

CURRICULUM VITAE ET STUDIORUM E  
DELL'ATTIVITÀ DIDATTICA E SCIENTIFICA  
DEL DOTT. FAUSTO FASANO

## **1 Generalità**

Il dott. Fausto Fasano é nato a Salerno il 20 Febbraio 1975 e risiede a Salerno (SA) alla via Pietro del Pezzo, 54/A.

## **2 Posizione militare**

É stato dispensato dal servizio di Leva in virtù del Decreto del Ministero della Difesa 13 marzo 2003 (Rideterminazione delle condizioni per la concessione della dispensa dagli obblighi di leva) lettera e), ai sensi dell'art. 7, comma 3, lettera a), del decreto legislativo 30 dicembre 1997.

## **3 Posizione di lavoro corrente**

Ricercatore universitario per il settore scientifico disciplinare INF/01 (Informatica) dal 30 Dicembre 2008 presso la Facoltà di Scienze Matematiche, Fisiche e Naturali dell'Università degli Studi del Molise.

## **4 Formazione e Borse di Studio**

### **4.1 Dottorato di Ricerca**

Dal 2003 al 2006 ha frequentato il V ciclo (nuova serie) del corso di Dottorato di Ricerca in "Informatica" presso l'Università degli Studi di Salerno.

Il 13 Aprile 2007 ha conseguito il titolo di Dottore di Ricerca discutendo una tesi dal titolo "Fine-grained Management of Software Artefacts".

### **4.2 Lauree**

Il 24 Marzo 2004 ha conseguito la Laurea in Informatica Specialistica presso l'Università degli Studi di Salerno, con votazione 110/110 e lode, discutendo una Tesi dal titolo "IL PROBLEMA DELLA TRACCIABILITÀ IN SISTEMI PER LA GESTIONE DI ARTEFATTI SOFTWARE".

Il 17 Luglio 2003 ha conseguito la Laurea in Informatica presso l'Università degli Studi di Salerno, con votazione 110/110 e lode, discutendo una Tesi dal titolo "STUDIO E REALIZZAZIONE DI UN SISTEMA PER LA GESTIONE DEGLI ARTEFATTI SOFTWARE".

### **4.3 Altra formazione**

Ha partecipato alla II, III, IV, V, e VI edizione della "International School on Software Engineering", tenutasi presso il Dipartimento di Matematica e Informatica

della Facoltà di Scienze MM. FF. e NN. dell'Università degli Studi di Salerno tra il 2005 ed il 2009.

#### **4.4 Borse di Studio**

Dal 4 Giugno 2007 al 29 Dicembre 2009 ha usufruito di una Borsa di studio per lo svolgimento di attività di ricerca post-dottorato (D.R. 1.3.2007 Rep. N.654), presso il Dipartimento di Matematica e Informatica dell'Università degli Studi di Salerno.

Dal 01 Novembre 2003 al 31 Ottobre 2006 ha usufruito di una borsa di studio, durante il periodo del Dottorato di Ricerca.

### **5 Partecipazione a Progetti di Ricerca**

Il dott. Fasano ha partecipato a diversi progetti di ricerca nazionali e di ateneo. Nell'ambito di alcuni progetti, ha ottenuto contratti per lo svolgimento di attività di ricerca.

#### **5.1 Progetti di Ricerca con Contratto**

Dal 2007 partecipa al progetto di ricerca "TOCAI.IT" (Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet) finanziato dal MiUR (Ministero dell'Università e della Ricerca) nell'ambito del programma FIRB (Fondo per gli Investimenti della Ricerca di Base), in qualità di membro dell'unità operativa del CINI (Consorzio Interuniversitario Nazionale per l'Informatica) Roma sull'Attività 8 "*Modelli e strumenti di cooperazione per l'integrazione di dati e servizi*". Dal 01 Dicembre 2006 al 30 Aprile 2007, nell'ambito del progetto "TOCAI.IT", ha ottenuto un contratto finalizzato allo studio di metodi e architetture per il supporto alla composizione automatica ed all'orchestrazione adattiva di servizi, in particolar modo mediante tecniche di workflow management, lo studio di metodi per l'integrazione di dati e servizi mediante la modellazione e lo sviluppo collaborativi di documenti strutturati, e lo studio di metodi di "natural language processing" e di "information retrieval" per l'integrazione di contenuti informativi non strutturati.

Nel 2006 ha partecipato al progetto "Migrazione di sistemi legacy verso infrastrutture web-based", in convenzione con l'azienda MTSYS srl e finanziato dalla Regione Campania, ai sensi della legge 598/94. Nell'ambito di tale progetto ha ottenuto un contratto per attività di ricerca finalizzato alla *Definizione e implementazione di una metodologia per la implementazione di interfacce utenti di sistemi di legacy verso il web*.

## 5.2 Progetti di Ricerca Nazionali

Dal 2007 al 2008 ha partecipato, in qualità di membro dell'unità operativa del Dipartimento di Matematica e Informatica della Facoltà di Scienze MM. FF. e NN. dell'Università degli Studi di Salerno, al progetto PRIN (Programmi di ricerca di Rilevante Interesse Nazionale) finanziato dal MiUR (Ministero dell'Università e della Ricerca) con concessione nr. 2006098097, dal titolo *“Metodi e tool per la migrazione di sistemi software verso il web e architetture service-oriented: valutazione sperimentale, usabilità e trasferimento tecnologico”* (titolo in inglese: *“MEthods and Tools for migrATING software systeMs towards web and service Ori-ented aRchitectures: exPerimental evaluation, usability, and technology tranSfer”*). La durata del progetto è stata di 24 mesi ed il coordinatore nazionale è stato il prof. Andrea De Lucia dell'Università di Salerno (sito web: <http://www.sesa.dmi.unisa.it/metamorphos>).

## 5.3 Progetti di Ricerca di Ateneo

Dal 2008 partecipa al progetto di ricerca di Ateneo finanziato del Ministero dell'Università e della Ricerca (ex MURST 60%), dal titolo *“Metodi e Strumenti per il lavoro cooperativo”*, presso il Dipartimento di Matematica e Informatica dell'Università degli Studi di Salerno. Coordinatrice del progetto: prof.ssa Genoveffa Tortora.

Nel 2007 ha partecipato al progetto di ricerca di Ateneo finanziato del Ministero dell'Università e della Ricerca (ex MURST 60%), dal titolo *“Ingegneria del software e studi empirici”*, presso il Dipartimento di Matematica e Informatica dell'Università degli Studi di Salerno. Coordinatrice del progetto: prof.ssa Genoveffa Tortora.

