

UNIVERSITÀ DEGLI STUDI DI BOLOGNA
SCUOLA DI INGEGNERIA E ARCHITETTURA

Corso di Laurea in Ingegneria Informatica
Dipartimento di Informatica — Scienza e Ingegneria

TESI DI LAUREA
in
Reti di Calcolatori X???

**Il titolo deve essere generale, non troppo verboso, ma neanche troppo fumoso, e non deve avere troppe sigle e termini inglesi.
Evitare i nomi di prodotto, se possibile**

**Sistemi ad Agenti per l'Accesso Efficiente
ad Informazioni Distribuite**

Relatori e Correlatori

Se la tesi è stata molto limitata, il solo relatore

Se la tesi è di impegno, almeno due correlatori che hanno seguito il lavoro

Tesi di impegno, tesi più approfondita e sperimentale.

Tesi di Laurea di:

Relatore:

Chiar.mo Prof. Ing. Antonio Corradi

Correlatori:

Prof. Ing. Paolo Bellavista

Dott. Ing. Luca Foschini

I relatori e correlatori devono essere indicati con i titoli di studio giusti.

Anno Accademico 2014 - 2015

**Ricordatevi di non usare il bollo simbolo dell'Ateneo di Bologna:
assolutamente proibito!**

Metallivello

In ogni documento scritto tecnico abbiamo un obiettivo di comunicazione riguardo ad un argomento specifico: ovviamente, questo ci porta alla organizzazione della tesi secondo tutti i criteri espressi qui.

Dobbiamo però enunciare un principio generale.

Contenuto e Metallivello

Insieme con il contenuto, dobbiamo continuamente, o almeno in alcune parti e più spesso possibile, veicolare anche le motivazioni alla base delle considerazioni che stiamo facendo. In altre parole dobbiamo spiegare anche perché stiamo dicendo quello che stiamo dicendo e dirigere il lettore (ipotetico e in fabula come direbbero alcuni) a capire meglio la struttura mentre gliela esponiamo.

Notate che maggiore è la giustapposizione dei due piani, il contenuto e la descrizione del contenuto, migliore è la accoglienza del lettore e la nostra capacità di fare capire quello che stiamo scrivendo: in altre parole stiamo convincendo il lettore di quelle che sono le nostre intenzioni e scopi, e glieli facciamo condividere...

A questo punto non può non continuare a leggerci, che è appunto il nostro obiettivo, specie se parliamo di cose tecniche che a volte possono non sembrare affascinanti.

Nel seguito tenderemo a mescolare molti consigli anche a livello molto diverso, in modo da affastellare i suggerimenti, spesso cercando di mantenerli vicini a dove possono servire, ma facendo senza ritegno salti e passaggi di livello molto diversi (e cross-cutting), passando dagli obblighi ortografici, ai consigli sintattici, fino ai suggerimenti semantici. A volte alcuni sono opinabili ma possono servire a farvi un quadro e a decidere in modo autonomo.

Per lo stile usato in questo documento,
i suggerimenti in Arial, il testo fittizio della tesi in Times.

In rosso, i consigli all'utente.

In blu, il metallivello per la organizzazione.

In viola, i principi del metallivello.

Ad esempio, partiamo dallo strumento di word processing: usate se possibile uno strumento di supporto WYSIWYG (What You See Is What You Get) per vedere immediatamente gli effetti finali di quello che state scrivendo.

Anche se siete appassionati di ambienti esoterici e molto difficili da usare, state introducendo complessità e usando risorse che vi servono per la stesura della tesi.

Per favore, usate gli spell checker per migliorare la ortografia e non scrivere parole sbagliate: **in italiano gli accenti non si scrivono quasi mai, ma per favore è si scrive con l'accento giusto, diversamente dai pochi poiché e benché (è lo è anche al maiuscolo: È).**

Allo stesso modo mettiamo gli apostrofi giusti e non esageriamo, tipo qual'è il principio (giusto qual è).

Per le parole chiave,

inseritele dal generale al particolare, come ad esempio:

*Parole chiave: **Sistemi distribuiti**
 Codice Mobile
 Agenti Mobili
 Java
 Il Sistema Pippo*

Per lo stile e font da usare:

Ottimi i set di font con serif (ossia le grazie in italiano, ossia le curvette dei singoli caratteri, che non ci sono in Arial), ma sono presenti in font accettabilissimi in tesi o documenti scritti come in Times New Roman.

Inserite pure bold (grassetto) ed italico (magari non insieme), ma non sottolineature: cerchiamo di economizzare sul set di variazioni nel testo, sia come font, come tipo di caratteri, come grafica, come stile.

Si raccomanda di usare il giustificato dovunque (piuttosto che il solo allineato a sinistra).

Magari non fate troppi cambiamenti di stile per non produrre effetti che stancano il lettore.

La tesi deve essere ispirata al principio del 'NON stanchiamo il lettore', ossia riconosciamo al lettore delle difficoltà e cerchiamo di facilitarlo attraverso una gestione economica della varietà in ogni senso, grafico, ortografico, grammaticale, di stile, e in ogni aspetto, mantenendo solo le difficoltà necessarie,

Il testo è diviso in

- capitoli (1° livello di organizzazione),
- sezioni (2° livello di organizzazione),
- poi in sottosezioni (3° livello di organizzazione)

si consiglia di non andare troppo oltre il terzo livello - poi in paragrafi che strutturano il testo.

