

Corso di formazione

La progettazione del software per il controllo di macchine automatiche: Design Patterns e Framework di Riferimento

1. Problematiche ed attuale (tipico) scenario industriale

È indubbio che il "successo" di una macchina automatica, dal punto di vista funzionale e prestazionale, discende primariamente dalle scelte progettuali operate in sede di definizione della sua struttura meccanica. Tuttavia, dagli stessi punti di vista, senz'altro fondamentale è anche il ruolo svolto dal sistema di elaborazione preposto al controllo della macchina, essendo sempre più ampio e rilevante lo spettro dei compiti ad esso affidati. Oltre alle attività strettamente inerenti l'automazione dei cicli di lavorazione della macchina, altre funzionalità complementari ma non meno importanti che tale sistema è tipicamente chiamato ad espletare riguardano la possibilità di:

- emulare il comportamento di tradizionali organi meccanici, quali camme e riduttori, consentendo non soltanto una drastica semplificazione costruttiva della macchina stessa, ma anche la sua automatica predisposizione, peraltro in tempi nulli, a fronte di mutate condizioni di lavoro (cambio formato);
- adattare dinamicamente strategie e parametri di controllo in relazione a differenziate esigenze produttive, garantendo in ogni condizione di funzionamento il pieno sfruttamento delle potenzialità operative della macchina;
- conseguire un'elevata qualità del prodotto finito attraverso una capillare verifica della correttezza delle lavorazioni su di esso effettuate, prevenendo nel contempo lavorazioni disutili e spreco di materie prime;
- indirizzare l'operatore addetto alla conduzione della macchina nell'intraprendere in maniera tempestiva e puntuale le azioni volte a stabilire o a ripristinare le condizioni ottimali di funzionamento, sollevandolo parimenti da interventi pericolosi o quantomeno onerosi;
- gestire in tempo reale informazioni di diagnostica e dati di produzione, a supporto delle attività di manutenzione della macchina, di gestione delle materie prime e dei prodotti finiti, di organizzazione e pianificazione della produzione.

Se, in generale, la definizione della struttura hardware del sistema di controllo non pone particolari difficoltà, essendo direttamente reperibili sul mercato "soluzioni sufficientemente universali" basate su architetture con spiccate caratteristiche di modularità, di espandibilità, di diretta compatibilità con il campo, più problematico ed eterogeneo risulta spesso il processo di progettazione e realizzazione del software. Diverse possono essere le motivazioni:

- carenza di modelli di riferimento e di "design patterns" specifici, non essendo ancora oggi emerse proposte significative (sia da parte delle Aziende che occupano una posizione di rilievo a livello internazionale nella produzione di tecnologie informatiche per l'automazione, sia nell'ambito della letteratura tecnico-scientifica specializzata) che possano rivestire il ruolo di "standard de facto", o quantomeno di linee guida, nella progettazione del software per il controllo di macchine automatiche;
- limitata disponibilità di componenti "off-the-shelf" e di infrastrutture tecnologiche, essendo il supporto orientato alla realizzazione delle funzionalità di usuale interesse tipicamente confinato al dominio della regolazione in catena chiusa, del "motion control", dell'interconnessione di dispositivi tramite bus di campo, dello sviluppo di interfacce utente evolute;
- limitata applicabilità di modelli computazionali evoluti, essendo i sistemi PLC-based normalmente utilizzati per il controllo di macchine automatiche non adatti in maniera nativa a supportare la naturale strutturazione di una applicazione in termini di processi concorrenti ed opportunamente cooperanti;
- limitata espressività dei linguaggi di programmazione disponibili, con conseguente più laboriosa, e comunque non sistematica o coerente, applicabilità di consolidati principi e metodi propri dell'ingegneria del software.