Nel 2006 ha partecipato al progetto di ricerca di Ateneo finanziato del Ministero dell'Istruzione, dell'Università e della Ricerca (ex MURST 60%), dal titolo *“Sistemi visuali ed interattivi”*, presso il Dipartimento di Matematica e Informatica dell'Università degli Studi di Salerno. Coordinatrice del progetto: prof.ssa Genoveffa Tortora.

Nel 2005 ha partecipato al progetto di ricerca di Ateneo finanziato del Ministero dell'Istruzione, dell'Università e della Ricerca (ex MURST 60%), dal titolo *“Web Engineering: metodologie e applicazioni”*, presso il Dipartimento di Matematica e Informatica dell'Università degli Studi di Salerno. Coordinatrice del progetto: prof.ssa Genoveffa Tortora.

Nel 2004 ha partecipato al progetto di ricerca di Ateneo finanziato del Ministero dell'Istruzione, dell'Università e della Ricerca (ex MURST 60%), dal titolo *“Tracciabilità di artefatti software in processi evolutivi”*, presso il Dipartimento di Matematica e Informatica dell'Università degli Studi di Salerno. Coordinatrice del

progetto: prof.ssa Genoveffa Tortora.

Nel 2003 ha partecipato al progetto di ricerca di Ateneo finanziato del Ministero dell'Istruzione, dell'Università e della Ricerca (ex MURST 60%), dal titolo "*Un sistema di supporto alla gestione di processi software basato su artefatti*", presso il Dipartimento di Matematica e Informatica dell'Università degli Studi di Salerno. Coordinatrice del progetto: prof.ssa Genoveffa Tortora.

## **6 Attività organizzativa**

### **6.1 Organizzazione e coordinamento di gruppi di ricerca**

Ha partecipato alla realizzazione dell'X-lab, il Laboratorio di didattica avanzata e ricerca in Informatica e Calcolo Scientifico presso l'Università degli Studi del Molise (sito web: <http://distat.unimol.it/STAT/laboratori-1/laboratorio-di-informatica-e-calcolo-scientifico>).

Ha partecipato alla realizzazione del SE@SA Lab, il Laboratorio di didattica avanzata e ricerca in Ingegneria del Software presso l'Università degli Studi di Salerno (sito web: <http://www.sesa.dmi.unisa.it>).

Dal 2007 al 2009 é stato amministratore del Laboratorio di Ingegneria del Software presso l'Università degli Studi di Salerno, sotto la supervisione del prof. Andrea De Lucia.

### **6.2 Ruoli in comitati di programma e revisione di lavori scientifici**

Membro del comitato di programma dei seguente congressi e scuole di dottorato internazionali

- IADIS International Conference e-Society, edizioni 2007 (Lisbon, Portugal), 2008 (Algarve, Portugal), 2009 (Barcellona, Spagna) e 2010 (Porto, Portugal).
- CSEDU International Conference on Computer Supported Education, edizione 2009 (Lisbon, Portugal).
- Advanced Software Engineering & Its Applications, edizione del 2010 (Bali, Indonesia).
- International Summer School on Software Engineering, edizione del 2010 (Salerno, Italy).

Ha curato la recensione, in qualità di revisore, di lavori scientifici presentati alle seguenti riviste internazionali:

- Advances in Software Engineering
- International Journal on Software Tools for Technology Transfer.

Ha curato la recensione, in qualità di revisore aggiunto, di lavori scientifici presentati ai seguenti congressi internazionali:

- ACM/IEEE International Conference on Software Engineering, edizione del 2007 (Minneapolis, MN, USA).
- IEEE International Conference on Software Maintenance, edizioni del 2005 (Budapest, Ungheria), del 2006 (Philadelphia, Pennsylvania, USA), del 2007 (Parigi, Francia) e del 2009 (Edmonton, Canada).
- IEEE International Conference on Global Software Engineering, edizioni del 2006 (Florianopolis, Brasile), del 2008 (Bangalore, India) e del 2009 (Limerick, Ireland).
- European Conference on Software Maintenance and Reengineering, edizioni del 2005 (Manchester, UK), del 2006 (Bari, Italia), del 2007 (Amsterdam, Olanda), del 2008 (Atene, Grecia) e del 2009 (Kaiserslautern, Germania).
- IEEE International Conference on Program Comprehension, edizioni del 2005 (St. Louis, Missouri, USA) e del 2006 (Atene, Grecia).
- IEEE International Workshop on Source Code Analysis and Manipulation, edizioni del 2005 (Budapest, Ungheria) e del 2006 (Philadelphia, Pennsylvania, USA).
- IEEE International Working Conference on Source Code Analysis and Manipulation, edizione del 2007 (Paris, Francia) e del 2009 (Edmonton, Canada).
- IEEE Working Conference on Reverse Engineering, edizioni del 2006 (Benevento, Italy), del 2007 (Vancouver, BC, Canada), del 2008 (Antwerp, Belgio) e del 2009 (Lille, France).
- International Conference on Software Engineering and Knowledge Engineering, edizioni del 2006 (San Francisco, CA, USA) e del 2007 (Boston, USA).
- International Conference on Software and Data Technologies, edizione del 2006 (Porto, Portogallo).
- International Conference on Enterprise Information Systems, edizione del 2008 (Barcellona, Spagna), del 2009 (Milan, Italy) e del 2010 (Madeira, Portugal).
- International Symposium on Empirical Software Engineering and Measurement, edizione del 2008 (Kaiserslautern, Germany).

- International Conference on Distributed Multimedia Systems, edizioni del 2008 (Boston, USA) e 2009 (San Francisco, USA).
- IEEE World Congress on Computer Science and Information Engineering, edizione del 2009 (Los Angeles/Anaheim, USA).
- IEEE International Symposium on Web Systems Evolution, edizione del 2009 (Edmonton, Canada).

### **6.3 Attività organizzativa in scuole di dottorato e giovani ricercatori**

É stato membro della Segreteria Scientifica della II, III, IV e VI edizione della “International School on Software Engineering”, tenutasi presso l’Università degli Studi di Salerno negli anni 2005, 2006, 2007 e 2009.

É stato Finance Chair della V edizione della “International School on Software Engineering”, che si é tenuta presso l’Università degli Studi di Salerno dal 22 al 25 Settembre 2008.

## **7 Attività didattica e seminariale**

L’attività didattica e seminariale considerata dal dott. Fasano di grande interesse e di notevole valore formativo. In particolare, egli si é sempre dedicato all’attività di supporto agli studenti con impegno e disponibilità, valutando essenziale il ruolo del docente al fine di motivare opportunamente il discente e coinvolgerlo durante l’intero percorso formativo. Dal 2003 al 2007 ha svolto numerosi seminari ed esercitazioni per i corsi di Ingegneria del Software I, Ingegneria del Software II e Gestione dei Progetti Software, tenuti dal prof. De Lucia presso l’Università degli Studi di Salerno.

