20th International Conference on
Information Systems Development (ISD2011)
Edinburgh, Scotland
August 24-26, 2011

Keynote Address: Wednesday 24th August 2011 – Cairn Auditorium – 13:00 – 14:00

Professor Nicholas Taylor
Heriot Watt University,
Edinburgh,
Scotland.

Is there really a conflict between privacy and personalisation?

There is growing concern about the protection of user privacy on the Internet. This is entirely understandable and long overdue. Since the advent of the web, the amount of personal information and media types which users are invited to disclose have been increasing inexorably. There are good reasons for this. Through the customisation of systems to each individual user, the provision of personal information offers significant benefits in terms of usability. Indeed, as the complexity of these systems increases, their usability becomes ever more dependent on the benefits of personalisation. There would appear to be a conflict between privacy and personalisation and a need for a balance to be struck between them – greater privacy resulting in less personalisation and vice versa. This keynote will argue that the apparent conflict is the result of a particular perspective on the problem; a perspective which assumes that the individual user is limited to a binary decision on whether or not to disclose items of personal information to all and sundry or to nobody. There are many reasons why this assumption is made, not least of which are the privacy policies of service providers which offer little or no fine tuning and frequently read more like disclaimers than guarantees. Investigation of these reasons indicates that they are not inescapable and reveals an alternative approach to the design of personalised systems in which the user can retain full control of what they disclose to whom.

20th International Conference on
Information Systems Development (ISD2011)
Edinburgh, Scotland
August 24-26, 2011

Keynote Address: Thursday 25th August 2011 – Cairn Auditorium – 09:00 – 10:00

Prof. Ian Sommerville
School of Computer Science
University of St Andrews
Scotland

Designing for Failure: Rethinking the development of complex information systems

Increasingly, large complex systems are now constructed by an assembly and configuration process where existing systems or components inter-operate to deliver the required services. We thus create 'systems of systems' where the parts of the whole system are independently managed systems in their own right. Often, these constituent systems are 'legacy systems', where we have incomplete information about the specification, design and testing of the original system.

While this approach to system construction reduces time to system deployment, it challenges the 'reductionist' model of engineering where a large system is decomposed a set of smaller components, which are then independently implemented and tested. These can then be assembled to create the whole with the testing process focusing on the integration rather than the testing of the components. The system developer has control over the system components and can change these as required to allow them to operate effectively with other components.

However, when we create system by integrating existing systems, we neither understand these independent systems nor the implicit relationships that may exist between them. There is no 'system specification' that can serve as a reference point that defines what the system should do and, critically, what the system must not do. Furthermore, because these systems are independently managed, we cannot simply change them nor can changes to these systems be 'controlled' by a central authority.

Consequently, with such systems of systems, 'failures' are inevitable. But the notion of 'failure' is a difficult one to pin down when we have no canonical description of what the 'correct' behavior of a system is. What appears to be a 'failure' to one stakeholder in a system may be seen as normal system behavior by another. Changing a system component to correct a 'failure' in one system of systems may lead to different 'failures' in another system of systems.

I argue that, because of the fundamental characteristics of systems of systems, the notion that we can produce a 'correct' system is one that is no longer supportable. Our approach to systems engineering should change to embrace failure and that a fundamental driver for system design should be failure recovery. This recovery process cannot be a completely automated process (because failure depends on the perception of the system user) and so we need to think about how to design systems to make it easier for system users to use their knowledge and experience to recover from failures without unacceptable costs being incurred.