

1) Titolo della ricerca

Approcci di EUD alla composizione di servizi per la creazione di spazi di lavoro interattivi in sistemi ubiqui

2) Area nella quale si inquadra la ricerca

Human-Computer Interaction

3) Obiettivi della ricerca

La crescente disponibilità di servizi e risorse disponibili su Internet e le possibilità offerte dal Web 2.0 stanno spingendo gli utenti a evolvere da meri consumatori di informazioni (information consumers) a creatori di nuovi contenuti (content providers), ottenuti accedendo e manipolando le risorse disponibili sul Web. Da un punto di vista della Human-Computer Interaction (HCI), questo richiede la creazione di nuovi paradigmi di interazione che possano consentire agli utenti di accedere ai contenuti, muoverli attraverso i confini della specifica applicazione da cui sono generati portali all'interno di spazi di lavoro interattivi, dove possono integrarli, creare nuovi contenuti e condividerli con altri utenti. Bisogna inoltre tener conto della pluralità di tecnologie che danno vita al cosiddetto *ubiquitous computing*, che determina nuove modalità d'interazione uomo-macchina non confinate più all'uso dei pc desktop ma estese all'utilizzo ubiquo di dispositivi come tablet, smartphone, display interattivi installati in vari luoghi, sia pubblici che privati, e sensori di vario genere.

La ricerca cui sono interessato per il mio dottorato mira a combinare i principi e tecniche di End User Development (EUD) con il potenziale offerto dalle tecnologie di composizione di servizi, come i mashup. Lo scopo è di progettare e sviluppare modelli e meccanismi di composizione intuitivi e facili da usare, che diano a utenti che non hanno competenze tecniche specifiche la possibilità di creare spazi di lavoro interattivi e personali componendo servizi e risorse che sono disponibili sul Web, in modo da creare nuovi contenuti e applicazioni che possano soddisfare i loro bisogni determinati dallo specifico contesto (situational needs). Inoltre, la crescente disponibilità di nuove tecnologie che arricchiscono lo scenario dell'*ubiquitous computing* incentiva l'uso di dispositivi alternativi al pc desktop. In particolare, sulla base delle mie esperienze sullo sviluppo di applicazioni per dispositivi multitouch di grandi dimensioni, acquisite durante lo svolgimento delle

tesi di laurea triennale e magistrale, saranno sperimentati i meccanismi di composizione usando in particolare schermi interattivi di grandi dimensioni.

4) Motivazioni della ricerca

Intorno agli anni '90 sono stati proposti i primi paradigmi di composizione. Più di recente sono stati avanzati metodi di mashup per il Web, con lo scopo di far creare applicazioni Web usando le risorse del Web stesso. In particolare, i Web mashup prevedono la creazione di nuove applicazioni da sorgenti eterogenee come servizi web SOAP/WSDL/RESTful, librerie JavaScript, contenuti catturati da pagine HTML. Le tecnologie di mashup sono state proposte per essere più aderenti ai bisogni degli utenti finali, ma questo obiettivo si è subito rivelato irrealistico perché i linguaggi di composizione adottati non sono attualmente aderenti a questi bisogni. Probabilmente ciò è avvenuto perché la ricerca sui mashup si è concentrata inizialmente sulle tecnologie e gli standard tecnologici da usare, dedicando poca attenzione all'adeguatezza verso l'utente delle tecniche di mashup, che spesso richiedono la programmazione manuale per l'integrazione dei servizi. Alcuni tools di mashup, chiamati mashup maker, forniscono un'interfaccia grafica per combinare più facilmente servizi, come in [7, 8, 17, 18]. Tuttavia, in una certa misura essi richiedono ancora una comprensione della logica d'integrazione (ad esempio flusso di dati, uso dei parametri, ecc). Studi recenti hanno mostrato che, anche se alcune piattaforme di rilievo (ad esempio Yahoo! PipesTM) semplificano lo sviluppo dei mashup, sono ancora difficili da usare per gli utenti finali [14, 19]. Differenti aspetti della creazione di spazi di lavoro interattivi devono essere studiati in profondità per meglio adattare le piattaforme ai bisogni degli utenti finali. Un aspetto importante riguarda la *personalizzazione*, poiché le piattaforme di composizione proposte fino ad adesso hanno tentato di essere generali, cioè indipendenti dal dominio [7]. La generalità spesso contrasta con l'usabilità. Noi siamo invece interessati a identificare meccanismi che consentano l'adattamento della piattaforma a specifici utenti e specifici contenuti.