Dall’A.A. 2005-2006 cultore della materia per i corsi di Ingegneria del Software presso la Facoltà di Scienze MM. FF. e NN. dell’Università degli Studi di Salerno. Dall’A.A. 2006-2007 cultore della materia per il corso di Laboratorio di Ingegneria del Software presso la Facoltà di Scienze MM. FF. e NN. dell’Università degli Studi del Molise.

Dall’A.A. 2007-2008 cultore della materia per il corso di Linguaggi di Programmazione presso la Facoltà di Scienze MM. FF. e NN. dell’Università degli Studi del Molise.

Il dott. Fasano stato relatore e correlatore di numerose tesi di laurea nel settore dell’Ingegneria del Software.

## **7.1 Incarichi accademici presso l'Università del Molise**

Il dott. Fasano ha ricoperto incarichi accademici presso l'Università del Molise su corsi di Ingegneria del Software, di Linguaggi di Programmazione e Sistemi Operativi.

### **7.1.1 Corsi di Ingegneria del Software e Laboratorio**

Ha tenuto i seguenti corsi di "Ingegneria del Software e Laboratorio", esame del II anno (40 ore di lezione frontali per un totale di 5 CFU) presso la Facoltà di Scienze Matematiche, Fisiche e Naturali dell'Università del Molise:

- A.A. 2009-2010: corso di Laurea in Informatica (I livello)

*Sintesi del programma:* Concetti di base dell'Ingegneria del Software. I principi, i metodi, le tecniche, le metodologie e gli strumenti dell'Ingegneria del Software. Il processo di sviluppo del software. Modelli di ciclo di vita del software. Modellazione orientata agli oggetti e Unified Modeling Language (UML). Analisi e specifica dei requisiti. Progettazione di un sistema software e architetture software. Design Patterns. Object Design e trasformazioni del modello, refactoring, forward e reverse engineering. Software testing. Gestione dei Cambiamenti: Configuration Management e Rationale Management.

### **7.1.2 Corsi di Ingegneria del Software**

È stato professore a contratto per i seguenti corsi di "Ingegneria del Software", esame del II anno (48 ore di lezione frontali per un totale di 6 CFU) presso la Facoltà di Scienze Matematiche, Fisiche e Naturali dell'Università del Molise:

- A.A. 2008-2009: corso di Laurea in Informatica (I livello)
- A.A. 2007-2008: corso di Laurea in Informatica (I livello)
- A.A. 2006-2007: corso di Laurea in Informatica (I livello)
- A.A. 2005-2006: corso di Laurea in Informatica (I livello)
- A.A. 2004-2007: corso di Laurea in Produzione e Gestione dei Servizi Informatici (I livello)

*Sintesi del programma:* Concetti di base dell'Ingegneria del Software. I principi, i metodi, le tecniche, le metodologie e gli strumenti dell'Ingegneria del Software. Il processo di sviluppo del software. Modelli di ciclo di vita del software. Modellazione orientata agli oggetti e Unified Modeling Language (UML). Analisi e specifica dei requisiti. Progettazione di un sistema software e architetture software. Design Patterns. Object Design e trasformazioni del modello, refactoring, forward e reverse engineering. Software testing. Gestione dei Cambiamenti: Configuration Management e Rationale Management. Cenni su qualità del software, tracciabilità, metriche del software e manutenzione del software.



### 7.1.3 Corsi di Linguaggi di Programmazione

É stato professore a contratto per i seguenti corsi di “Laboratorio di Linguaggi di Programmazione”, esame del I anno (48 ore di lezione frontali per un totale di 6 CFU) presso la Facoltà di Scienze Matematiche, Fisiche e Naturali dell’Università del Molise:

- A.A. 2006-2007: corso di Laurea in Informatica (I livello)
- A.A. 2007-2008: corso di Laurea in Informatica (I livello)

*Sintesi del programma:* Panoramica sul Linguaggio C: elementi lessicali, operatori, tipi di dati fondamentali, flusso di controllo, funzioni, array e puntatori. L’accesso ai File in linguaggio C. Richiami sulle Struct, gestione dinamica della memoria. Le strutture di dati lista, pila e coda. La programmazione ad oggetti: il linguaggio Java. Principi fondamentali della programmazione a oggetti. Le classi e l’astrazione dei dati. Gli attributi e le funzioni membro. Costruttori e Distruttori. Gestione della memoria e garbage collection. La visibilità a livello di classe e l’accesso ai membri di una classe. Overloading delle funzioni e degli operatori. Conversioni di tipo. Ereditarietà. Le classi base e le classi derivate. Overriding e Polimorfismo.

### 7.1.4 Corsi di Sistemi Operativi

Ha insegnato “Sistemi Operativi” (48 ore di lezione frontali per un totale di 6 CFU) presso la Facoltà di Scienze Matematiche, Fisiche e Naturali dell’Università del Molise:

- A.A. 2008-2009: corso di Laurea in Informatica (I livello)
- A.A. 2009-2010: corso di Laurea in Informatica (I livello)

*Sintesi del programma:* Introduzione ai sistemi operativi. Cenni storici. Attività e struttura di un sistema operativo. Il kernel ed i moduli di un sistema operativo. I sistemi a processi. Proprietà dei processi. Operazioni sui processi. Stati di un processo. Cambio di contesto. Creazione e terminazione di un processo. Processi sequenziali, concorrenti ed in tempo reale. Processi leggeri (thread). Gestione dell’unità centrale. Criteri ed Algoritmi di scheduling. Programmazione concorrente. Cooperazione e sincronizzazione. Il problema della mutua esclusione. I semafori. Comunicazione tra processi: condivisione di memoria, scambio di messaggi. Deadlock. Gestione della memoria. Spazi di indirizzamento. Rilocalizzazione statica e dinamica. Memoria virtuale e swapping. Allocazione contigua della memoria, paginazione e segmentazione. Gestione della memoria secondaria e terziaria. Il file system. Struttura del file system. Attributi dei file, operazioni e metodi di accesso. Allocazione dei file. Gestione dello spazio libero.