Organizzate anche i titoli delle sezioni ai diversi livelli con stili opportuni differenziati in modo da dare la migliore idea anche grafica di organizzazione.

Un paragrafo è un insieme di periodi che fanno riferimento ad una unità logica ed è costituito da un insieme di frasi correlate che stanno insieme bene, In genere, le metteremo insieme in una sequenza di testo che viene mantenuta anche graficamente insieme. Come in questo paragrafo, e non come nei precedenti che sono frasi singole isolate.

Per la struttura sintattica e semantica:

Devono esserci una Introduzione ed una Conclusione molto generali (di due pagine al massimo l'una).

Le due parti (Introduzione e Conclusioni) devono essere senza dettagli implementativi. Obiettivo è di fare capire a chi legga solo queste due parti quali sono state le motivazioni del lavoro e cosa abbiamo raggiunto.

La Introduzione deve essere scritta come se lo avessimo fatto prima di cominciare il lavoro di tesi e deve indicare come vogliamo muoverci (naturalmente la scriviamo per ultima).

La Conclusione deve riportare cosa abbiamo imparato nel lavoro di tesi, ma senza dettagli; se eventualmente avessimo scoperto cose importanti e direzioni fondamentali, sarebbero stati scritti prima delle pagine conclusive. Sicuramente inseriamo alcune direzioni di possibili sviluppi futuri.

I capitoli normali sono organizzati dal generale (il primo) al particolare (l'ultimo), ricordandoci che non mescoleremo mai dettagli e principi.

Ogni singolo capitolo deve avere una parte iniziale che lo giustifica brevemente ed una parte di finale di conclusioni (senza esagerare!).

I primi capitoli sono generali e creano il background.

I successivi usano le informazioni date nei precedenti e entrano in dettagli, di scelte di progetto, di implementazione, di ... con gradi di dettaglio ulteriore, fino al massimo dettaglio, secondo necessità.

Per ogni sezione o paragrafo, andiamo dal generale al particolare, dando anche delle connessioni con le parti più vicine e correlate. Ogni cosa che diciamo è motivata dall'uso successivo (non diciamo cose che non servono), e non possiamo parlare di qualcosa, che non abbiamo preparato (l'organizzazione è molto importante).

In genere, la nostra tesi scritta (o ancora da scrivere) quindi è una sequenza molto ben organizzata di capitoli, e una sequenza attesa, in una sorta di crescendo per arrivare a dare tutta la nostra competenza su un argomento specifico, su cui dobbiamo trasferire la nostra conoscenza.

Deve in generale essere il più possibile auto-contenuta e permettere di essere letta e di dare informazione anche a chi ne sa poco.

Un tesi di Ingegneria può partire con alcune assunzioni di argomenti noti (ad esempio quelli trattati nei corsi). Anche per alcuni di questi compiremo delle selezioni e spesso faremo dei riferimenti ad altri contenuti bibliografici.

In questa logica, in una tesi scientifica, dove possibile, non ci sono note a più di pagina e si risparmia sulle appendici: o le cose sono significative ed importanti e sono nel testo, ben incluse ed organizzate, o non ci sono per niente, perché quanto meno irrilevanti.

Se alcune parti di background possono diventare dominanti, possiamo fare una Appendice, e nella tesi, possiamo non ripartire da zero. Ad esempio, se ci servono alcune informazioni laterali sull'argomento PLUTO, che, per il resto, non è rilevante, possiamo sintetizzarlo in una Appendice PLUTO. **Non esageriamo con le appendici.**

Naturalmente non mettiamo in appendice parti che sono molto rilevanti: se il ClassLoader di Java è il nostro argomento, non lo metteremo mai in appendice, neanche la descrizione dei dettagli infimi di funzionamento, ma ne parleremo prima ad alto livello, poi con maggiore organizzazione e livello di dettaglio.

Progetto della struttura

Ovviamente decidere una struttura è difficile e dipende dal progetto che ci ispira e dagli obiettivi che talvolta sono chiari dall'inizio di un lavoro, ma talvolta diventano chiari e più motivati, una volta che stiamo procedendo nel lavoro (il tutto si applica allo scritto).

Non escludiamo i casi in cui non sono chiari anche alla fine :-(
Comunque, un impianto ragionato e ben fatto aiuta anche questi casi (speriamo rari).

(non usiamo emoticon o uno stile troppo scherzoso e leggero.

Evitiamo stili troppo personali: lunghi periodi di: "io ho fatto questo e io quello" stancano il lettore, così come non facciamo grandi domande: "cos'è il Web?", specie nei titoli di sezioni e sottosezioni.

Se ci piace, adottiamo uno stile piano e tecnico, senza svolazzi (ci sono consentiti solo i serif e le grazie))

Ricordiamo che stiamo scrivendo una tesi tecnica e quindi quello che dobbiamo fare risaltare è la nostra capacità di organizzazione e di esprimere con ordine problemi complessi e strategie di soluzione altrettanto difficili, ma rese semplici dalle nostre capacità.

Notiamo che è bene non sottovalutare la attività di scrittura e non tutta la scrittura della tesi si deve differire al termine del lavoro, anzi un piano ragionato di scrittura facilita il lavoro ed aiuta il suo progresso.

Anche gli strumenti aiutano:

un word processor ci consente di non occuparci dei dettagli dell'indice, se abbiamo definito gli stili in modo opportuno: l'indice viene generato in ogni momento con i corretti numeri di pagina e con i link al testo necessari. Inoltre, può aiutarci nella numerazione delle figure, nei riferimenti alle stesse, nei riferimenti bibliografici, ecc.ecc.