Al di là delle attuali limitazioni tecnologiche e delle oggettive difficoltà che la progettazione di un sistema indubbiamente complesso comporta, altri fattori (forse i principali !?) possono in generale concorrere ad estendere in maniera indesiderata i tempi, e conseguentemente i costi, di sviluppo e manutenzione del software. Tra questi:

- il ruolo ancillare spesso attribuito all'attività svolta dai progettisti software tende ad avvallare la realizzazione non tanto di sistemi caratterizzati da una solida e flessibile struttura, quanto piuttosto di prototipi "rapidamente

operativi" che possano fungere da veicolo per la verifica sperimentale delle prestazioni effettivamente conseguibili con le macchine dal punto di vista meccanico;

- le fondamenta delle soluzioni prototipali, già spesso precarie all'origine, raramente sono poi di fatto oggetto di un processo di consolidamento, ma, al contrario, subiscono nel tempo ulteriori perturbazioni imputabili agli inevitabili continui aggiornamenti derivanti dalla personalizzazione di funzionalità esistenti o dall'integrazione di nuove funzionalità richieste dai clienti;
- i pressanti impegni di lavoro quotidiano, l'esigenza di rispettare spesso stringenti scadenze temporali, non solo rendono inattuabile l'auspicabile "brain storming" e scambio di esperienze significative tra i progettisti software, ma spesso possono indurli ad operare scelte mirate al soddisfacimento di obiettivi realizzativi specifici, precludendo l'adozione di un approccio organico e di più ampio respiro finalizzato all'individuazione di soluzioni direttamente riutilizzabili in macchine contraddistinte da funzionalità similari;
- indubbiamente rilevante è anche il fatto che il background culturale dei progettisti software, ancorché derivante da studi universitari, di fatto risulta inevitabilmente soggetto a rapida obsolescenza stante la straordinaria evoluzione in atto nel settore delle tecnologie dell'informazione, rendendo decisamente elevato il rischio che la loro attività risulti sempre più diffusamente guidata dall'intuizione o dall'esperienza piuttosto che fondata su solide basi metodologiche.

2. Il programma di aggiornamento formativo

Obiettivi:

- Stimolare la sensibilità dei progettisti software, nel momento in cui sono chiamati ad affrontare tematiche "trasversali" sovente ricorrenti in sede di realizzazione del sistema di controllo di una macchina automatica, ad individuare soluzioni di più ampia portata rispetto alle particolari e contingenti specifiche di progetto.
- Delineare l'approccio metodologico ed i criteri di progetto che possono condurre alla realizzazione di un'architettura software "machine independent"
 - in grado di supportare l'esecuzione di tutte quelle attività che comunemente sono delegate al sistema di controllo di una macchina automatica;
 - ampiamente riutilizzabile, attraverso semplice configurazione piuttosto che sviluppo di codice ad hoc, in scenari simili;
 - aperta a possibili integrazioni e/o estensioni finalizzate al soddisfacimento di specifiche esigenze applicative.
- Enfatizzare, nell'ottica di un'auspicabile e non più differibile riusabilità del software, l'efficacia dell'approccio "model-driven", che prevede di definire, per ogni tipologia di problemi affrontati, un modello di riferimento di adeguata capacità espressiva ("la base di conoscenza"), e di procedere una volta per tutte allo sviluppo del correlato programma ("il motore inferenziale"), in modo da conseguire prestazioni e comportamenti anche fortemente differenziati a partire dalla semplice configurazione parametrica del modello.
- Guidare il processo di acquisizione di competenze specifiche e di capacità operative sulle modalità di integrazione in nuove applicazioni del framework architetturale, delle infrastrutture tecnologiche, degli strumenti di ausilio alla progettazione recentemente sviluppati nell'intento sia di ridurre drasticamente i tempi di progetto per la realizzazione di nuovi sistemi, sia di favorire l'intercambiabilità dei progettisti, oltre che delle piattaforme computazionali, in contesti affini.

Argomenti:

A) ASPETTI METODOLOGICI DI PROGETTO DI CARATTERE GENERALE.