### **7.1.5 Corsi di Tecnologie per lo Sviluppo del Web**

Ha insegnato “Tecnologie per lo Sviluppo del Web” (corso a scelta di 48 ore di lezione frontali per un totale di 6 CFU) presso la Facoltà di Scienze Matematiche, Fisiche e Naturali dell’Università del Molise:

- A.A. 2009-2010: corso di Laurea in Informatica (I livello)

*Sintesi del programma:* Il Web ed il protocollo HTTP. Richiami sul linguaggio di markup HTML, XML e XML Schema, XPath e XSLT, CSS Il modello MVC. Il web 2.0 La programmazione lato server con Java Richiami sulle Java Servlet, Sessioni, cookie e contesto. Aspetti avanzati sulle Java Servlets: Expression Language, Custom Tags, Property files, Resource Bundles La programmazione lato client con Java Server Pages. Javascript Esempi di framework per lo sviluppo di applicazioni web: Struts2 ed Hibernate AJAX (Asynchronous JavaScript and XML). Servizi web.

## **7.2 Incarichi di tutorato presso l’Università di Salerno**

Il dott. Fasano ha svolto incarichi di tutorato presso l’Università di Salerno sul corso di Ingegneria del Software.

### **7.2.1 Corsi di Ingegneria del Software**

- A.A. 2006-2007: attività di tutorato on-line per il corso di “Ingegneria del Software”, esame del II anno (48 ore di lezione frontali e 36 ore di laboratorio per un totale di 9 CFU) del corso di Laurea in Informatica (I livello), presso la Facoltà di Scienze Matematiche, Fisiche e Naturali dell’Università di Salerno, nell’ambito del progetto “Percorsi di Formazione a distanza E-Learning” (Fondo Sociale Europeo, POR Campania 2006 - Misura 3.22).
- A.A. 2005-2006: attività di tutorato, didattico - integrative, propedeutiche e di recupero sul corso di “Ingegneria del Software”, esame del II anno (48 ore di lezione frontali e 36 ore di laboratorio per un totale di 9 CFU) del corso di Laurea in Informatica (I livello), presso la Facoltà di Scienze Matematiche, Fisiche e Naturali dell’Università di Salerno (Fondo di ateneo per l’incentivazione al tutorato).

## **7.3 Attività seminariale**

Il dott. Fasano ha svolto attività seminariale sia presso l’Università di Salerno sia presso l’Università del Molise.

- Nel 2005, 2006, 2007 e 2008 ha tenuto diversi seminari sul tema del “Configuration Management”, ed in particolare sugli strumenti per la gestione di

artefatti software, per gli studenti del corso di Laurea in Informatica (I livello) della Facoltà di Scienze Matematiche, Fisiche e Naturali dell'Università degli Studi di Salerno. Durante il seminario stato presentato ADAMS, un sistema software per la gestione di progetti e artefatti software. Tale tool stato utilizzato dagli studenti del corso dal 2005 al 2007 ed i risultati della sperimentazione sono stati pubblicati in [C7, C8, C12, T1].

- Nel 2006 ha tenuto un ciclo di seminari sul tema delle “Ispezioni Software”, per gli studenti del corso di Laurea in Informatica (I livello) della Facoltà di Scienze Matematiche, Fisiche e Naturali dell'Università degli Studi di Salerno. Durante il seminario stato presentato un tool per il supporto alle Ispezioni Software Distribuite. Tale tool stato utilizzato dagli studenti del corso durante un'esercitazione controllata ed i risultati della sperimentazione sono stati pubblicati in [C11].
- Nel 2007 e nel 2008 ha tenuto un ciclo di seminari sul tema delle “Ispezioni Software” e della “Gestione del Razionale”, per gli studenti del corso di Laurea in Informatica (II livello) della Facoltà di Scienze Matematiche, Fisiche e Naturali dell'Università degli Studi di Salerno.

#### **7.4 Attività seminariale presso scuole internazionali**

Nel 2007 é stato invitato a tenere un ciclo di seminari sul tema della qualità del software, ed in particolare sul Configuration Management, sul Rationale Management e sulle Ispezioni Software, nell'ambito della IV edizione della “International School on Network Security Impact on Software Engineering” tenutasi presso l'Università Cattolica Duoc UC, a Viña del Mar (Cile), dal 29 al 31 Ottobre 2007.

### **8 Attività professionale**

Dal 2004 il dott. Fasano é abilitato all'esercizio della professione di Ingegnere (Laurea Specialistica – Sezione A). Dal 04/10/2004 é iscritto all'Albo Professionale presso l'Ordine degli Ingegneri della Provincia di Salerno (Nr. iscrizione 4660).

Dal 2006 é membro delle seguenti associazioni non-profit:

- “Institute of Electrical and Electronics Engineers” (IEEE);
- “IEEE Computer Society”;
- “Association for Computing Machinery” (ACM).

Dal 2007 é membro del Rotary International nel Club di Salerno.

Nel 2008 ha ottenuto un contratto di collaborazione con il Dipartimento di Matematica e Informatica presso la Facoltà di Scienze Matematiche, Fisiche e Naturali

dell'Università degli Studi di Salerno, per la realizzazione di contenuti multimediali, nell'ambito del progetto "Percorsi di Formazione a distanza E-Learning" (Fondo Sociale Europeo, POR Campania 2006 - Misura 3.22).

## **9 Collaborazioni con Università straniere**

Dal 01 Dicembre 2005 al 28 Febbraio 2006 il dott. Fasano ha trascorso un periodo di studi presso la Technische Universität di Monaco, Germania, svolgendo attività di ricerca sul tema "Sistemi cooperativi sincroni di supporto allo sviluppo software" sotto la supervisione del prof. Bernd Brügge. Ha inoltre continuato la collaborazione con il prof. Brügge ed i risultati di tale collaborazione sono stati presentati nella pubblicazione [C8].

## **10 Attività scientifica**

L'attività di ricerca del dott. Fasano si colloca prevalentemente nel settore dell'Ingegneria del Software, con particolare riferimento ai sistemi a supporto dello sviluppo di progetti software in ambito cooperativo. Il dott. Fasano si è anche occupato di tecniche e strumenti per la gestione ed il recupero di link di tracciabilità tra artefatti software e di gestione della qualità software. Quindi, i suoi principali interessi di ricerca possono essere così classificati:

1. Sistemi di supporto allo sviluppo cooperativo;
2. Gestione della tracciabilità tra artefatti software e recupero dei link di tracciabilità;
3. Tecniche e strumenti a supporto della qualità di artefatti software;

Di seguito è descritta in dettaglio l'attività scientifica svolta con riferimento alle pubblicazioni prodotte e indicate nell'elenco allegato (vedi Appendice A).

### **10.1 Sistemi di supporto allo sviluppo cooperativo**

Il cuore dell'attività di ricerca del dott. Fasano riguarda gli strumenti a supporto dell'Ingegneria del Software in ambito cooperativo.