L'indice, che faremo fino dai primi passi del lavoro, ci rappresenta continuamente quali sono gli argomenti che riteniamo importanti e ci conforta sul lavoro svolto e da svolgere.

- Esso sia organizzato e dettagliato.
- Possiamo ricorrere a capitoli, sezioni, sottosezioni, ecc. fino ai paragrafi, senza andare oltre un certo livello di innestamento.
- Vale la pena di usare gli stili di Word e di produrlo automaticamente.

Indice

INTRODUZIONE.....	I
...	
CAP.3 SISTEMI DISTRIBUITI: VERSO LA MOBILITÀ	2
3.1 SISTEMI DISTRIBUITI.....	2
3.1.1 <i>Modelli di comunicazione</i>	3
3.1.1.1 Modello cliente/servitore.....	4
3.1.1.2 Modelli basati sulla mobilità del codice.....	5
3.1.1.3 Agenti.....	7
3.1.1.3.1 Infrastrutture ed applicazioni basate sugli agenti.....	9
3.1.1.3.2 Comunicazione tra agenti.....	12
3.1.1.3.3 Linguaggi per la comunicazione inter-agente.....	16
3.2 MOBILITÀ DEL CODICE	20
3.2.1 <i>Paradigmi basati sulla mobilità del codice</i>	21
3.2.2 <i>Problemi legati alla mobilità del codice</i>	23
3.3 PARADIGMA AD AGENTI MOBILI.....	24
3.3.1 <i>Mobilità degli agenti</i>	25
3.3.1.1 Mobilità dello stato.....	27
3.3.1.2 Mobilità del codice.....	28
3.3.1.3 Gestione dello spazio dei dati.....	30
3.3.2 <i>Identificazione</i>	31
3.3.3 <i>Comunicazione tra agenti</i>	32
3.3.4 <i>Sicurezza</i>	33
3.3.4.1 Riservatezza ed integrità.....	35
3.3.4.2 Autenticazione.....	36
3.3.4.3 Autorizzazione e controllo degli accessi.....	37
3.3.5 <i>Caratteristiche dei linguaggi per la programmazione di agenti</i>	39
3.3.6 <i>Linguaggi per la programmazione di sistemi ad agenti</i>	42
CAP.4 JAVA: UN LINGUAGGIO PER SISTEMI AD AGENTI.....	56
...	
CAP.7 MOBILE DATA PIPPO: UNA INFRASTRUTTURA PER INFORMAZIONI DISTRIBUITE.....	67
CAP.8 MOBILE DATA PIPPO: IMPLEMENTAZIONE.....	98
CONCLUSIONI.....	138
BIBLIOGRAFIA	140

Stile

- **La Tesi deve essere scritta con stile tecnico, senza troppi svolazzi, ma con l'intento di dare informazioni significative per chi debba affrontare argomenti simili.**
- **Non si usano stili da articolo, che assumono conoscenza pregressa e spesso ammiccando ad un lettore già esperto, ma si deve privilegiare la completezza e l'auto-contenimento degli argomenti interessanti per il lavoro: ogni informazione rilevante, che non sia elementare deve essere spiegata in modo il più possibile completo.**
 - Non usiamo domande retoriche, né nel testo né nei titoli ed indici, ma cerchiamo di fare capire bene quello che vogliamo dire.
 - Per ogni cosa, ci deve essere, prima, una definizione e, dopo, tutto quello che è necessario per capire meglio le specificità, fino a un giudizio sulla tecnologia, tecnica, modo, che stiamo descrivendo.
- **Lo stile deve prevedere una continua organizzazione del contenuto: il primo compito che abbiamo è di organizzare bene il contenuto, partendo dalle basi fino ad arrivare al corpo centrale ed ai dettagli. Questo richiede anche che si decida per ogni elemento quale sia la sua migliore collocazione a secondo della importanza e rilevanza per la tesi stessa.**
 - Ogni cosa che diciamo deve essere parte della organizzazione.
 - Se qualcosa non serve, deve essere tagliata (e se importante riferita solamente).
 - Tutto quello che è necessario in un certo punto, deve essere già stato introdotto in precedenza.
- **In caso riferiamo parti ed opinioni di altri, facciamole avvertendo che stiamo riferendo proprio queste: diamo credito a chi ci ha aiutato e riportiamo le informazioni (magari con “”) ed i nomi.**

Ortografia

- **Le frasi (terminate da un punto) possono essere organizzate in periodi (o paragrafi) che contengono una serie di frasi logicamente correlate. Al termine del periodo (paragrafo) possiamo andare a capo, producendo un magnifico effetto di visualizzazione sullo scritto. A volte possiamo prevedere o indentazione, ossia rientranza della prima riga del paragrafo (indentation o indentazione), o una interlinea maggiore inter-paragrafo rispetto a quella intra-paragrafo.**
- **Nello scritto, ogni segno di interpunzione - virgole, punto, punto e virgola, due punti, etc. - sono preceduti dall'ultimo carattere della parola precedente, senza bianchi, e sono seguiti da almeno un bianco.**
- **Per le sigle, la prima volta che le citiamo devono essere definite per esteso e solo dopo possiamo usarle 'liberamente' nel testo. NON usiamo troppe sigle altrimenti le tesi diventa un cifrario illeggibile (non siamo anglosassoni: le sigle servono solo se giustificate dall'uso frequente).**

Introduzione

La società moderna viene molto spesso definita “Società dell’informazione” sottolineando l’importanza dell’informazione, quale elemento fondamentale delle nostre attività, lavorative e private. L’informazione, se puntuale e precisa, permette di compiere al meglio qualunque processo decisionale, consentendo di raggiungere gli obiettivi desiderati.

