Aims of the project

- The focus in this project is on client-owned and client-controlled user-models, in which the user manages and controls their own user-model.
- In a truly ubiquitous computing environment, user-modelling frameworks will want to (provided they have permission) communicate with one another, and be able to synchronise content and resolve conflicts from a variety of devices including smart phones, TVs, and other more traditional computers.

Scenario

- Tom is looking for a job and he wants to fine tune his current resume. On going to a workshop to improve his resume writing skills he uses his Smartphone to make notes so that he does not forget when he edits his resume at home. Tom does not like the small Smartphone screen so when he arrives home, he will continue working on his resume on his desktop.

Personis and PersonisJ

- The Personis server is a user modelling framework which is able to support the use and reuse of a user’s user model over multiple applications and multiple devices.
- PersonisJ is a user modelling framework which supports client-side personalisation on mobile devices running the Android platform.

Scenario walkthrough

1. Each device has a unique ID that is used for the root user model.
2. Application contexts of the user model.
3. The context “Career Organiser” is marked for synchronisation.
4. A menu pops up to activate the synchronisation process.
5. List of components under “Career Organiser”
6. List of evidences for a particular component

Synchronisation algorithm

LAST_SYNC_TIME = LAST SYNCHRONISATION TIME;
CLIENT EVIDENCE = ALL EVIDENCE SINCE LAST_SYNC_TIME FROM LOCAL CLIENT
SERVER EVIDENCE = ALL EVIDENCE SINCE LAST_SYNC_TIME FROM SERVER
IF( CLIENT EVIDENCE AND SERVER EVIDENCE == NULL)
    RETURN;
IF( CLIENT EVIDENCE IS MORE RECENT THAN SERVER EVIDENCE)
    SEND EVIDENCE FROM LOCAL CLIENT TO SERVER
ELSE IF ( CLIENT EVIDENCE IS OLDER THAN SERVER EVIDENCE)
    RECEIVE EVIDENCE FROM SERVER TO LOCAL CLIENT

Usability study

- Participants found that it is very important that their user models on their devices should be synchronised with each other since people are increasingly obtaining more devices with requires user modelling.
- Having a user model anywhere, anytime and on any device is especially useful. This allows the user model to be used locally while offline and for it to be synchronised when next online.

Structure of the user model

The diagram below shows one example of structure of the hierarchy of the user model including sub-models from a number of different mobile applications.

Treatise contribution

- The synchronisation functionality of user models between different applications with different form factors.
- Has the ability to synchronise parts of the user model. It is important that the user is in control of their entire model and allow parts of their own user model to other applications and devices.
- Users will be able to inspect their entire user model in a tree-like structure. Also inspecting previous evidence of components.
- This functionality allows users to use and reuse their user models on one device to another and through different applications.
- Synchronisation allows the future work of other aspects of user modelling which are privacy, life-long modelling, time latency between synchronisation and conflict resolution of the synchronised data.

Diagram of the link achieved

- Below shows the link which this project as achieved between the two frameworks.