Nella pubblicazione [C1] viene presentato ADAMS, un sistema a supporto dei processi basato su artefatti, che fornisce funzionalità proprie di sistemi di versioning dei documenti, funzionalità per la gestione di progetti software e funzionalità per la comunicazione e la cooperazione dei membri di un team per lo sviluppo di sistemi software.

La gestione del processo è estremamente flessibile ed basata sul livello di tracciabilità tra gli artefatti, piuttosto che sulla definizione di un workflow. Le dipendenze tra gli artefatti e una infrastruttura basata su eventi, consentono di notificare

gli sviluppatori su quando la lavorazione di un semilavorato del processo debba cominciare o debba essere riavviata a causa di modifiche in un artefatto da cui dipende.

Il sistema consente infine agli sviluppatori di inviare feedback nel caso in cui vengano riscontrati problemi su artefatti, consentendo così di riaprire il processo di lavorazione di un artefatto.

Il sistema include inoltre un tool basato su tecniche di information retrieval [J1, C4, C5] per supportare l'ingegnere del software nel recupero dei link di tracciabilità.

ADAMS consente di definire ruoli ed associare a questi i permessi per l'accesso alle funzionalità fornite dal sistema. Nella pubblicazione [C3] viene presentato un linguaggio visuale per definire politiche di autorizzazioni in ADAMS. In particolare, viene proposta una gerarchia di linguaggi visuali che supporta l'ingegnere del software nella modellazione di politiche di accesso basate su ruoli, in accordo al modello RBAC (Role Based Access Control). Le sentenze visuali vengono quindi tradotte in XACML (eXtended Access Control Markup Language) e memorizzate in un policy repository. In questo modo il sottosistema di policy management è in grado di processare richieste XACML durante l'esecuzione di ADAMS e confrontarle con le politiche definite nel repository.

Una tool demo del sistema ADAMS è stata presentata nella pubblicazione [C7].

Gli artefatti software di alto livello prodotti durante il ciclo di vita di un sistema software (documenti di pianificazione, analisi e progettazione) sono solitamente gestiti in maniera atomica dagli strumenti di Configuration Management. Tuttavia, un approccio a grana fine, in cui ciascun artefatto viene scomposto in una gerarchia di artefatti più semplici, è preferibile in quanto consente una migliore gestione della concorrenza, una più dettagliata definizione delle politiche di accesso e modifica della documentazione prodotta e consente una migliore gestione della tracciabilità tra gli artefatti.

Uno strumento di Configuration Management solitamente fornisce funzionalità per la gestione del versioning degli artefatti gestiti. Un approccio a grana fine deve dunque gestire coerentemente le diverse versioni degli artefatti composti così come di ciascuno dei suoi componenti.

Nella pubblicazione [C8] viene presentata la gestione del versioning gerarchico di ADAMS, con particolare riferimento a documenti di alto livello (documentazione e diagrammi) ed illustrando il supporto fornito ad un team di sviluppo distribuito. In particolar modo, vengono presentati i risultati di una sperimentazione dell'approccio e particolare enfasi viene posta sul versioning di diagrammi UML (Unified Modeling Language). Infatti, la complessità degli odierni sistemi software, richiede un estensivo uso della modellazione e, dunque, la produzione di numerosi diagrammi. Tali diagrammi sono spesso prodotti in diverse versioni e sono il frutto della collaborazione di diversi ingegneri del software.

L'approccio presentato nelle pubblicazioni [C8, C16, C19], prevede la modellazione dei diagrammi in modalità asincrona, come avviene per i più comuni sistemi di gestione delle versioni.

Nella pubblicazione [C18] viene inoltre anche affrontato il problema del merge automatico di versioni parallele di uno stesso diagramma e della gestione dei conflitti che si possono verificare qualora piú ingegneri del software modifichino lo stesso elemento del diagramma concorrentemente. L'approccio proposto utilizza una gestione a grana molto fine dei modelli (sono gestite le singole proprietà di ciascun possibile elemento di un modello UML), basata sullo standard per lo scambio di modelli XML. In tal modo, in maniera del tutto trasparente all'utente, possibile gestire la maggior parte dei conflitti in maniera automatica, richiedendo l'intervento dell'ingegnere del software esclusivamente nel caso in cui la modifica sia apportata concorrentemente ad una stessa proprietà di un medesimo elemento del modello.

L'approccio proposto in [C16, [C18]] é stato integrato in un tool di modellazione di diagrammi UML disponibile come plug-in per Eclipse. Nella pubblicazione [J2] viene discusso l'approccio a grana fine come supporto alla modellazione sincrona di diagrammi UML e viene presentato un tool che implementa l'idea presentata. Tale tool, denominato STEVE (Synchronous collaborative modelling Tool Enhanced with VErSIONing management), consente agli sviluppatori di accedere e modificare, nel medesimo istante, uno stesso diagramma UML, consentendo in tal modo, ai membri di un team distribuito, di discutere e modellare il sistema da realizzare direttamente all'interno di ADAMS. L'integrazione con uno strumento per la gestione di progetti e artefatti software, permette di rendere disponibili a tutti e mantenere sempre aggiornate (sotto forma di artefatti collegati tramite link di tracciabilità) sia le diverse versioni di ciascun elemento del modello, sia l'informazione testuale scambiata e condivisa durante i meeting. Infine, STEVE utilizza il sistema di notifiche fornito da ADAMS, per tenere informati i membri del team di progetto di tutte le modifiche a seguito dell'utilizzo del tool per modellare artefatti sui quali essi sono allocati.

Anche gli artefatti prodotti durante le fasi basse del ciclo di vita del software, quali il codice sorgente e i documenti di testing, possono beneficiare di un approccio a grana fine. Nella pubblicazione [C13], viene presentato un tool per la gestione a grana fine di codice Java, con particolari enfasi sul supporto alla context awareness che un simile strumento può fornire. Tale tool, denominato FineVMS, estende la command line di Subversion, un diffuso strumento open source per il versioning di documenti, per supportare il versioning a grana fine di codice Java. Per ciascun file gestito, una sua rappresentazione basata su XML contenente la struttura logica del file originale viene generato automaticamente (analizzando il codice in esso contenuto). Assieme a tale file aggiuntivo, viene generato un ulteriore file XML, contenente le informazioni relative ai vincoli di collaborazione. Quest'ultimo file consente di migliorare la context awareness tra gli sviluppatori, fornendo ad essi informazioni relative alle operazioni effettuate da altri sviluppatori sulla stessa unit logica (una classe, un metodo o un attributo) del file Java. Il tool comprende anche una estensione di Subversion, un plug-in di Subversion per Eclipse, che fornisce le funzionalità di versioning a grana fine direttamente all'interno dell'ambiente di sviluppo.