L’importanza dell’informazione è cresciuta da quando essa è disponibile per tutti in grande quantità, grazie alla diffusione di Internet, che consente un accesso facile e veloce ad una enorme mole di dati. Ciò spinge sempre più persone ad usufruire di questo mezzo.

....

Nel Cap. 1 vengono introdotte le problematiche relative alla gestione di una serie di fonti di dati distribuite. Nel Cap. 2 si descrivono i servizi auspicati in un ambiente per la ricerca di informazione. Nel Cap. 3 vengono presentati i modelli di computazione distribuita basati sulla mobilità del codice. Nel Cap. 4 si analizzano le caratteristiche di Java come linguaggio per l’implementazione di Sistemi ad Agenti Mobili, e nel Cap. successivo viene presentato il sistema S.O.M.A. Il Cap. 6 illustra le possibilità disponibili in Java per l’accesso a database relazionali. Nel Cap. 7 si descrive l’architettura di un’infrastruttura per la ricerca di informazione basata sul modello ad agenti mobili. Il Cap. 8, infine, descrive nel dettaglio l’implementazione di un prototipo di tale infrastruttura.

La Introduzione contiene anche un sommario dei capitoli successivi e non eccede la pagina, pagina e mezzo.

I primi capitoli sono quelli che devono fornire il background: sono generali e evidenziano le aree di interesse ed i problemi, oltre che categorizzare eventuali strategie note di soluzione.

In prima approssimazione, la metà dei capitoli sono dedicati a costruire il background e gli elementi necessari a capire di cosa vogliamo parlare, gli altri a descrivere il reale nostro lavoro.

Cap.1

Fonti di informazione in un sistema distribuito

Le risorse presenti all'interno della rete sono di vario tipo. A noi interessano quelle risorse la cui principale caratteristica è consentire l'accesso ad un insieme di dati, di qualunque tipo. Esse verranno nel seguito chiamate "risorse informative".

Le risorse informative sono la fonte delle nostre ricerche. Studieremo le loro caratteristiche principali, senza mai perdere di vista il nostro obiettivo: estrarre da esse informazione rilevante in modo efficiente.

...

1.1 Eterogeneità dell'informazione

L'informazione è presente sulla grande rete in varie forme, dalle pagine del World Wide Web alle news di Usenet, ed ogni diversa forma presenta le proprie caratteristiche distintive, per quanto riguarda contenuto, formati di rappresentazione e metodi di accesso e ricerca.

...

1.1.1 Eterogeneità della rappresentazione e modelli semantici

Il nostro scopo è nascondere ...

In **Figura 1** è mostrato uno schema delle interazioni tra applicazioni e risorse, mediate dall'uso di un modello semantico dei dati.

La **Figura 1** intende anche mostrare come le interazioni tra i diversi enti messi in gioco avvenga in modo del tutto indipendente dagli schemi che si possono applicare nei settori ...

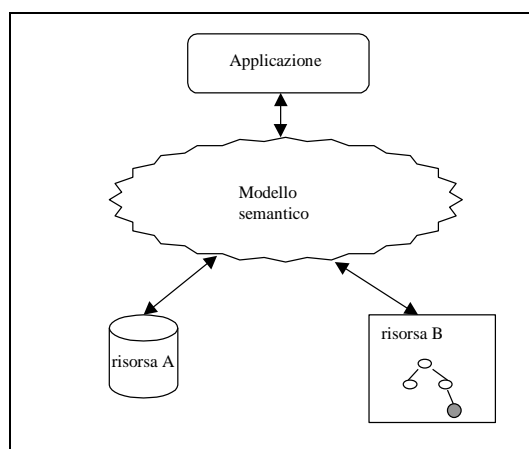


Figura 1 Il modello semantico dei dati media le interazioni tra applicazioni e risorse rendendo ininfluente la struttura ed i metodi di accesso all'informazione

- **Le figure hanno tutte una legenda che ne spiega la necessità e la utilità; sono anche sempre riferite nel testo e descritte opportunamente nel testo stesso.**
- **Curiamo anche l'aspetto grafico delle cose scritte nelle figure e tabelle stesse.**
- **Non facciamo troppi sottoparagrafi e liste troppo lunghe.**

1.1.1.1 Modello Entity – Relationship

Questo modello nasce per la descrizione concettuale di basi di dati. Viene utilizzato nelle prime fasi della progettazione, per poi essere trasformato negli schemi definitivi. Proprio perché di tipo concettuale, il modello è uno strumento adattabile alla specifica di modelli semantici ed alla descrizione delle informazioni attraverso alcuni concetti fondamentali:

- **Entità:** classificano gli oggetti del dominio. Gli oggetti di una classe hanno alcune proprietà comuni che caratterizzano l'entità stessa. Gli oggetti vengono detti istanze (o occorrenze) dell'entità. Ogni entità ha un nome che la identifica.
- **Relazioni:** rappresentano un legame logico tra due o più entità. Sono identificate da un nome. Possiedono una cardinalità che individua numeri minimi e massimi di legami che un singolo oggetto può possedere.
- **Attributi:** sono le proprietà assegnabili alle entità (e dunque a tutte le loro istanze) e alle relazioni. Ogni attributo ha un nome identificativo, un valore, un tipo che stabilisce il dominio dei valori ammissibili e dei vincoli quali la cardinalità (cioè il numero di valori che un attributo può assumere contemporaneamente).
- **Generalizzazioni:** rappresentano legami logici tra una entità padre e una o più entità figlie. L'entità padre è più generale delle entità figlie, nel senso che le comprende come caso più generale. Le entità figlie sono specializzazioni dell'entità padre.