5) Stato dell'arte

Negli anni '80, con l'avvento dei pc ad interfaccia grafica sul mercato di massa e dei tool per la produttività aziendale e personale si è avvertito presto il bisogno di creare linguaggi e/o ambienti che consentissero a utenti non esperti di programmazione di creare le proprie soluzioni software.

Un approccio che ha ricevuto grande attenzione è stata la programmazione visuale. Nel 1984 si svolge a Hiroshima, in Giappone, il primo convegno internazionale sui linguaggi visuali, che dal 1986 si svolge ogni anno. L'idea è di sfruttare la rappresentazione grafica bi-dimensionale per

facilitare lo sviluppo. In generale, questa tendenza ha avuto successo grazie alla semplificazione dello sviluppo, anche se questo non si è sempre tradotto realmente nella possibilità da parte di qualsiasi utente finale di sviluppare le proprie applicazioni. Nella programmazione visuale, i linguaggi di visualizzazione del flusso dei dati sono un paradigma che è stato spesso applicato: l'idea di base è di associare icone a funzionalità di alto livello che sono importanti per l'esperto dello specifico dominio. Tipicamente queste icone hanno porte di input e di output che rappresentano i dati in ingresso e in uscita; lo sviluppo dell'applicazione consiste quindi nella composizione interattiva delle istanze di queste icone che avviene indicando dove queste ricevono l'input e dove loro inviano i risultati del loro processo. Uno dei primi ambienti che applicò questo paradigma fu Cantata [15], un linguaggio visuale per Khoros, un ambiente che era usato principalmente per l'elaborazione delle immagini.

Un altro approccio considerò la programmazione guidata dagli esempi (Programming-by-example, PBE), apparsa per la prima volta intorno alla metà degli anni '80. L'utente crea un'applicazione eseguendo i passi di programmazione fornendo degli esempi: l'utente mostra un esempio di ciò che dovrebbe fare il programma e da qui l'ambiente di programmazione inferisce un'applicazione quanto più generale possibile, capace di implementare il comportamento desiderato.

Oggi, l'innovazione tecnologia spinge gli utenti finali a evolvere dal tradizionale ruolo di passivo consumatore passivo di informazioni a un ruolo più attivo. Gli utenti richiedono di modellare i software che utilizzano per adattarli ai loro bisogni, compiti e abitudini manipolando e personalizzando gli artefatti software. End-user Development e End-User Software Engineering sono termini conati per indicare queste attività degli utenti [5]. Compiti che sono tradizionalmente eseguiti dagli sviluppatori professionisti del software sono così trasferiti agli utenti finali, che diventano co-progettisti degli strumenti e dei prodotti che loro utilizzeranno. Gli utenti devono però essere supportati in questo nuovo ruolo di co-progettista. Alcuni autori sottolineano che tutti gli stakeholders di un sistema interattivo, inclusi gli utenti finali, sono 'proprietari' di una parte del problema: gli ingegneri del software conoscono la tecnologia, gli esperti di interazione uomo-macchina conoscono i fattori umani, i grafici sanno come creare soluzioni grafiche accattivanti e, infine, gli utenti finali conoscono il dominio applicativo [2]. Questi stakeholders devono contribuire alla progettazione unendo tutte le loro professionalità; per fare questo, hanno bisogno di utilizzare ambienti software specifici per le loro culture, conoscenze, abilità, attraverso i quali possono contribuire alla 'modellazione' del software.

La progettazione di sistemi che consentano attività di EUD richiede un nuovo paradigma di progettazione, detto meta-design, caratterizzato da due fasi principali [6]. La prima fase consiste nella creazione, da parte di sviluppatori professionisti, degli ambienti di progettazione che

consentano ai diversi stakeholders di partecipare alla progettazione (fase di meta-design). La seconda fase (fase di design) consiste nella progettazione del sistema finale, portata a termine dal lavoro congiunto dei vari stakeholders, che collaborano attraverso i loro ambienti di progettazione. Gli sviluppatori professionisti diventano quindi meta-designers, che creano gli ambienti software grazie ai quali gli altri stakeholders possono essere creativi e adattare il software ai loro bisogni. In [2], tali ambienti software sono chiamati *Software Shaping Workshops* (SSWs) proprio per indicare che sono laboratori virtuali (workshops) in cui i vari stakeholders modellano (shape) il software.

