

## CURRICULUM

Nome: Marino Segnan  
Indirizzo: via Palmieri 57, 10138 TORINO- tel. 011/433.5354  
e-mail: marino@di.unito.it  
sito www: <http://www.di.unito.it/~marino>  
Luogo e data di nascita: Torino, 05/04/1959

### Fatti essenziali:

• Laurea in Scienze dell'Informazione (Università di Torino; indirizzo: Teorico; votazione: 110/110 e Lode; data: 15/03/1983)

- Ingegnere del Software presso la Olivetti Advanced Technology Center, Cupertino, USA e Olivetti, S.p.A., Ivrea, (1983/89)
- Responsabile tecnico presso la L.P.S., Torino, (1989/92)
- Software Engineer presso la Sun Microsystem Corp., Mountain View, USA (1992/93)
- Software Engineer presso la Western Software Lab, Digital Equipment Corp., Palo Alto, USA (1993/94)
- Software Engineer presso la Sun Microsystem Corp., Mountain View, USA (1994/96)
- Collaborazione di ricerca con il Dipartimento di Informatica (Università di Torino; 1999/2003)
- Assegno di ricerca presso il Dipartimento di Informatica (Università di Torino; dal 2002)

### Formazione e attività di ricerca

Il mio curriculum riflette una lunga esperienza maturata in due ambiti diversi: ricerca e sviluppo in laboratori di industrie informatiche di avanguardia negli Stati Uniti e in Italia e la ricerca universitaria. Ho tentato dunque di realizzare una sintesi fra due esigenze diverse: da una parte l'interesse per l'attività di ricerca e la sperimentazione, dall'altra la consapevolezza che la medesima deve essere condotta utilizzando metodologie che consentano la valorizzazione delle risorse disponibili ed una loro organizzazione in un piano coerente e finalizzato. La mia attività ha avuto come tema centrale un interesse per lo sviluppo di architetture software innovative per l'accesso ad informazioni, con una particolare attenzione per l'interazione uomo-macchina.

Mi sono iscritto alla facoltà di Scienze dell' Informazione, scegliendo l'indirizzo teorico, associato ad un forte interesse per le tematiche di Intelligenza Artificiale. Mi sono laureato con il Professor Renato De Mori (attualmente all'Università di Avignone) realizzando un sistema software che permetteva di selezionare un piccolo sottoinsieme delle possibili parole pronunciate nel parlato continuo da un parlatore sconosciuto che utilizzasse un lessico di grandi dimensioni, basandosi sulle caratteristiche fonetiche delle sillabe della lingua italiana, i cui risultati sono stati pubblicati in [P1].

**Olivetti S.p.A.**

Dopo il conseguimento della laurea ho iniziato a lavorare presso la Olivetti di Ivrea come Software Engineer, nel Gruppo di Linguaggi e Compilatori.

Ho partecipato alla realizzazione di Dual, uno strumento di CASE per lo sviluppo e la documentazione di programmi scritti in Pascal+, una versione di Pascal utilizzata all'epoca da Olivetti per lo sviluppo del suo sistema operativo proprietario. La realizzazione è stata fatta nello stesso linguaggio in ambiente MOS. Dual conteneva diverse idee innovative tra cui una forma embrionale di rappresentazione ipertestuale per l'accesso alla documentazione degli elementi del codice.

Nel febbraio del 1987 ho raggiunto il gruppo di Intelligenza Artificiale, appena formato in Cupertino (USA). Ho partecipato alla progettazione e allo sviluppo di un *Intelligent Decision System*, AIDA, realizzato utilizzando tecniche di Intelligenza Artificiale e di Decision Analysis (Influence Diagram Theory). AIDA realizzava una analisi deterministica e probabilistica, con un modello di *risk assessment*.

Il progetto AIDA, sviluppato in collaborazione con il dipartimento di Economic Engineering System della **Stanford University**, ha portato anche alla realizzazione di AIE, un Expert System Shell, implementato in Smalltalk. Il sistema AIE, includeva Backward e Forward Chaining, Procedural Attachment e trattamento dell'incertezza. Io ho partecipato a questo progetto nelle fasi di design e implementazione. Lo strumento AIE è stato presentato a varie conferenze e seminari di Intelligenza Artificiale ed il suo funzionamento è stato dimostrato all'IJCAI nella sezione applicazioni industriali di Intelligenza Artificiale, tenutosi a Milano nel 1987.