Lista di elementi nel testo.

Le liste hanno anche un supporto dedicato nei word processor.

Non abbondate con le liste: liste con un numero molto elevato di elementi, singolarmente anche lunghi, rischiano di fare perdere l'orientamento al lettore.

In genere, un depliant può avere anche liste di 30 elementi, in genere impressionanti e schiacciati caratteristiche positive di quello che vogliamo piazzare, ma non uno scritto tecnico.

Una tesi, riguardo alle liste, puntate, numerate, o con altre forme di separatore:

- limita il numero degli elementi;
- limita la lunghezza dei singoli elementi (la lista deve essere visibile tutta in una pagina);
- organizza la lista in modo ragionato, magari raggruppando gli item e determinando classificazioni più articolate;
- prevede per ogni elemento la corretta punteggiatura e ortografia (; come separatori e minuscole iniziali, . come separatori e maiuscole nelle iniziali, ecc.);
- prevede sempre liste del tutto omogenee di elementi.

Una lista è un tutto uno e deve essere esteticamente omogenea: se un elemento chiave è un sostantivo, tutti gli altri devono avere la stessa forma; se un elemento è una frase completa, tutti lo siano, ecc.

Se listiamo proprietà come un sostantivo che spieghiamo dopo un :, tutti gli elementi ricevano lo stesso trattamento e rispettino il principio del ‘non stanchiamo il lettore’.

Rispettiamo dove possibile anche la uniformità in indentation e in stile ortografico.

Ad esempio una lista troppo lunga, sviluppata su due pagine e non visibile a colpo d’occhio non rispetta le specifiche, e deve essere modificata.

1.1.1.2 Confronto tra i modelli presentati

Per capire meglio somiglianze e differenze tra i modelli presentati nella **Tabella 1** viene mostrato un quadro riassuntivo degli esempi mostrati.

<i>Modello</i>	<i>Rappresentazione</i>
LOOM	<pre>(defconcept Comparsa :is (and Attore (exactly 1 tariffa_oraria) (> tariffa_oraria 0))) (defrelation tariffa_oraria :domain comparsa :range number :characteristics :closed-world))</pre>
Ontolingua	<pre>(define-class Comparsa (?comparsa) "Una comparsa è un attore che fa brevi apparizioni, poco importanti, all'interno di un film" :def (and (Attore ?comparsa) (= (value-cardinality ?comparsa tariffa_oraria) 1) (value-type ?comparsa tariffa_oraria number) (> (tariffa_oraria ?comparsa 0))))</pre>

KEE	Unit: Comparsa Superclasses: Attore Member Of: CLASSES Member Slot: tariffa_oraria ValueClass: (MEMBER.OF NUMBER) Min.Cardinality: 1 Max.Cardinality: 1
-----	---

Tabella 1. Confronto tra gli strumenti utilizzabili per rappresentare modelli semantici

Le tabelle hanno tutte una legenda che ne spiega la necessità e la utilità; sono anche sempre riferite nel testo.

Devono essere sistemate su una unica pagina (NON COME QUESTA SOPRA O LA TABELLA 2)

Si devono citare i riferimenti bibliografici ai lavori consultati, sia articoli, sia libri, che sono serviti e possono servire di riferimento: i riferimenti possono essere inseriti nel testo come [Pia98].

1.1.1.3 XML, un linguaggio per la rappresentazione dell'informazione

XML (eXtensible Markup Language) [Pia98] nasce anch'esso come applicazione di SGML, ma con con l'obiettivo di superare i limiti di HTML. I problemi principali di HTML sono la sua chiusura nella definizione dei marcatori ed il fatto di mescolare nello stesso contenitore la definizione dell'informazione ed il modo in cui debba essere rappresentata. XML supera entrambi questi limiti: esso infatti permette la definizione di un qualsiasi numero di marcatori e separa nettamente la definizione dei dati dal modo in cui questi debbano essere rappresentati.

Lo standard XML scinde in 3 elementi separati i seguenti aspetti: la definizione della sintassi di un documento, il documento vero e proprio, il modo in cui il documento deve essere rappresentato. La possibilità di estendere a piacere il numero dei marcatori è dovuta all'uso dei DTD (Data Type Definition). Un DTD definisce in maniera completa la sintassi di un'applicazione XML, cioè l'insieme dei marcatori ed il modo in cui possono essere utilizzati. In questo modo chiunque può definire un nuovo formato di dati scrivendone il DTD. Un documento XML dovrà, per prima cosa, specificare il DTD che usa, in modo da poter essere correttamente utilizzato.

L'ultimo aspetto di un documento XML è la sua visualizzazione. Essa viene descritta mediante fogli di stile (secondo lo standard XSL), che specificano come debba essere rappresentato il contenuto di ogni marcatore, a seconda del dispositivo di rappresentazione (il monitor, una stampante, un dispositivo di lettura braille, ecc.). Immaginiamo dunque di definire un DTD per la rappresentazione di oggetti ...