Le pubblicazioni [C12, T1] presentano, infine, i risultati della ricerca svolta, durante il periodo di dottorato, sul problema della gestione di artefatti software a grana fine, includendo i risultati di una sperimentazione condotta tra il 2005 ed il 2006, che ha visto l'utilizzo di ADAMS e della gestione a grana fine degli artefatti, quale strumento di gestione dei progetti e degli artefatti software prodotti dagli studenti dei corsi di Ingegneria del Software del Corso di Laurea in Informatica presso l'Università degli Studi di Salerno.

## **10.2 Gestione della tracciabilità tra artefatti software**

Un processo di sviluppo software prevede la realizzazione di numerosi artefatti, ciascuno dei quali può essere collegato ad altri artefatti. Ciò ancora più evidente nel caso in cui si adotti un approccio a grana fine, in cui un artefatto complesso (quale, ad esempio, un documento di specifica dei requisiti software) può essere considerato come una composizione a più livelli di artefatti più semplici (ad esempio i singoli requisiti software) e dunque il link di tracciabilità può interessare artefatti a diversi livelli della gerarchia.

La gestione della tracciabilità ampiamente riconosciuta come un fattore determinante per la buona riuscita di un progetto software. Durante la fase di realizzazione del sistema, infatti, le informazioni relative alla tracciabilità supportano l'ingegnere del software durante la stima della copertura dei requisiti del sistema e l'individuazione di componenti riusabili. In fase di manutenzione, invece, le stesse informazioni sono essenziali per una accurata valutazione dell'impatto di una eventuale modifica, nonché per la comprensione del codice esistente. Nella pubblicazione [C6] viene presentata la gestione della tracciabilità realizzata in ADAMS [C1, C7]. In particolare, oltre a fornire il supporto all'ingegnere del software durante l'intero ciclo di vita del software, le informazioni sulla tracciabilità gestite in ADAMS, sono anche utilizzate per propagare gli eventi riguardanti cambiamenti apportati su di un artefatto, agli artefatti che da esso dipendono. In tal modo, il livello di context awareness tra i membri del team aumenta. ADAMS gestisce un grafo di tracciabilità, ossia un grafo avente per nodi gli artefatti software e per archi i link di tracciabilità. Tale grafo può essere visualizzato e navigato per accedere ad artefatti collegati tra di loro, scaricarne le diverse versioni o sottoscrivere eventi di cui essere notificati.

Uno dei principali problemi relativi alla gestione della tracciabilità tra artefatti software riguarda l'identificazione dei link. Ci dipende in parte dal numero elevato di artefatti prodotti per un sistema di medie dimensioni, ed in parte dal fatto che, anche dopo aver definito il layer di tracciabilità, questo può variare dopo ogni modifica, inserimento o cancellazione di uno degli artefatti del sistema. Una risposta al problema è rappresentata da un supporto automatico o semi-automatico alla fase di recupero dei link di tracciabilità. Basandosi sulla considerazione che gran parte degli artefatti contengono informazioni in formato testuale, nella pubblicazione [C2] viene presentato e sperimentato un metodo per il recupero dei link di tracciabilità tra artefatti (in particolare, casi d'uso e diagrammi di interazione). Tale

metodo é basato sul Latent Semantic Indexing (LSI), una tecnica di Information Retrieval (IR), per l'individuazione di link candidati sulla base della somiglianza testuale tra gli artefatti legati dalla dipendenza.

Nella pubblicazione [C4] l'approccio é stato esteso anche ad altre tipologie di artefatti software (casi di test e codice sorgente). I risultati ottenuti durante la valutazione dell'approccio presentato in [C2, C4] hanno fornito lo spunto per la realizzazione di un tool, integrato all'interno di ADAMS, per il supporto al recupero dei link di tracciabilità basato sul LSI. Tale tool é stato presentato nella pubblicazione [C5].

Nella pubblicazione [C9] viene discusso il problema legato al fatto che, utilizzando tecniche di IR, recuperare tutti i link di tracciabilità esistenti é generalmente impraticabile, a causa dell'elevato sforzo richiesto per rimuovere i link erroneamente proposti da un tool automatico. Pertanto sorge la necessità di stabilire una "soglia ottimale", durante il processo di recupero, che consenta di ottenere un compromesso accettabile tra i link tracciati ed i link erroneamente identificati. Sfortunatamente, tale soglia non é conosciuta a priori, pertanto viene suggerito di utilizzare un approccio incrementale durante il processo di recupero dei link di tracciabilità. Tale approccio consiste nel definire un'alta soglia di somiglianza quale parametro del tool, limitando in questo modo il numero di link proposti, ma ottenendo una maggiore precisione nei risultati. Nelle iterazioni successive, tale soglia viene abbassata, per recuperare link esclusi nelle iterazioni precedenti. Ovviamente, man mano che la soglia viene abbassata, la precisione del tool diminuisce. L'obiettivo é procedere finché la precisione del tool non sia considerata troppo bassa ovvero finché lo sforzo richiesto per scartare i numerosi link erroneamente proposti non sia troppo elevato. Nella pubblicazione [J1] vengono presentati i risultati di diversi casi di studio finalizzati alla valutazione dell'utilizzo di tecniche di IR come supporto al recupero dei link di tracciabilità. Nella pubblicazione [J1] viene anche evidenziato come lo stesso approccio utilizzato per individuare potenziali link di tracciabilità tra gli artefatti prodotti, possa essere utilizzato per evidenziare problemi di qualità tra gli stessi artefatti. Infatti, nel caso in cui un link tracciato dall'ingegnere del software coinvolga due artefatti la cui somiglianza sia molto bassa, ci potrebbe essere indice di una scarsa qualità di uno o entrambi gli artefatti (ad esempio, a causa dell'uso di identificativi poco significativi ovvero di descrizioni o commenti insufficienti).

### **10.3 Tecniche e strumenti a supporto della qualità di artefatti software**

Le ispezioni software sono una pratica dell'Ingegneria del Software il cui scopo consiste nell'identificare difetti all'interno di artefatti software, ridurre la necessità di manutenzione correttiva e produrre sistemi software di elevata qualità. Uno degli aspetti più controversi, relativamente ai processi di ispezione tradizionali, riguarda la necessità di eseguire un meeting in cui tutti gli ispettori si incontrano per discutere sui problemi individuati nell'artefatto oggetto dell'ispezione. Tale meeting



sincrono é costoso, in quanto richiede la presenza simultanea di tutti i partecipanti al processo di ispezione e, affinché sia efficace, esso richiede un'adeguata preparazione e moderazione, nonché cooperazione tra i membri del gruppo. In un ambiente distribuito, tali problemi sono ancora più evidenti, in quanto le distanze geografiche rendono ancora più costosi gli incontri in co-presenza.