- Il fondamentale principio della distinzione in sede di progetto dei tre aspetti "what to do", "when to do", "how to do".
- L'approccio progettuale "divide et impera".
 - Criteri per la decomposizione funzionale di un sistema complesso in termini di una gerarchia di entità (sottosistemi, moduli, ..., componenti elementari), astratte o concrete, opportunamente cooperanti.
 - Definizione del ruolo e delle funzionalità di ciascuna entità ("what to do"), nonché delle relative interfacce e dei protocolli previsti per l'interazione con altre entità operanti nello stesso livello o nei livelli adiacenti della gerarchia.
- Modelli di riferimento per la definizione formale del comportamento delle singole entità ("how to do").
- Modelli di riferimento per la definizione delle modalità di esecuzione dei compiti delle singole entità ("when to do").
 - Strutturazione di un'applicazione in termini di processi cooperanti e criteri di progetto orientati a favorirne la portabilità in ambienti di esecuzione differenziati.
 - Tipologie di processi (hard real-time, soft real-time, non real-time).
 - L'approccio "clock driven" per la gestione di (pseudo-)processi in ambienti di esecuzione non concorrenti del tipo PLC-based.
 - L'approccio "priority driven" per la gestione di processi in ambienti di esecuzione concorrenti del tipo Soft-PLC o PC-based.
- Esempificazione degli aspetti metodologici con riferimento ad un semplice caso applicativo.

B) UN FRAMEWORK DI RIFERIMENTO "MACHINE-INDEPENDENT & PLATFORM INDEPENDENT".

1) Il sistema di gestione delle informazioni di diagnostica.

- Contesto, finalità e problematiche.

- Modelli di riferimento per l'identificazione automatica di malfunzionamenti o guasti ed una possibile tassonomia delle corrispondenti segnalazioni (allarmi, anomalie, avvertimenti, indicazioni).
- Modalità di specifica dell'impatto di ciascuna segnalazione sulle condizioni operative della macchina (arresto incondizionato (in emergenza, immediato, in fase, ...), arresto temporaneo, ...).
- Modalità di gestione delle segnalazioni ("time-driven" o "priority-driven") e della loro presentazione (tramite messaggi, avvisi acustici, avvisi luminosi, ...) all'operatore.
- Modalità di reset e di autoreset (incondizionato, condizionato, temporaneo, a macchina ferma, ...) delle singole segnalazioni.
- Architettura del sistema.
- Esempificazione degli aspetti metodologici e di configurazione del sistema con riferimento a tipici casi applicativi.

3. ORGANIZZAZIONE DEL CORSO

Il corso dura 30 ore, prevedendo nel complesso 5 interventi settimanali della durata di 6 ore ciascuno (9.30-16.30).

Calendario:

- **30 Gennaio 2018 (9.30-16.30)** presso sede LIAM Via Vignolese 1910 Spilamberto
- **6 Febbraio 2018 (9.30-16.30)** presso sede LIAM Via Vignolese 1910 Spilamberto
- **13 Febbraio 2018 (9.30-16.30)** presso sede LIAM Via Vignolese 1910 Spilamberto
- **20 Febbraio 2018 (9.30-16.30)** presso sede LIAM Via Vignolese 1910 Spilamberto
- **27 Febbraio 2018 (9.30-16.30)** presso sede LIAM Via Vignolese 1910 Spilamberto

Il corso sarà tenuto dal **Prof. Eugenio Faldella**, della Scuola di ingegneria e Architettura (nuovo nome della Facoltà di Ingegneria), Università di Bologna, Professore Ordinario del dipartimento di informatica - scienza ed ingegneria (DISI) Docente di "Sistemi digitali" di "Sistemi in tempo reale" e di "Sistemi di elaborazione e controllo".