Dal 1988 al 1989 ho collaborato alla realizzazione di un progetto congiunto tra il Banco di Napoli e l'Olivetti, per lo sviluppo di un Sistema Esperto per la valutazione di prestiti industriali. Tale sistema veniva realizzato utilizzando per la sua costruzione il sopra citato strumento AIE. A questo punto AIE era ormai in fase di utilizzo presso vari clienti dell'Olivetti, tra i quali l'ENEL e altri istituti bancari.

### **L.P.S. srl, Torino**

Dall'agosto 1989 all'aprile 1992, sono stato responsabile per il design e la implementazione dell'interfaccia utente di KeyOne, uno strumento di CASE per il supporto al disegno dettagliato e allo sviluppo del software. Lo strumento, partendo da una singola rappresentazione grammaticale ad alto livello, generava degli strumenti di gestione personalizzati ai diversi linguaggi di programmazione, che, utilizzando tecniche ipertestuali e metodi diretti da sintassi, associava informazioni in linguaggio naturale ("astrazioni") alla struttura del programma, facilitando in questo modo le fasi di progettazione e manutenzione. Il sistema utilizzava le librerie X Windows che permettevano una notevole indipendenza dal sistema operativo sottostante. Il sistema KeyOne aveva concorso come componente della stazione di sviluppo dell'IBM di Houston per la Space Station della NASA ed era stato poi adottato dall'Agenzia Spaziale Europea (ESA) per il progetto Columbus e utilizzato per lo sviluppo di software da diverse grandi aziende europee fra cui l'Alcatel e la Bull francese. Ho contemporaneamente realizzato ed insegnato corsi di Object-Oriented Programming in C++ .

### **Sun Microsystems, Mountain View, CA**

Dopo l'interessante esperienza di lavoro californiana presso il Centro di Ricerche dell'Olivetti di Cupertino, avevo maturato il desiderio di lavorare nel settore dello sviluppo di software di base di aziende leader dell'ICT. A seguito di precedenti contatti mi è stata offerta una posizione nel gruppo di linguaggi e compilatori della Sun Microsystem di Palo Alto, dove ho contribuito dal 1992 al '93 alla manutenzione ed ottimizzazione del compilatore C per la linea di macchine della Sun.

## **Digital Equipment Corporation , Palo Alto, CA**

Dopo l'esperienza di sviluppo presso la SUN, ho voluto fare un'esperienza di lavoro di ricerca in laboratori americani di ricerca applicata. Mi sono quindi spostato presso un centro che la Digital aveva a Palo Alto (1993-1994), dove si stava sviluppando un nuovo sistema di Workflow. All'interno di questo progetto, ho realizzato un' interfaccia grafica e studiato la possibilità di utilizzare componenti elementari e regole di composizione per la realizzazione di Workflow complessi.

## **Sun Microsystems, Mountain View, CA**

Nel giugno 1995 sono rientrato alla Sun, nel gruppo di ricerca Openstep, che stava realizzando un ambiente di sviluppo innovativo basato sul linguaggio Objective-C, precursore di Java. Objective-C si proponeva di fondere le caratteristiche di efficienza del linguaggio C con alcune delle funzionalità object-oriented del linguaggio Smalltalk. Il linguaggio era già stato usato per l'ambiente di sviluppo della workstation NeXt. Ho contribuito allo sviluppo del compilatore per la parte Object-Oriented ed allo sviluppo ed alla ottimizzazione del sistema di run-time-support per l'esecuzione dei programmi del linguaggio.

Questa lunga serie di esperienze a un livello industriale avanzato non aveva assopito un persistente personale interesse per problematiche di ricerca e di sperimentazione e, poiché d'altro canto esigenze di natura familiare richiedevano il mio rientro in Italia e l'industria informatica italiana non offriva un terreno favorevole alla ricerca e sviluppo, ho pensato di applicare la mia esperienza in un contesto universitario, cercando di coniugare innovazione e ricerca da un lato con una gestione razionale e una implementazione avanzata del progetto dall'altro. D'altronde le competenze acquisite presso la Digital e la Sun americana nella realizzazione di architetture software avanzate non sono oggi di facile acquisizione nel contesto italiano e si sono rivelate essenziali nella realizzazione di alcuni progetti di ricerca (vedi oltre, ad esempio, il Cantiere Multimediale).