Il termine meta-design e' stato introdotto da Fisher et. al. [11]. In [9], Fischer si riferisce al modello SSW proposto in [2] come uno dei modelli più significativi per l'EUD, menzionando anche il modello Hive-Mind Space (HMS) (un'evoluzione del modello SSW) [20] e, ovviamente, il modello da lui proposto, denominato SER (Seeding, Evolutionary and Reseeding) [10].

Passando ora ai modelli di composizione, negli ultimi anni diversi tool hanno fornito interfacce grafiche per il collegamento di componenti, come avvenuto in Yahoo! Pipes. Questo tool fornisce meccanismi di composizione per aggregare, manipolare e fare mashup di contenuti presi dal web.

Grandi aziende come Amazon, Google, Yahoo!, IBM, HP, Sun Microsystems, SAP, Apple hanno capito che il loro futuro su Internet è fortemente legato all'adozione di una serie di principi aziendali di base [1], come ad esempio l'offerta di software come servizi (SaaS), assicurando che tali servizi vengano eseguiti in modo efficiente nel cloud e possano anche essere utilizzati per soddisfare le necessita della cosiddetta *Long Tail* [4].

Per questo motivo, queste aziende hanno iniziato a rendere disponibili componenti e strumenti che coprono alcune delle premesse dell'EUD descritte in precedenza. Queste componenti consentono a milioni di persone di utilizzare repository di servizi come in ProgrammableWeb.com [11], una piattaforma per la ricerca e la combinazione di servizi per comporre soluzioni per l'utente finale. Tutto ciò da' un'idea dell'interesse ad espandere ulteriormente il target di riferimento in grado di sfruttare l'ecosistema di servizi user-centered che molte aziende stanno producendo (come il Google Chrome Web Store [13]). A dispetto di quanto affermato, purtroppo queste piattaforme non sono ancora effettivamente utilizzabili da utenti non programmatori.

Tra le sfide emergenti nel mondo dei mashup c'è sia la composizione di contenuti e servizi utilizzando diversi dispositivi, sia l'utilizzo di queste composizioni su dispositivi diversi rispetto a quelli su cui sono stati realizzati. Infatti, in questi ultimi anni una forte innovazione e diversificazione della tecnologia sta caratterizzando l'esperienza dell'utente; un esempio sono gli smartphone e tablet che nel 2012 hanno pareggiato e superato i normali pc (<http://blog.mixpanel.com/>) sia come numero che come utilizzo. Altra tecnologia in forte ascesa nel mondo dei sistemi ubiqui sono i display interattivi di grandi dimensioni [16]: questi schermi spesso

sono installati in luoghi pubblici e si prestano bene all'interazione con l'utente tramite tocco ma anche tramite altri dispositivi. Nell'ottica della realizzazione di sistemi ubiqui, il mashup e l'EUD possono ricoprire un ruolo molto importante poiché uno stesso spazio di lavoro interattivo creato da un utente potrebbe essere utilizzato su diversi dispositivi come, ad esempio, i grandi display multitouch. Questa visione apre nuove sfide che riguardano il disaccoppiamento delle composizioni dall'interfaccia utente, ma soprattutto dai dispositivi su cui possono essere utilizzati. Partendo dalla mia esperienza sui display multitouch di grandi dimensioni, studierò in particolare l'utilizzo dei display interattivi con le piattaforme di mashup. In questa direzione, assieme al mio supervisore e ad altri collaboratori abbiamo cominciato a studiare soluzioni e approcci possibili [3]. L'obiettivo su cui concentrerò il mio lavoro di ricerca sarà di consentire all'utente di realizzare mashup di contenuti e servizi da poter utilizzare non solo su piattaforme web fruibili da dispositivi desktop o portatili, ma anche su ampi display multitouch.