Nella pubblicazione [C10] viene presentato un processo di ispezione software distribuito, nel quale il meeting sincrono viene preceduto da una discussione asincrona il cui scopo quello di eliminare i conflitti emersi durante l'individuazione dei difetti, evitando, in molti casi, la necessità del meeting sincrono. Viene anche presentato WAIT (Web-based Artefact Inspection Tool), un tool che supporta il processo di ispezione distribuito. WAIT é stato integrato in ADAMS [C1, C7] e sperimentato per mezzo di un esperimento controllato. I risultati dell'esperimento controllato, presentati nella pubblicazione [C11], hanno dimostrato l'efficacia del processo e del tool nell'individuare un numero di difetti maggiore di quello individuato utilizzando un processo classico a discapito di un maggior numero di falsi difetti identificati. Tuttavia, il costo da sostenere per scartare i difetti erroneamente segnalati é ripagato dai minori costi derivanti dall'eliminazione del meeting sincrono (che non é mai stato necessario durante la sperimentazione) e dal minore tempo di esecuzione del processo rilevato.

A conferma del riconoscimento dell'attività di ricerca svolta nel campo della gestione della Qualità nei Sistemi Software, il dott. Fasano é stato invitato a tenere nel 2007 un ciclo di seminari su tale argomento nell'ambito della IV edizione della *"International School on Network Security Impact on Software Engineering"* tenutasi presso l'Università Cattolica Duoc UC a Viña del Mar (Cile).

#### **10.4 Reverse engineering e Migrazione di Sistemi Software Ereditati verso Piattaforme moderne**

Durante il suo ciclo di vita, un sistema software può trovarsi nella situazione di dover essere adattato, reingegnerizzato e migrato ai fini della coesistenza con sistemi software diversi e sviluppati in nuovi ambienti tecnologici, in particolare in ambienti client-server e/o basati su tecnologia ad oggetti. La sostituzione del vecchio sistema con un sistema sviluppato conformemente ai nuovi requisiti tecnici è quasi sempre inammissibile a causa degli elevati costi che una tale soluzione comporta. Una soluzione più plausibile da un punto di vista economico è la decomposizione di un tale sistema in componenti funzionalmente omogenei sui quali è possibile compiere delle scelte indipendenti. Le scelte che un ingegnere del software potrà effettuare su tali sottosistemi variano dalla reingegnerizzazione, all'incapsulamento di tali componenti in "object wrapper", alla loro sostituzione incrementale. Nella pubblicazione [C19], viene presentato un plug-in per Eclipse, realizzato per supportare il processo di migrazione di sistemi software ereditati verso piattaforme moderne. In particolare, si affronta il problema di voler migrare la gestione dei dati gestiti da una applicazione scritta nel linguaggio di programmazione COBOL verso un sistema che utilizzi basi di dati relazionali.

Nella pubblicazione [CI1] viene presentato un altro tool realizzato come plugin Eclipse per la definizione e l'esecuzione di processi di reverse engineering e di comprensione di sistemi informativi tradizionali e basati sul web. I processi sono definiti in termini di activity diagram UML in cui a ciascuna attività è possibile associare componenti software predefinite o da realizzare. Tali componenti, implementate realizzando linguaggi di programmazione tradizionali e ambienti software per l'analisi dei dati, possono essere ovviamente riutilizzati.

## Appendice A: Elenco complessivo delle pubblicazioni

### Dissertazioni

- T1 F. Fasano, “Fine-grained management of Software Artefacts”, *Tesi di Dottorato di Ricerca in Informatica*, Dipartimento di Matematica e Informatica, Facoltà di Scienze MM. FF. e NN., Università degli Studi di Salerno, 2007, Disponibile online: <http://www.sesa.dmi.unisa.it/thesis/fasano.pdf>

### Pubblicazioni in riviste internazionali

- J1 A. De Lucia, **F. Fasano**, R. Oliveto, G. Tortora, “Recovering Traceability Links in Software Artefact Management Systems using Information Retrieval Methods”, *ACM Transactions on Software Engineering and Methodology*, vol. 16, no. 4, article no. 13, ISSN: 1049-331X, 2007.
- J2 A. De Lucia, **F. Fasano**, G. Scanniello, and G. Tortora, “Enhancing Collaborative Synchronous UML Modelling with Fine-grained Versioning of Software Artefacts”, *Journal of Visual Languages and Computing (JVLC)*, vol. 18, no. 5, pp.492-503, ISSN: 1045-926X, 2007.
- J3 **F. Fasano** and R. Oliveto, “Supporting Project Management with Fine-Grained Artefact Management in ADAMS”, *International Journal of Computers and Applications* special issue on the International Summer School on Software Engineering, vol. 31, no. 3, pp. 145-152, 2009.
- J4 A. De Lucia, **F. Fasano**, G. Scanniello, and G. Tortora, “Evaluating distributed inspection through controlled experiments”, *IET Software*, vol. 3, no. 5, pp. 381 - 394, 2009.

### Pubblicazioni in atti di congressi internazionali

- C1 A. De Lucia, **F. Fasano**, R. Francese, and G. Tortora, “ADAMS: an Artefact-based Process Support System”, in *Proceedings of 16<sup>th</sup> International Conference on Software Engineering and Knowledge Engineering*, Banff, Alberta, Canada, Knowledge Systems Institute, USA, pp. 31-36, ISBN 1-891706-14-4, 2004.
- C2 A. De Lucia, **F. Fasano**, R. Francese, and R. Oliveto, “Recovering Traceability Links between Requirement Artefacts: a Case Study”, in *Proceedings of 16<sup>th</sup> International Conference on Software Engineering and Knowledge Engineering - Workshop on Knowledge Oriented Maintenance*, Banff, Alberta, Canada, Knowledge Systems Institute, USA, pp. 453-456, ISBN 1-891706-14-4, 2004.