## **Cantiere Multimediale**

Da maggio 1999 ad aprile 2000 ho lavorato nel **progetto Servizi Telematici Adattativi** (SeTA), presso il Dipartimento di Informatica. Tale progetto è parte dell'iniziativa di Telecom-Italia *Cantieri Multimediali* ed ha portato allo sviluppo di un sistema adattativo per il commercio elettronico. Nell'ambito di tale progetto la mia attività di ricerca si è focalizzata sulle **architetture multi-agente**. Il tipo di problematica che dovevamo affrontare ci ha portato alla definizione di una architettura multi-agente che permettesse di far collaborare agenti con ruoli diversi realizzati utilizzando diverse metodologie.

La fase iniziale della ricerca su sistemi multi-agente è stata orientata alla definizione di architetture innovative per sistemi multi-agente, sfruttando infrastrutture già disponibili in letteratura. In seguito, si è però notato che tali infrastrutture soffrono di vari tipi di limitazione, che rendono difficile la loro applicabilità ai sistemi Web adattativi. Per esempio, molte infrastrutture sono pensate per distribuire le attività di sistema tra copie multiple di uno stesso tipo di agente, sulla base della disponibilità e del carico di lavoro che tali agenti stanno portando avanti. Tuttavia, in un sistema che integra funzionalità complesse, gli agenti devono essere fortemente specializzati e non sono tra loro intercambiabili. Dunque, la coordinazione delle loro attività deve essere basata su principi diversi da quelli che hanno guidato lo sviluppo di tali infrastrutture per agenti "general-purpose".

Un altro tipo di problema nasce dalla necessità di supportare una forte configurabilità delle funzionalità offerte dai vari agenti, in modo da permettere di adattare facilmente tali agenti ai requisiti di un particolare dominio applicativo. Per esempio, nello sviluppo di un strumento per la costruzione di sistemi Web adattativi, si può considerare una gerarchia di agenti di modellizzazione dell'utente che offrono funzionalità di modellizzazione stereotipale, o dinamica, sulla base del comportamento dell'utente. Quale sia la funzionalità più adatta ad un particolare dominio applicativo dipende dai requisiti del dominio stesso; pertanto, è fondamentale permettere di configurare lo strumento per ogni particolare dominio, scegliendo le funzionalità appropriate con agevolezza.

Nell'ambito del progetto SeTA, ho progettato e realizzato l'architettura del sistema, coordinando inoltre il team di sviluppo. Durante la fase di realizzazione del prototipo, sono state fondamentali, sia per la definizione degli agenti che della loro comunicazione, le metodologie di Object Oriented Design, che avevo già utilizzato durante l'attività di progettazione nei laboratori della Digital e Sun.

L'architettura software è qui concepita come un sistema basato su tre livelli di funzionalità, modello che si è affermato anche a livello industriale sotto il nome di "Three Tier architecture"; ho inoltre adottato linguaggi e strumenti disponibili su tutte le architetture hardware. La risultante applicazione distribuita è implementata in linguaggio Java e usa gli strumenti di supporto ad esso associati (Java Servlet e Applet, RMI, JDBC), che in seguito sono diventati standard nella costruzione di applicazioni Web. Per quanto riguarda la piattaforma software per la realizzazione di agenti ho individuato in Voyager (della ObjectSpace) lo strumento che offriva la possibilità di realizzare lo scambio di messaggi tra agenti sia in modo sincrono che asincrono, permettendo ad alcuni dei nostri agenti di lavorare autonomamente, senza pregiudicare l'efficienza del dialogo del sistema con l'utente. Lo sviluppo di uno shell per la costruzione di sistemi di supporto alle vendite on-line costituisce il principale risultato applicativo del progetto. Particolare impegno è stato rivolto al rendere dichiarativa la conoscenza necessaria per istanziare il sistema su vari domini, anche attraverso la costruzione di strumenti per automatizzare l'acquisizione della conoscenza di dominio.

Una serie di articoli descrive il prototipo sviluppato nell'ambito del progetto SeTA sia dal punto di vista dell'approccio utilizzato che delle funzionalità realizzate: **[MC0]**, presentato allo *Spring Symposium* del AAI su *Adaptive User Interfaces*, a Stanford, nel 2000; **[CO0]**, presentato al Simposio su *Human-Computer Interaction*, a Firenze, nel 2001. Inoltre **[Le0]**, pubblicato su *Lecture Notes in Artificial Intelligence* nel 2000, rappresenta una sintesi delle scelte architettoniche e delle funzionalità di personalizzazione del prototipo. Il **[JR0]** pubblicato su *Communications of the ACM, Special Issue "The Adaptive Web"* rappresenta una sintesi di questi approcci.