I display multitouch rappresentano una tecnologia profondamente diversa dai normali pc desktop o dai dispositivi portatili. Questi, infatti, sono caratterizzati da una forte intuitività e semplicità d'uso grazie alla possibilità di interagire in maniera 'istintiva' e naturale attraverso l'interazione diretta delle mani sullo schermo per manipolare gli elementi dell'interfaccia grafica tramite gesti di traslazione, zoom, rotazione, etc. Purtroppo però, nonostante il forte "appeal" di questa tecnologia e la semplicità d'uso, risulta oggi ancora poco sfruttata dal momento che sia sul mercato che nella ricerca sono stati quasi sempre sviluppati sistemi software semplici e destinati al mero intrattenimento. Solo in questi ultimi anni si stanno intravedendo i primi sistemi software complessi e addirittura i primi sistemi operativi progettati per sfruttare l'interazione tattile. Recentemente sto facendo un'analisi accurata della letteratura scientifica prodotta su display di grandi dimensioni. A oggi, ho considerato oltre 160 lavori di vario genere che hanno portato all'individuazione di una tassonomia e una serie di challenges. Sono state definite 5 dimensioni di classificazione di questi sistemi con schermi interattivi: 1) *display set-up*, che riguarda l'installazione del display che può essere verticale, orizzontale, a pavimento, etc. 2) *visualization technology*, che concerne la tecnologia utilizzata per visualizzare il video del display 3) *type of application*, che riporta le tipologie di applicazioni emergenti su questa tecnologia 4) *location*, che riguarda i posti dove più di frequente sono installati questi schermi 5) *interaction modality*, che invece riguarda le modalità con cui gli utenti possono interagire con questi schermi (es. tocchi, movimenti del corpo, etc.).

6) Approccio al problema

Il mio lavoro di ricerca prevede la combinazione dei principi e tecniche di EUD con le potenzialità dei modelli e tecnologie di mashup, con particolare attenzione all'impiego di grandi display

interattivi. Questo dovrebbe portare alla definizione di meccanismi di composizione intuitivi e facili da usare. All'interno di un progetto già avviato all'interno del laboratorio di ricerca coordinato dal mio supervisore, partendo da una piattaforma generica di mashup sviluppata dai ricercatori del politecnico di Milano con cui stiamo collaborando, stiamo studiando come poterla adattare rispetto ai requisiti specifici identificati all'interno di una comunità d'utenti. In particolare, abbiamo già sviluppato un primo prototipo nel contesto dei beni culturali implementando specifiche versioni della piattaforma su tablet e grandi display multitouch [3].

La nuova piattaforma che stiamo sviluppando consente a un utente di accedere a servizi e risorse disponibili sul web e di creare spazi di lavoro interattivi dove essi compongono questi servizi e risorse per supportare specifici bisogni. La creazione di questi spazi interattivi propone un paradigma per l'integrazione delle risorse adatto a utenti che non hanno conoscenze tecniche. Una delle idee innovative alla base dell'approccio proposto è di sostituire applicazioni preconfezionate con ambienti di composizione che, grazie alla separazione tra i dati, funzioni e loro rappresentazioni, fa emergere gli ambienti interattivi a run-time sulla base delle azioni di composizione eseguite dagli utenti finali [3, 12].

La piattaforma fornisce meccanismi visuali intuitivi che consentono agli utenti finali di integrare facilmente contenuti eterogenei senza bisogno di programmare o l'uso di complesse notazioni di progettazione. L'attuale paradigma d'interazione è basato su drag&drop di contenuti estratti da servizi di terze parti o da sorgenti di dati personali, da muovere dentro "template" predefiniti di interfacce grafiche.

7) Ricadute applicative

L'idea principale dell'EUD è di consentire a qualsiasi utente, soprattutto a chi non ha competenze di programmazione, di creare le proprie applicazioni, consentendogli di non rimanere più all'interno dei confini 'imposti' dalle applicazioni preconfezionate. Questa esigenza è trasversale a ogni tipologia di utente e ogni dominio, poiché ogni utente desidera avere ambienti software interattivi capaci di essere modellabili in base alle proprie necessità. Il mio lavoro di ricerca prevede anche una parte più applicativa, relativa alla creazione di una piattaforma che consenta all'utente di accedere a servizi di terze parti, a contenuti propri o remoti, e a composizioni già realizzate per creare composizioni che meglio soddisfino i propri bisogni. Tale piattaforma può essere vantaggiosa in vari domini. In questa prima fase della ricerca stiamo considerando la sua adozione nel dominio dei beni culturali, in un contesto di supporto alle visite di luoghi di interesse culturale, ad esempio parchi archeologici. La stessa piattaforma potrebbe essere adattata per supportare il flusso di informazioni tra differenti stakeholders nello svolgimento di attività collaborative come,

ad esempio, nel contesto dell'e-health, dove la creazione di spazi di lavoro interattivi può facilitare la comunicazione e la collaborazione tra differenti figure, quali medici, infermieri, pazienti e i loro familiari, coinvolte nella cura di particolari patologie. Altro contesto che prenderemo in esame può essere quello delle aziende, dove la piattaforma potrebbe essere personalizzata per essere utilizzata all'interno del progetto VINCENTE.