<i>Livello</i>	<i>Dirty reads</i>	<i>Non-repeatable reads</i>	<i>Phantom reads</i>
READ COMMITTED	Sì	No	No

READ UNCOMMITTED	No	No	No
REPEATABLE READ	Sì	Sì	No
SERIALIZABLE	Sì	Sì	Sì

Tabella 2. Livelli di isolamento

1.1.2 UserAgent

L'agente UserAgent rappresenta un profilo utente. Ogni profilo utente è specifico per un dato utente ed un dato modello semantico. L'agente è in grado di memorizzare l'elenco delle interrogazioni sottoscritte e di gestirle interagendo con un ActivityManager. L'agente è in grado di risiedere presso l'utente quando questo sta interagendo con il sistema e presso l'ActivityManager quando l'utente risulta scollegato. In questo modo è in grado sia di rispondere in maniera pronta all'utente che di minimizzare l'uso delle risorse di rete quando questo non è presente.

L'interazione con l'utente avviene mediante l'uso di un oggetto condiviso che implementa l'interfaccia **UserInteractInterface**. Tale interfaccia prevede i seguenti metodi:

```
public void addQuery(LanguageCommand query, long interval);
public void removeQuery(Ticket ticket);
public QueryState getResult(Ticket ticket);
```

In questo modo risulta possibile sottoporre nuove sottoscrizioni, cancellarne di esistenti o conoscerne il risultato corrente.

Lo schema di funzionamento dell'agente è il seguente:

```
while(<l'utente è collegato>) {
    <ricevi i comandi dall'utente e memorizzali>
}
goToPlace(); // spostati presso l'ActivityManager
checkPlace(); // ottieni l'interfaccia di comunicazione con l'ActivityManager
updatePlace(); // aggiorna lo stato dell'utente aggiungendo o rimuovendo le
    sottoscrizioni secondo i comandi ricevuti dall'utente
while(<l'utente è scollegato>) {
    checkMail(); // attendi l'arrivo dell'utente e nel frattempo raccogli gli
    aggiornamenti delle sottoscrizioni
}
goToUser(); // torna dall'utente e portando con te i risultati delle
    sottoscrizioni richieste dall'utente
```

Se riportiamo del codice, non siano pagine e pagine che sono assolutamente illeggibili e non si possono seguire e leggere. Si devono riportare solo alcune parti di codice significative (che non possono eccedere la dimensione di una pagina sicuramente). Il codice riportato deve essere quindi essenziale, commentato e introdotto in modo comprensibile. Si consiglia anche di riportare qualche riferimento ai numeri di linea # e potere commentare nel testo della tesi le diverse parti e le loro caratteristiche: ovviamente dove possibile il codice riporti dei commenti in linea per spiegare le cose non ovvie.

Merita, se vale la pena di riportare il codice stesso, anche una parte di spiegazione che introduca il codice, e una parte che ne spieghi il significato.

Se è necessario riportare interi pezzi di codice, magari interi componenti o package, allora si devono metterli o in appendice, o, meglio, in un allegato come floppy o CD.

È meglio che anche le parti di codice abbiano una loro legenda e siano anche riferite nel testo.

...

1.2 Applicazione Utente

L'applicazione utente consente l'interazione degli utenti con il sistema in modo semplice e trasparente. Una volta avviata l'applicazione richiede l'inserimento del nome dell'host e della porta di ascolto di un AccessAgent (Figura 2).

...

Curiamo la versione finale del documento in termini estetico grafici:

Non lasciamo spazi lunghi bianchi nella versione finale, ma cerchiamo di fare scorrere il testo, in modo sensato.

Non lasciamo titoli di sezioni separate dal contenuto delle sezioni (vedove ed orfani – vedi subito sotto), ecc.ecc.

1.3 Altre applicazioni

Tipicamente richiediamo che i titoli di sezione siano lasciati insieme al testo successivo (vedi word processor).

Le figure sono parte del testo, e prima e dopo ne prevedono: possibilmente non sono né all'inizio, né alla fine della sezione.

Le figure sono sempre descritte nel testo della tesi: diamo con il testo scritto l'intero contenuto; figure, didascalie, tabelle, ... danno un contenuto di aiuto, che non sostituisce il testo stesso.

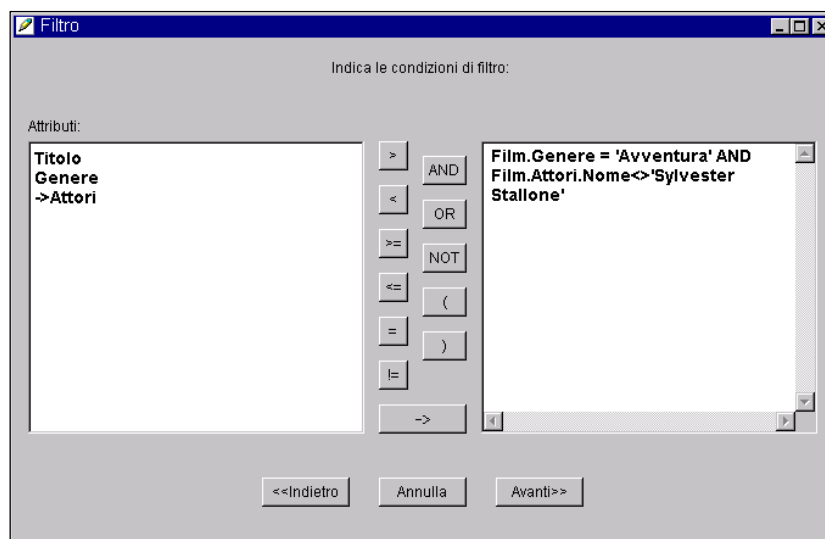


Figura 2. Terza schermata del wizard di inserimento delle interrogazioni: indicazione del filtro di selezione

Non eccedete con figure di GUI e modi di uso: magari, se non se ne può fare a meno, vale la pena di fare una appendice come un manuale di uso di una applicazione di un componente.