Alcuni articoli descrivono in maggior dettaglio il progredire dell'architettura software del prototipo: **[CO1]**, presentato sempre ad *Agents*, a Barcellona, nel 2000; **[CO2]**, presentato anch'esso ad *Agents*, a Montreal, nel 2001.

In particolare, **[MC0]** mostra come sia l'architettura che le funzionalità dei singoli moduli rappresentino un significativo avanzamento rispetto allo stato dell'arte dei sistemi adattativi di commercio elettronico on-line, integrando tecniche avanzate di Intelligenza Artificiale in un'applicazione concreta.

## **Collaborazione di ricerca con il Dipartimento di Informatica**

L'esperienza maturata nell'ambito del progetto SeTA e l'interesse suscitato a livello nazionale e internazionale ha portato a raggruppare alcune delle persone che vi avevano partecipato nel gruppo di ricerca su *Intelligent User Interfaces* ([www.di.unito.it/~seta](http://www.di.unito.it/~seta)), attivo presso il Dip. di Informatica dell'Università di Torino. Un primo risultato è stata la realizzazione di un prototipo per la gestione di servizi finanziari personalizzati su Web, all'interno di una collaborazione stabilita con l'Istituto Bancario SanPaolo Imi durante il 2000, del quale ho seguito lo sviluppo per quanto riguarda l'utilizzo del nostro shell per la costruzione di negozi personalizzati su Web.

La continuazione della ricerca sulle infrastrutture multi-agente per la costruzione di sistemi Web personalizzati, ha portato, nell'ambito del progetto di interesse nazionale cofinanziato dal MURST dal titolo "Agenti Intelligenti: Interazione e acquisizione di conoscenza", allo sviluppo di Seta2000, un'infrastruttura per la costruzione di sistemi multi-agente che operano sul Web. L'architettura generale di Seta2000 rappresenta un'evoluzione di SeTA: anche in SeTA2000 agenti specializzati si coordinano per gestire l'interazione con l'utente. Tuttavia, gli agenti hanno una diversa architettura interna: essi sono organizzati come agenti autonomi, le cui attività sono descritte mediante l'uso di un formalismo ad azioni che ne massimizza la configurabilità e l'estensibilità.

Le attività di ciascun agente sono gestite da un interprete che permette di eseguire azioni in parallelo, soddisfacendo le richieste che vengono inviate all'agente in modo efficiente. L'interprete permette inoltre di scatenare attività interne, che permettono all'agente di far progredire attività autonome, indipendenti dai servizi richiesti dal resto del sistema. Come descritto in **[MC1]**, presentato all'IJCAI, a Seattle, nel 2001 (con versione ridotta apparsa **[CO2]**), l'interprete si discosta dai tradizionali motori per la gestione di agenti in quanto si basa sull'uso di thread che permettono l'esecuzione parallela di attività nell'ambito del processo principale dell'agente. Ogni richiesta che perviene all'agente può essere eseguita in parallelo (nell'ambito del processo principale), appena viene ricevuta, se le sue precondizioni ne rendono possibile l'esecuzione; altrimenti, viene sospesa, in attesa che lo stato interno dell'agente cambi e ne permetta l'esecuzione. Secondo lo stesso schema, l'agente porta avanti le attività interne, che gli permettono di offrire funzionalità speciali: per esempio, mentre servizi quali la generazione della prossima pagina da mostrare all'utente vengono attivati sulla base delle azioni dell'utente (quali click egli esegue sull'interfaccia del sistema), le attività di revisione dei modelli di utente sono gestite in modo autonomo da un agente Modellizzatore, durante l'interazione con l'utente, senza bisogno che uno scheduler le attivi in modo esplicito.