8) Riferimenti bibliografici

- [1] Anderson, C. The long tail : why the future of business is selling less of more. Hyperion, 2006.
- [2] Ardito, C., Buono, P., Costabile, M. F., Lanzilotti, R. and Piccinno, A. 2012. End users as co-designers of their own tools and products. J. Vis. Lang. Comput., 23, 2 (2012), 78-90.
- [3] Ardito, C., Costabile, M. F., Desolda, G., Matera, M., Piccinno, A. and Picozzi, M. 2012. Composition of situational interactive spaces by end users: a case for cultural heritage. In Proceedings of the Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design (Copenhagen, Denmark, 2012). ACM, 2399029, 79-88.
- [4] Burnett, M., Atwood, J., Djang, R., Gottfried, H., Reichwein, J. and Yang, S. Forms/3: A First-Order Visual Language to Explore the Boundaries of the Spreadsheet Paradigm. City, 2001.
- [5] Costabile, M. F., Fogli, D., Fresta, G., Mussio, P. and Piccinno, A. 2003. Building environments for end-user development and tailoring. In Proceedings of the Human Centric Computing Languages and Environments, 2003. Proceedings. 2003 IEEE Symposium on (28-31 Oct. 2003, 2003), 31-38.
- [6] Costabile, M. F., Fogli, D., Mussio, P. and Piccinno, A. 2007. Visual Interactive Systems for End-User Development: A Model-Based Design Methodology. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 37, 6 (2007), 1029-1046.
- [7] Daniel, F., Casati, F., Benatallah, B. and Shan, M.-C. 2009. Hosted Universal Composition: Models, Languages and Infrastructure in mashArt. In Proceedings of the Proceedings of the 28th International Conference on Conceptual Modeling (Gramado, Brazil, 2009). Springer-Verlag, 1694213, 428-443.
- [8] Ennals, R., Brewer, E., Garofalakis, M., Shadle, M. and Gandhi, P. 2007. Intel Mash Maker: join the web. SIGMOD Rec., 36, 4 (2007), 27-33.
- [9] Fischer, G. End-User Development: From Creating Technologies to Transforming Cultures., 2013.

- [10] Fischer, G. 1998. Seeding, Evolutionary Growth and Reseeding: Constructing, Capturing and Evolving Knowledge in Domain-Oriented Design Environments. *Automated Software Engg.*, 5, 4 (1998), 447-464.
- [11] Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G. and Mehandjiev, N. 2004. Meta-design: a manifesto for end-user development. *Commun. ACM*, 47, 9 (2004), 33-37.
- [12] Latzina, M. and Beringer, J. 2012. Transformative user experience: beyond packaged design. *interactions*, 19, 2 (2012), 30-33.
- [13] Myers, B., Ko, A. and Burnett, M. 2006. Invited research overview: end-user programming. In *Proceedings of the CHI '06: CHI '06 extended abstracts on Human factors in computing systems* (Montreal, Québec, Canada, 2006). ACM, 75-80.
- [14] Namoun, A., Nestler, T. and Angeli, A. D. 2010. Conceptual and usability issues in the composable web of software services. In *Proceedings of the Proceedings of the 10th international conference on Current trends in web engineering* (Vienna, Austria, 2010). Springer-Verlag, 1927270, 396-407.
- [15] Rasure, J. R. and Williams, C. S. 1991. An integrated data flow visual language and software development environment. *J. Vis. Lang. Comput.*, 2, 3 (1991), 217-246.
- [16] Sebastian, B. *Making public displays interactive everywhere*. City, 2013.
- [17] Wong, J. and Hong, J. I. 2007. Making mashups with marmite: towards end-user programming for the web. In *Proceedings of the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA, 2007). ACM, 1240842, 1435-1444.
- [18] Yahoo! Inc.
- [19] Zang, N. and Rosson, M. B. 2008. What's in a mashup? And why? Studying the perceptions of web-active end users. In *Proceedings of the Proceedings of the 2008 IEEE Symposium on Visual Languages and Human-Centric Computing* (2008). IEEE Computer Society, 1550043, 31-38.
- [20] Zhu, L. 2011. Cultivating collaborative design: design for evolution. In *Proceedings of the Proceedings of the Second Conference on Creativity and Innovation in Design* (Eindhoven, Netherlands, 2011). ACM, 2079253, 255-266.