...

1.4 Risultati

Per verificare l'efficacia del nostro progetto abbiamo effettuato alcune sessioni di test, che riuscissero ad evidenziare le reali prestazioni del prototipo realizzato. Abbiamo individuato due tipi di test. Il primo intendeva evidenziare l'utilità del sistema di cache per raggiungere buoni livelli di efficienza. Il secondo mette a confronto la soluzione ad agenti mobili con una di tipo client/server, per evidenziare i vantaggi che il primo modello introduce.

Quando riportiamo i necessari risultati, dobbiamo descrivere in modo chiaro non solo quali ipotesi ci hanno guidato, ma anche quali architetture abbiamo considerato (sia risorse hardware, sia software) con la massima precisione possibile.

Per eseguire i test abbiamo utilizzato una serie di 5 PC, così configurati:

- **PC1:** Pentium 133Mhz, 80Mb RAM
- **PC2, PC3, PC4, PC5:** Pentium 166MMX, 64Mb RAM

PC2, PC3, PC4 e PC5 erano connessi mediante una LAN di tipo Ethernet a 10 Mbit/sec., mentre PC1 era connesso a questa rete mediante un collegamento diretto via modem a 28.800 bps collegato a PC2. Tutti i PC eseguivano il Sistema Operativo Windows NT Workstation 4.0 (con applicato il Service Pack 4).

Abbiamo individuato una serie di risorse informative, costituite da database relazionali in formato Access 97, e costruito un modello semantico per descriverle. A questa base sono stati applicati tre differenti filtri:

1. WHERE Film.Genere='Western'
2. WHERE Film.Genere='Cartoni_animati'
3. WHERE Film.Genere='Fantascienza'

Le interrogazioni sono state eseguite in tre modi diversi:

- nella configurazione LAN, dove le prestazioni del sistema ad agenti mobili e di quello client/server, con dimensioni limitate, si equivalgono (**T1**)
- nella configurazione WAN con il sistema ad agenti mobili (**T2**)
- nella configurazione WAN con la simulazione client/server (**T3**)

Le misure sono state ripetute tre volte, abilitando di volta in volta, una due o tutte e tre le risorse informative contenenti dati sugli Attori (Risorsa 2, Risorsa 3 e Risorsa 4). Nelle tabelle seguenti vengono mostrati i risultati di tali misure:

Query	N. oggetti risultato	Dim. Risultato (byte)	T1 (sec)	T2 (sec)	T3 (sec)
1	85	4.2K	11.51	13.02	38.13
2	140	7K	12.33	14.98	39.43
3	495	24.7K	32.85	40.21	62.58

Tabella 3. Risultati del test con la sola **Risorsa 2** abilitata

Query	N. oggetti risultato	Dim. Risultato (byte)	T1 (sec)	T2 (sec)	T3 (sec)
1	85	4.2K	19.12	21.35	72.12
2	140	7K	25.34	28.46	79.78
3	495	24.7K	49.71	58.12	105.20

Tabella 4. Risultati del test con **Risorsa 2** e **Risorsa 3** abilitate

Query	N. oggetti risultato	Dim. Risultato (byte)	T1 (sec)	T2 (sec)	T3 (sec)
1	85	4.2K	26.36	28.45	105.28
2	140	7K	39.06	41.03	117.41
3	495	24.7K	72.95	82.18	154.15

Tabella 5. Risultati del test con tutte le risorse abilitate

Se dovete riportare molti dati e relative tabelle, dove possibile, compattatele e riportatele insieme in modo da stimolare una discussione congiunta.

I risultati sono riportati anche in alcuni grafici. Di seguito vediamo l'andamento di T1, T2, T3 al variare della dimensione del risultato nei tre casi:

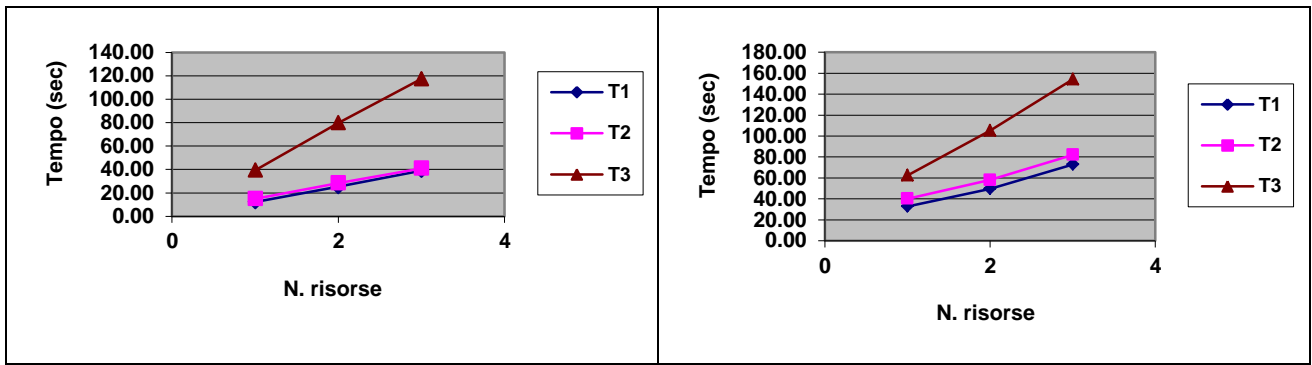


Figura 5. Grafici dei tempi di esecuzione in funzione del numero di risorse coinvolte

Tenete insieme i componenti correlati:

ad esempio, le figure e la legenda; la riga finale di un paragrafo; i pezzi unici di codice; etc.