La rappresentazione ad azioni delle attività degli agenti ne supporta l'estensione, facilitando la manutenzione e la revisione dell'architettura generale. In particolare, oltre alla possibilità di estendere l'insieme di funzionalità offerte da un agente, tale rappresentazione supporta l'aggiornamento delle funzionalità stesse. Per esempio, il Dialog Manager gestisce il flusso dell'interazione con l'utente eseguendo un insieme di azioni che descrivono i tipi di "turno" che il sistema può eseguire nelle varie fasi dell'interazione, e sulla base del contesto precedente di interazione. Tali azioni implementano un automa a stati finiti che definisce le sequenze di turni ammissibili e può essere aggiornato senza difficoltà per inserire nuovi tipi di turno e nuove transizioni di stato. L'infrastruttura SeTA2000 è descritta in dettaglio in **[JR2]**.

All'interno del gruppo ho collaborato all'attività di ricerca e sviluppo nell'ambito di vari progetti, tra cui lo sviluppo del sistema INTRIGUE (INteractive TouRist Information GUIDE). Con il sistema INTRIGUE, si è anche sperimentato l'adattamento di cataloghi a dispositivi differenziati: le tecniche di generazione dell'interfaccia utente sono state recentemente estese per permettere un accesso da Web browser e WAP minibrowser. Si sono curati molto aspetti che riguardano l'utilizzazione di servizi in modalità locale al dispositivo mobile: l'idea di base è portare parte delle funzionalità del sistema su un dispositivo mobile per permettere all'utente di utilizzarle senza una connessione continua al server turistico remoto. A parte i dettagli tecnici che permettono l'accesso

locale alle funzionalità del sistema, sono interessanti aspetti di gestione del contesto dell'interazione con l'utente, che deve essere sincronizzato durante le varie fasi di interazione remota (che possono avvenire se l'utente desidera navigare il catalogo turistico in rete) e locale. Ho inoltre realizzato un configuratore ad hoc che, dati vincoli di orario e topologia delle attrazioni turistiche, fornisce un itinerario di visita che cerca di rispettare l'agenda dell'utente.

In particolare ho ridefinito parte dell'architettura software estendendola anche all'*ubiquitous computing* sia per quanto riguarda l'accesso del sistema da WAP, adattando la presentazione delle informazioni a schermi ridotti, che per la possibilità di utilizzare parte delle funzionalità del sistema senza interazione con il server (in locale al dispositivo palm o WAP) usando le possibilità offerte dagli strumenti Enterprise Java Beans, Java Servlet, XML, Java MicroEdition. Una descrizione di alcuni degli aspetti innovativi del sistema si trovano in: **[Le1]**, presentato al workshop su *Adaptive Hypertext and Hypermedia* (AH 2001), ad Aarhus (Danimarca), nel 2001; **[MC2]**, presentato alla conferenza dell'ACM *Hypertext*, anch'essa ad Aarhus (Denmark), nel 2001 e in **[Le2]** presentato a Malaga al *2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*. Una descrizione approfondita del sistema INTRIGUE è apparsa in *Applied Artificial Intelligence, Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries* **[JR1]**,

In seguito ho realizzato, all'interno dell'attività di ricerca di model-based diagnosis con il prof. Luca Console, un sistema Software per la simulazione basata su equazioni qualitative di sistemi complessi, con lo scopo di sviluppare sistemi di diagnosi che a fronte di situazioni anomale possano fornire una azione di recovery. Il sistema contiene due gruppi di funzionalità: da un lato definizione e costruzione dei componenti e del sistema, dall'altra molteplici possibilità di simulazione ed ispezione dello stato di esecuzione. Il sistema è scritto in Java e funziona su UNIX e Windows. I risultati del lavoro sono presentati in **[CO3]**.

Negli ultimi mesi di attività mi sono concentrato sullo studio della comunicazione con i Web Services, con l'obiettivo di migliorare le attuali possibilità di interazione tra service consumer e provider e di estendere la possibilità di integrazione dei servizi disponibili in rete in servizi personalizzati all'utente. In particolare, mi sono dedicato alla definizione di modelli di interazione che permettano la gestione di interazioni complesse e flessibili tra consumatore e produttore di servizi. In tale attività, mi sono basato sulla mia precedente esperienza nello sviluppo di tecniche per la gestione di sistemi multiagente e sulla ricerca esistente nel campo della comunicazione tra agenti, per definire un linguaggio di comunicazione con web services che offra le dovute garanzie di applicabilità a casi reali e scalabilità e, al tempo stesso, permetta la gestione di dialoghi ad iniziativa mista. L'estensione delle capacità comunicative dei servizi web è necessaria perché i Web Services attualmente sono soggetti ad alcune limitazioni che riducono la loro effettiva utilizzazione in servizi complessi. In particolare, gli standard attualmente disponibili, come WSDL, permettono di definire la specifica delle interfacce dei servizi offerti, ma non un flusso di interazione. Quindi, al momento, l'interazione tra clienti di Web Services e i fornitori di Web Services deve limitarsi a invocazioni di servizi semplici che non implicano turni di conversazione. La proposta di modello conversazionale a cui mi sono dedicato estende le capacità conversazionali dei Web Services, fornendo supporto ad interazioni complesse, dove più messaggi devono essere scambiati prima che il servizio sia completato. Il modello è stato definito prendendo, come punto di partenza, il modello teorico degli atti linguistici (Cohen,