9) Fasi del progetto:

Il mio progetto di ricerca mira a individuare un paradigma di composizione leggero e flessibile da implementare in una piattaforma che consenta a utenti non tecnici di creare spazi di lavoro interattivi capaci di soddisfare i loro bisogni. Con tale piattaforma l'utente finale potrà creare in maniera semplice e intuitiva artefatti software partendo dalla composizione di contenuti e servizi

messi a disposizione da altri utenti, consentendone la visualizzazione su diversi dispositivi. In particolare, in ordine temporale, il lavoro sarà organizzato secondo le seguenti fasi e sotto-fasi:

- 1) Studio approfondito dello stato dell'arte (I anno di dottorato)
 - a. Studio dei principali paradigmi di composizione e delle piattaforme di mashup orientate agli utenti non tecnici.
 - b. Rassegna dei sistemi che utilizzano i display interattivi di grandi dimensioni.
- 2) Sviluppo di un primo prototipo di piattaforma di composizione (I anno di dottorato)
 - a. Personalizzazione del prototipo in un contesto dei beni culturali, quale supporto a visite di parchi archeologici.
 - b. Valutazione formativa del prototipo, anche con il coinvolgimento degli utenti finali.
- 3) Studio e implementazione di tecniche di collaborazione per la creazione di spazi di lavoro interattivi (I e II anno)
- 4) Adattamento della piattaforma ad altri domini applicativi, in particolare per supportare la collaborazione all'interno di aziende, ad esempio nell'ambito del progetto VINCENTE (I e II anno)
 - a. Valutazione formativa del prototipo, anche con il coinvolgimento degli utenti finali.
- 5) Revisione del prototipo della piattaforma (III anno)
 - a. Realizzazione di un successivo prototipo della piattaforma di composizione che implementi meccanismi di collaborazione e di personalizzazione a vari domini applicativi.
 - b. Valutazione sperimentale del nuovo prototipo

I prototipi realizzati e gli articoli scientifici che riporteranno i risultati della ricerca costituiranno i risultati miliari del mio lavoro.

10) Valutazione dei risultati

Il mio lavoro di tesi ha intente definire modelli e tecniche di EUD per consentire ad utenti non tecnici la composizione di servizi e risorse disponibili sul Web. Per valutare i prototipi che si svilupperanno sono previste valutazioni sia di tipo formativo che di tipo sommativo, coinvolgendo utenti reali. Tali valutazioni saranno effettuate mediante triangolarizzazione di metodi quali osservazioni di utenti sul campo, test con utenti, interviste e/o questionari. L'obiettivo è di valutare essenzialmente l'usabilità dei prototipi sviluppati e la *user experience* complessiva. A tale scopo metriche usualmente considerate sono le seguenti:

- *Tasso di completamento dei task*: conosciuta anche come una delle metriche fondamentali dell'usabilità, il tasso di completamento dei task è una semplice misura d'usabilità comunemente di tipo binario (task completamento / task non completato).
- *Durata d'esecuzione dei task*: è la misura di fatto di efficienza e produttività. Si registra il tempo necessario a ogni utente per completare un compito.
- *Problemi d'usabilità*: descrive i problemi d'usabilità emersi nell'interazione degli utenti con il sistema in termine di quante volte si sono verificati e la loro severità.
- *Esperienza d'uso complessiva*: misura il livello di soddisfazione e di coinvolgimento e apprezzamento degli utenti nell'uso del sistema.

L'originalità, l'innovatività e la validità dei risultati ottenuti nella ricerca svolta durante il dottorato saranno ovviamente valutabili tenendo conto della rilevanza delle sedi di pubblicazioni in cui appariranno gli articoli che riporteranno tali risultati.

11) Referenti esterni al Dipartimento

- Carla Simone, professore ordinario presso il dipartimento di informatica, sistemistica e comunicazione dell'Università degli studi di Milano-Bicocca
(<http://www.unimib.it/go/888889012/Home/Italiano/Elenco-Docenti/SIMONE-CARLA-dipartimento-di-informatica-sistemistica-e-comunicazione>)
- Nikolay Mehandjiev, professore di Enterprise Information Systems al Centre for Service Research, della Manchester Business School
(<http://www.mbs.ac.uk/research/people/profiles/nmehandjiev>)