- C3 A. De Lucia, **F. Fasano**, M. Giordano M., G. Polese G., and G. Tortora, “Visual Authorization Policies in a Process Support System”, in *Proceedings of 3<sup>rd</sup> Workshop on Cooperative Supports for Distributed Software Engineering Processes*, Linz, Austria, Austrian Computer Society, pp. 83-97, ISBN 3-85403-180-7, 2004.
- C4 A. De Lucia, **F. Fasano**, R. Oliveto, and G. Tortora, “Enhancing an Artefact Management System with Traceability Recovery Features”, in *Proceedings of 20<sup>th</sup> International Conference on Software Maintenance*, Chicago, Illinois, USA, IEEE Computer Society Press, USA, pp. 306-315, ISBN 0-7695-2213-0, 2004.
- C5 A. De Lucia, **F. Fasano**, R. Oliveto R., and Tortora G., “ADAMS Re-Trace: a Traceability Recovery Tool”, in *Proceedings of 9<sup>th</sup> European Conference on Software Maintenance and Reengineering*, The Manchester Conference Centre, Manchester, UK, pp. 32-41, ISBN 0-7695-2304-8, 2005.
- C6 A. De Lucia, **F. Fasano**, R. Francese, and R. Oliveto, “Traceability Management in ADAMS”, in *Proceedings of the 1<sup>st</sup> International Workshop on Distributed Software Development*, Paris, France, Austrian Computer Society, pp. 135-149, ISBN 3-85403-193-9, 2005.
- C7 A. De Lucia, **F. Fasano**, R. Oliveto, and G. Tortora, “ADAMS: ADvanced Artefact Management System”, in *Proceedings of the 10<sup>th</sup> European Conference on Software Maintenance and Reengineering*, Bari, Italy, pp. 347-348, ISBN 0-7695-2536-9, 2006.
- C8 B. Brügge, A. De Lucia, **F. Fasano**, and G. Tortora, “Supporting Distributed Software Development with fine-grained Artefact Management”, in *Proceedings of 1<sup>st</sup> IEEE International Conference on Global Software Engineering*, Costão do Santinho, Florianopolis, Brazil, pp.213-222, ISBN 0-7695-2663-2, 2006.
- C9 A. De Lucia, **F. Fasano**, R. Oliveto, and G. Tortora, “Can Information Retrieval Techniques Effectively Support Traceability Link Recovery?”, in *Proceedings of the 14<sup>th</sup> International Conference on Program Comprehension*, Athens, Greece, IEEE Computer Society Press, pp. 307-316, ISBN 0-7695-2601-2, 2006.
- C10 A. De Lucia, **F. Fasano**, G. Scanniello, and G. Tortora, “Integrating a Distributed Inspection Tool within an Artefact Management System”, in *Proceedings of 2<sup>nd</sup> International Conference on Software and Data Technologies*, Barcelona, Spain, pp. 184-189, ISBN 978-989-8111-06-7, 2007.
- C11 A. De Lucia, **F. Fasano**, G. Scanniello, and G. Tortora, “Assessing the Effectiveness of a Distributed Method for Code Inspection: A Controlled Experiment”, in *Proceedings of 2<sup>nd</sup> IEEE International Conference on Global*

*Software Engineering*, Munich, Germany, IEEE Computer Society Press, pp. 252-261, ISBN 0-7695-2920-8, 2007.

- C12 **F. Fasano**, "Fine-Grained Management of Software Artefacts", in *Proceedings of 23<sup>rd</sup> International Conference on Software Maintenance*, Paris, France, pp.507-508, ISBN 1-4244-1256-0, 2007.
- C13 A. De Lucia, **F. Fasano**, R. Oliveto, and D. Santonicola, "Improving Context Awareness in Subversion through Fine-Grained Versioning of Java Code", in *Proceedings of International Workshop on Principles of Software Evolution (IWPSE'07)*, Dubrovnik, Croatia, ACM Press. pp. 110-114, ISBN 978-1-59593-811-4, 2007.
- C14 A. De Lucia, **F. Fasano**, G. Scanniello, and G. Tortora, "Comparing Inspection Methods using Controlled Experiments", in *Proceedings of 12<sup>th</sup> International Conference on Evaluation and Assessment in Software Engineering*, Bari, Italy, G. Visaggio, M. T. Baldassarre, S. Linkman, and M. Turner (eds.), BCS eWIC, 2008.
- C15 A. De Lucia, **F. Fasano**, R. Francese, and I. Passero, "Enhancing Rationale Management with Second Life Meetings", to appear in *Proceedings of 14<sup>th</sup> International Conference on Distributed Multimedia Systems*, Boston, USA, pp. 118-123, ISBN/ISSN: 1-891706-23-3, 2008.
- C16 I. Barone, A. De Lucia, **F. Fasano**, E. Rullo, G. Scanniello, and G. Tortora, "COMOVER: Concurrent Model Versioning", in *Proceedings of 24<sup>th</sup> International Conference on Software Maintenance*, Beijing, China, pp.462-463, ISBN 1-4244-2614-0, 2008.
- C17 A. De Lucia, **F. Fasano**, and R. Oliveto, "'Traceability Management for Impact Analysis'", in *Frontiers of Software Maintenance*, 2008, pp. 21-30, ISBN: 978-1-4244-2654-6
- C18 A. De Lucia, **F. Fasano**, G. Scanniello, and G. Tortora "'Concurrent Fine-grained Versioning of UML Models'" In *Proceedings of 13th European Conference on Software Maintenance and Reengineering*, Fraunhofer, Kaiserslautern, Germany, March 24-27, 2009. IEEE Computer Society Press, pp. 89-98, 2009.
- C19 A. De Lucia, **F. Fasano**, M. Mennella, G. Scanniello, and G. Tagliamonte, "'DB-MELIS: An Eclipse Plug-in for Data Migration'" In *Proceedings of 13th European Conference on Software Maintenance and Reengineering*, Fraunhofer, Kaiserslautern, Germany, March 24-27, 2009. IEEE Computer Society Press, pp. 329-330, 2009.
- C20 A. De Lucia, **F. Fasano**, C. Grieco, and G. Tortora, "'Recovering design rationale from email repositories'" In *Proceedings of 25<sup>th</sup> International Con-*

*ference on Software Maintenance*, Edmonton, Alberta, Canada. IEEE Computer Society Press, pp. 543-546, 2009.

**Pubblicazioni in atti di congressi nazionali**

- CII A. De Lucia, **F. Fasano**, M. Risi, and G. Scanniello, “An Eclipse Based Environment to Define and Execute Processes with Application to the Reverse Engineering”, in *Proceedings of 3<sup>rd</sup> Italian Workshop on Eclipse Technologies*, Bari, Italy, pp.27-36, ISBN 978-88-902981-4-1, 2008.

---

Curriculum Vitae et Studiorum e dell'attività didattica e scientifica del dott. Fausto Fasano.  
Aggiornato al 15/02/2010.