Levesque, Rational Interaction as the basis for Communication, 1990 e J.R.Searle, Indirect Speech Acts, Syntax and Semantics, 1975) ed elaborando tale modello per permettere a due peer che interagiscono di:

- gestire interazioni complesse mantenendo i due peer "loosely coupled", senza cioè richiedere, come è tipico dei linguaggi di comunicazione per agenti, l'uso della stessa infrastruttura di comunicazione
- gestire un contesto di interazione semplificato, che permette ad un servizio di essere invocato da consumer più o meno dotati di capacità di gestione dell'interazione (da consumer ciechi, che portano avanti il dialogo secondo le istruzioni fornite via via dal provider, a consumer dotati di iniziativa propria, che interagiscono in modo proattivo)
- scambiare messaggi SOAP, per permettere massima interoperabilità

Una prima parte del lavoro apparirà in un capitolo di un libro edito dalla Kluwer sui Web Services **[CL1]**.

I più recenti risultati di questo lavoro sono stati presentati alla conferenza World Wide Web (WWW 04, NY, USA, Maggio 2004): Ardissono L., Petrone G., Segnan M., "A Framework for the Server-Side Management of Conversations with Web Services" **[MC3]**.

Una descrizione approfondita di questa apparirà in Ardissono L., Petrone G., Segnan M. A Conversational approach to the interaction with We Services Computational Intelligence **[JR3]**.

In **conclusione** il mio percorso di ricerca (in particolare negli ultimi anni) ha dimostrato come gli approcci utilizzati abbiano rappresentato una risposta ad alcuni importanti problemi aperti nell'ambito dei sistemi informativi. In particolare, i sistemi informativi basati su Web rappresentano una sfida in due direzioni: da un lato essi necessitano di architetture e infrastrutture flessibili e scalabili; dall'altro essi richiedono interfacce personalizzate, accessibili a utenti con esigenze e capacità molto eterogenee. Lo studio approfondito, la progettazione e la realizzazione di prototipi software basati su architetture multi-agente e su approcci knowledge-based alla modellazione dell'utente e alla generazione dinamica di interfacce personalizzate (inclusa la generazione linguistica) hanno rappresentato un contributo e una risposta concreta nelle due direzioni menzionate.

#### **Attività didattica in ambito industriale:**

- Durante l'attività presso la LPS nel 1991 ho preparato il materiale e tenuto un corso intensivo di una settimana, sulla programmazione Object-Oriented e Smalltalk per gli sviluppatori della Alfa Romeo di Pomigliano d'Arco.
- Nello stesso periodo ho preparato il materiale e tenuto un corso intensivo di una settimana, sulla programmazione C++ per giovani periti per conto della regione Lombardia a Milano.

#### **Pubblicazioni**

## Publicazioni su riviste

- [JR3] Ardissono L., Petrone G. and Segnan M. **A conversational approach to the interaction with Web Services**, *To appear on Computational Intelligence, special issue on Business Agents and the Semantic Web*
- [JR2] Ardissono L., Goy A., Petrone G., Segnan M. **"A multi-agent infrastructure for developing personalized Web-based systems"** *To appear on ACM Transaction on Internet Technologies*
- [JR1] Ardissono L., Goy A., Petrone G. and Segnan M. and Torasso P.  
**INTRIGUE: personalized recommendation of tourist attractions for desktop and handset devices**,  
*Applied Artificial Intelligence, Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries, pp. 687-714, vol. 17, September-October 2003.*
- [JR0] Ardissono L., Goy A., Petrone G. and Segnan M.  
**Personalization in Business-to-Consumer Interaction**  
*Communications of the ACM, Special Issue "The Adaptive Web", vol. 45, N. 5, pp. 52-53. ACM Press, May 2002.*

