (VINCI 2018), Växjö, Sweden, 13-15 August 2018
- Salleh, N., Hoda, R., Su, M.T., Kanij, T. and Grundy, J.C. Recruitment, Engagement and Feedback in Industrial Empirical Software Engineering Studies, to appear in Information and Software Technology, Elsevier. –
- Soomro, A.B., Salleh, N., Mendes, E., Grundy, J.C., Burch, G., Nordin, A., The Effect of Software Engineers' Personality traits on Team Climate and Performance: a Systematic Literature Review, Information and Software Technology, vol 73, Elsevier, pp 52-65.
- Liu, X., Fan, L., Xu, J., Li, X., Gong, L., Grundy, J.C., Yang, Y. FogWorkflowSim: An Automated Simulation Toolkit for Workflow Performance Evaluation in Fog Computing, 34th IEEE/ACM International Conference on Automated Software Engineering (ASE 2019), San Diego, CA, USA, November 11-15, 2019
- Grundy J., Khalajzadeh H., McIntosh J., Kanij T., Mueller I. (2021) HumaniSE: Approaches to Achieve More Human-Centric Software Engineering. In: Ali R., Kaindl H., Maciaszek L.A. (eds) Evaluation of Novel Approaches to Software Engineering. ENASE 2020. Communications in Computer and Information Science, vol 1375. Springer