Per iscriversi ed informazioni riguardante la quota di partecipazione potete inviare una e-mail a Giorgio Nipoti, giorgio.nipoti@liamlab.it o contattare telefonicamente al numero 059 8399580

Curriculum Vitae

Note biografiche

Nato a Bologna il 14 Ottobre 1948, nel settembre del 1972, subito dopo aver conseguita la laurea in Ingegneria Elettronica presso l'Università di Bologna con voti 100/100 e lode, ha iniziato la propria attività accademica presso l'Istituto di Elettronica (confluito in seguito nel Dipartimento di Elettronica, Informatica e Sistemistica) della Facoltà di Ingegneria della stessa Università, dapprima in qualità di contrattista, poi di professore incaricato e assistente ordinario, e, dal 1980, di professore associato di Elettronica Industriale.

Nel 1977, in seguito alla vincita di una borsa di studio CNR del Comitato Nazionale per le Scienze di Ingegneria e Architettura, e nel 1978, in seguito alla vincita di una borsa di studio NATO Senior Fellowships Scheme, ha trascorso un periodo di studi di 8 e 6 mesi rispettivamente presso il Polytechnic of Central London, svolgendo attività sia di ricerca che didattica.

Nell'A.A. 1990-91, a seguito di concorso, è stato chiamato dalla Facoltà di Ingegneria dell'Università di Perugia a coprire in qualità di professore straordinario la cattedra di Elettronica Applicata. Nell'A.A. 1993-94 ha svolto la propria attività presso la Facoltà di Ingegneria dell'Università di Ferrara in qualità di professore ordinario di Elettronica dei Sistemi Digitali. Dall'A.A. 1994-95 è professore ordinario (SSD ING-INF/05) presso la Facoltà di Ingegneria dell'Università di Bologna.

Dal 1983 al 2007 è stato membro del Collegio dei Docenti del Dottorato di Ricerca in Ingegneria Elettronica ed Informatica presso la Facoltà di Ingegneria dell'Università di Bologna. Presso la stessa Facoltà è stato Direttore del Centro di Calcolo dal 1997 al 2001 e Presidente del Consiglio dei Corsi di Studio (Laurea e Laurea Specialistica) in Ingegneria Informatica dal 2001 al 2007.

Attività didattica

A partire dall'A.A. 1976-77 è stato ininterrottamente titolare di insegnamenti universitari afferenti ai Corsi di Studio in Ingegneria Informatica o Elettronica. Specificatamente: Statistica Applicata, Elettronica Industriale, Elettronica Applicata, Reti Logiche, Calcolatori Elettronici, Sistemi di Elaborazione, Ingegneria del Software, Reti di Calcolatori. Attualmente è titolare di due insegnamenti, entrambi afferenti al Corso di Laurea Specialistica in Ingegneria Informatica: "Sistemi Digitali" e "Sistemi in Tempo Reale".

Su invito di Enti pubblici e privati (Scuola superiore Reiss Romoli della STET, SNAM Progetti, ENI, Regione Emilia-Romagna, Sogesta, EnAIP, IFOA, AEI, ASTER, IMA, G.D, SACMI, CRIT), ha tenuto anche numerosi corsi e seminari di carattere monografico, riguardanti: metodologie di progetto di sistemi digitali,

controllo digitale di azionamenti a velocità variabile, architetture e applicazioni di sistemi a microprocessore, sistemi di elaborazione in tempo reale per il controllo di macchine automatiche, robotica e automazione industriale.

Attività di ricerca

L'attività scientifica, documentata in più di 80 lavori a stampa pubblicati in qualificate riviste internazionali o nei proceedings di convegni internazionali, ha in un primo tempo riguardato i seguenti ambiti disciplinari:

- reti locali per applicazioni in tempo reale;
- sistemi di controllo a microprocessore per il condizionamento della potenza in apparati ed impianti di rilevante interesse applicativo;
- modelli neurali ed architetture computazionali per il riconoscimento di oggetti 3-D mediante mani robotiche ad elevata destrezza;
- metodologie di progetto per la sintesi a livello architeturale di sistemi digitali.

Più recentemente l'attività di ricerca è stata finalizzata allo studio e realizzazione di:

- tecniche avanzate per la gestione di certificati in infrastrutture a chiave pubblica;
- sistemi embedded per l'automazione di macchine e processi industriali.