## Publicazioni su libri

- [CL1] Ardissono L., Petrone G., Segnan M.  
**Enabling Flexible Conversation with Web Services**  
*to appear on "Web Services and Agent-Based Engineering", Kluwer, 2004*

## Publicazioni su Lecture Notes

- [Le2] Ardissono L., Goy A., Petrone G., Segnan M., Torasso P.  
**Ubiquitous user assistance in a tourist information server**,  
*Lecture Notes in Computer Science n. 2347: Adaptive Hypermedia and Adaptive Web-Based Systems, 2002, pp. 14-23*
- [Le1] Ardissono L., Goy A., Petrone G., Segnan M., Torasso P.  
**Tailoring the recommendation of touristic information to heterogeneous user groups**,  
*Lecture Notes in Computer Science n. 2266: "Hypermedia: Openness, Structural Awareness, and Adaptivity" Int. Workshops OHS-7, SC-3 and AH-3, 2001, pp.280-295*
- [Le0] Ardissono L., Goy A., Petrone G., Segnan M., Console L., Lesmo L., Simone C., Torasso P.  
**Agent Technologies for the development of adaptive web stores**,  
*Lecture Notes in Artificial Intelligence, n. 1991: Agent Mediated Electronic Commerce. A European Perspective, Springer-Verlag, 2000, pp.197-216.*



## **Pubblicazioni ai principali convegni internazionali**

- [MC4] Ardissono L., Goy A., Petrone G., Segnan M.  
**Interaction with Web Services in the Adaptive Web,**  
*to appear on AH 2004, August 2004.*
- [MC3] Ardissono L., Petrone G., Segnan M.  
**A Framework for the Server-Side Management of Conversations with Web Services,**  
*Proc. WWW 2004, ACM Press, NY, USA, May 2004, p. 124-143.*
- [MC2] Ardissono L., Goy A., Petrone G., Segnan M., Torasso P.  
**Dynamic generation of personalized tourist information on the Web,**  
*Proc. 12th ACM Conference on Hypertext and Hypermedia (Hypertext '01), Aarhus (Denmark), August 2001.*
- [MC1] Ardissono L., Goy A., Petrone G., Segnan M.  
**A software architecture for dynamically generated adaptive Web stores,**  
*Proc. 17th Int. Joint Conf. on Artificial Intelligence (IJCAI'01), Seattle, August 2001, pp.1109-1114.*
- [MC0] Ardissono L., Goy A., Petrone G., Segnan M.  
**Adaptive user interfaces for on-line shopping,**  
*Proc. Adaptive User Interfaces Spring Symposium of AAAI, AAAI Press, Menlo Park, CA, March 2000, pp.13-18.*

## **Pubblicazioni a convegni internazionali**

- [CO3] Console L., Correndo G., Picardi C., Segnan M., Bray R., Buffo A., Cascio F., Marchesini P., **Tools for integrating Diagnosis in the design process. An application to the Common Rail air and fuel delivery Systems.**  
*4th Int. Conf. on Control and Diagnostics in Automotive Applications (CDAUTO03) Sestri Levante , Italy, June 2003.*
- [CO2] Ardissono L., Goy A., Petrone G., Segnan M.  
**Software architecture of SETA, an adaptive Web store shell,**  
*Proc. 5th Int. Conf. on Autonomous Agents (Agents'01), ACM Press, Montreal, June 2001, pp.214-215.*
- [CO1] Ardissono L., Goy A., Petrone G., Segnan M.  
**Configurability within a multi-agent Web store shell,**  
*Proc. 4th Int. Conf. on Autonomous Agents (Agents'00), ACM Press, Barcelona, June 2000, pp.146-147.*
- [CO0] Ardissono L., Goy A., Petrone G., Segnan M.  
**Strategie di personalizzazione per sistemi di commercio elettronico su Web,**  
*Proc. VII Congresso nazionale SIE, Firenze, September 2001, pp.66-70.*

## **Publicazioni precedenti**

[P1] De Mori R., Laface P., Petrone G., Segnan M.  
**Lexical Level in Continuous Speech Recognition,**  
*EUSIPCO 1983 pp. 427-430*

## **Tesi**

Laurea:

*Modelli di percezione della voce: la rete di accesso lessicale*, Università di Torino,  
1983.