Fate controllare al word processor di tenere insieme i paragrafi (se usate Word, mettete a posto i paragrafi tenendo insieme le righe).

Conclusioni

Le conclusioni:

- **devono essere molto generali e non riportare cose nuove,**
- **si possono eventualmente accennare a nuove direzioni di sviluppo del lavoro svolto (sviluppi futuri),**
- **non eccedono le due pagine.**

Prima del capitolo di conclusioni, fate sempre delle conclusioni più tecniche del lavoro fatto: una sorta di ultime considerazioni prima delle conclusioni che mostrino quello che avete imparato.

Il nostro lavoro ha cercato di evidenziare l'utilità del modello ad agenti mobili applicato al problema della ricerca di informazioni all'interno di risorse eterogenee e distribuite su scala globale. È stata sviluppata un'infrastruttura che, sfruttando il supporto S.O.M.A. ad agenti mobili, realizza servizi di automatizzazione delle interrogazioni in funzione di preferenze specificate dagli utenti.

Lo studio delle cache per l'XML è ancora agli inizi, anche se molto lavoro è stato effettuato nel campo delle MANET (Mobile Ad-hoc NETWORK), reti capaci di operare, organizzarsi e coordinarsi senza una struttura predefinita, tipicamente connesse attraverso tecnologie wireless e delle applicazioni per clienti nomadi, in grado di fornire servizi indipendentemente dalla locazione e dal dispositivo utilizzato dall'utente.

Non sono da escludere nel futuro ulteriori evoluzioni relative alle tecnologie Web Service: molte ricerche sono attualmente dirette all'introduzione di aspetti semantici nel WSDL tramite annotazioni ed è prevedibile che vengano elaborati protocolli ad-hoc per la gestione delle cache all'interno delle specifiche WS-*, visto l'interesse dimostrato da molti enti e aziende che operano nel campo IT verso queste tematiche.

Con la presente tesi si è voluto sperimentare l'applicabilità delle strategie di caching dell'HTML nei server Web all'XML nei Web Service, evidenziandone limiti e potenzialità e prendendo in considerazione per la progettazione del sistema di cache un problema reale, il portale dell'Ateneo di Bologna. Sono stati utilizzati gli standard per l'interoperabilità del mondo Web Service e alcuni dei principi e degli elementi alla base di SOA.

...

La flessibilità e configurabilità dell'applicazione rendono il sistema utilizzabile anche in contesti più ampi, con un maggior numero di servizi da gestire e un'utenza più estesa ed eterogenea, non soltanto a livello accademico ma anche nell'ambito della pubblica amministrazione e dell'industria.

Bibliografia

- **La bibliografia si deve riferire dove necessario nel testo.**
 - **Si deve usare una forma omogenea per i riferimenti che consenta di ritrovare i contributi su libri, riviste, e anche URL. Si possono riferire anche documenti Web; anche in questo caso, usiamo una forma omogenea di notazione, e manteniamola per tutti i riferimenti. Ad esempio, usando la forma [InizAutoreAnno] che possono essere inserite e riferite nel testo e sono listate in questa parte in ordine alfabetico.**
- I vantaggi di questo formato sono che si possono fare i riferimenti senza cambiarli all'ultimo momento, continuando ad aggiungere fino alla fine.**

- [Are93] Y. Arens, C.Y. Chee, C. Hsu, e C.A Knoblock: "Retrieving and integrating data from multiple information sources", International Journal on Intelligent and Cooperative Information Systems, 1993.
- [Fug98] A. Fuggetta, G.P. Picco, e G. Vigna: "Understanding Code Mobility", IEEE Transactions on Software Engineering, 1998.
- [Gon97] L. Gong, M. Mueller, H. Prafullchandra, e R. Schemers: "Going Beyond the Sandbox: An Overview of the New Security Architecture in the Java Development Kit 1.2", in Proceedings of the USENIX Symposium on Internet Technologies and Systems, Monterey, California, Dec. 1997.
- [OBJ98] "ObjectSpace Voyager Core Technology 2.0", <http://www.objectspace.com>
- [Cha98] V.K. Chaudhri, A. Farquhar, R. Fikes, P.D. Karp, e J.P. Rice: "Open Knowledge Base Connectivity 2.0.2 - Proposed", February 1998.
- [Pia98] L. Piatti, e P. Bionda: "Creare un nuovo linguaggio con XML", Computer Programming, Vol. XV, n.68, Aprile 1998.
- [Sch95] B. Schneier: "Applied Cryptography" J. Wiley & Sons, 1995.
- [SOMA98] Sistema progettato nell'ambito del Progetto: "Design Methodologies and Tools of High Performance Systems for Distributed Applications" finanziato dal Ministero dell'Università e della Ricerca Scientifica e Tecnologica (MURST). Reperibile su: <http://lia.deis.unibo.it/Software/MA/>.

Buon lavoro e divertitevi.

**Qualunque contributo o commento a questa guida è benvenuto.
antonio.corradi@unibo.it**