Capitolo 3

Il sistema DEMI

Per poter gestire in modo efficiente e funzionale i problemi computazionali tipici del mondo della fisica delle alte energie, ma anche per "trattare" analoghi problemi propri di altri ambienti, viene proposto il modello di un sistema chiamato **DEMI** (Distributed hEterogeneous coMputing envIronment).

Figura 3.1: Il logo del progetto DEMI

3.1 Requisiti

Le caratteristiche e le funzionalità generali richieste da questo sistema si possono riassumere nei seguenti punti:

- DEMI deve gestire un sistema di tipo eterogeneo, costituito da risorse di calcolo di tipo, architettura e prestazioni differenti, distribuite geograficamente tra più siti;
- DEMI deve essere in grado di soddisfare tutte le richieste computazionali sottomesse dagli utenti, che possono essere di tipo e di caratteristiche diverse;
- le performance complessive del sistema devono essere ottimizzate;

- il sistema DEMI deve apparire a tutti gli utenti come un unico grande ambiente di calcolo;
- DEMI deve essere in grado di gestire grandi quantità di dati, che possono essere distribuiti tra più siti diversi. Questi dati devono comunque essere organizzati come un unico grande database logico. Ogni job sottomesso dagli utenti può in teoria accedere all'intero data set;
- il sistema deve essere scalabile, rispetto alle richieste computazionali e rispetto alle risorse di calcolo: DEMI deve essere in grado di gestire facilmente variazioni (anche sensibili) nel volume complessivo dei dati da elaborare, nel numero di job da eseguire, nel numero di calcolatori disponibili, ecc...;
- DEMI deve essere flessibile, in grado di "adattarsi" facilmente ad eventuali variazioni della configurazione del sistema;
- DEMI deve essere *self-managing*, in grado di gestire autonomamente eventuali failure: l'attività di gestione da parte dell'amministratore del sistema deve essere ridotta al minimo indispensabile;
- il sistema deve rimanere nel suo complesso sempre attivo, nonostante guasti o malfunzionamenti hardware e/o software;
- gli utenti del sistema non devono essere costretti a gestire dettagli implementativi non significativi (ad esempio dove i dati sono fisicamente memorizzati);
- il codice dei programmi che vengono eseguiti utilizzando il sistema DEMI deve essere del tutto "normale": non devono essere necessarie particolari istruzioni o particolari configurazioni, e non devono essere imposti vincoli speciali;
- DEMI deve essere un sistema semplice da usare, anche da parte di utenti non particolarmente esperti;
- DEMI non deve essere un sistema specifico per un singolo esperimento di fisica delle alte energie. DEMI, inoltre, deve essere un sistema "general purpose", utilizzabile anche in settori e discipline diverse dall'ambiente HEP, in cui si devono gestire problemi analoghi.

3.2 Ambiente

DEMI deve essere implementato per un ambiente come quello rappresentato in figura 3.2.

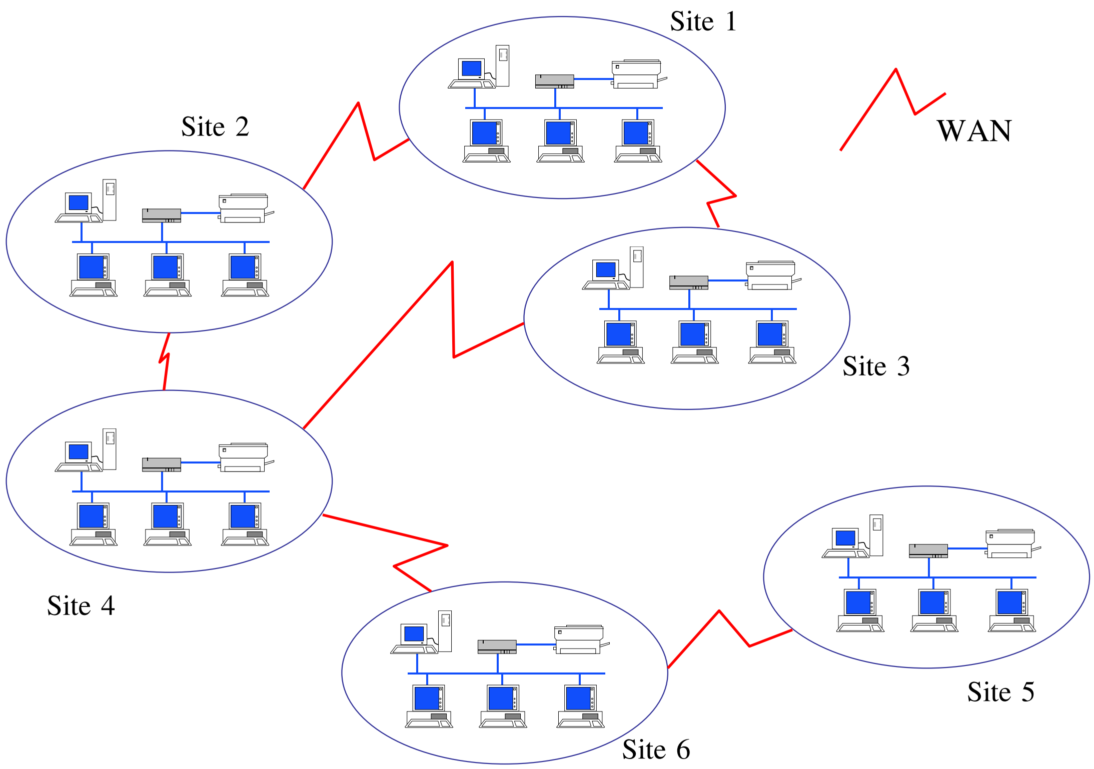


Figura 3.2: Un ambiente tipico, costituito da vari siti connessi attraverso una rete geografica

Deve essere quindi considerato un sistema composto da vari siti, direttamente o indirettamente connessi tra loro attraverso una rete WAN.

In generale i vari siti possono essere molto “diversi” tra loro: possono differire nel numero di computer disponibili, nei modelli di questi sistemi, nell’architettura e nella configurazione della rete locale, ecc.. Ogni sito costituisce un *metasistema*: un sistema di elaborazione distribuito, costituito da sistemi indipendenti ed eterogenei, collegati tra loro attraverso una rete locale, che può essere implementata in diversi modi (Ethernet, Fast Ethernet, FDDI, ATM, Gigabit Ethernet, ecc...).

Si può assumere che la comunicazione tra i vari sistemi, posti nello stesso sito o in siti diversi, è sempre implementata attraverso i protocolli TCP/IP; non è invece richiesto il supporto per altri protocolli (DECnet, SNA, ecc...).

3.2.1 Sistemi supportati

Il sistema DEMI deve essere in grado di gestire:

- Workstation Unix (il sottoinsieme più grande possibile);

- PC e workstation con sistema operativo Windows NT.

Deve comunque essere possibile effettuare facilmente il porting dell’applicazione ad eventuali nuove architetture, con un elevato rapporto prestazioni/prezzo, che si rendessero disponibili. Infatti è fondamentale poter disporre della massima potenza computazionale, utilizzando le risorse economiche disponibili.

Solitamente ogni computer deve essere considerato come un calcolatore “general purpose”, di cui il sistema DEMI può utilizzare solamente i cicli macchina liberi. Quindi i vari calcolatori devono ritenersi disponibili al sistema DEMI solamente quando essi non sono utilizzati dai rispettivi utenti. Si rende quindi necessaria una gestione delle priorità: l’utente di una workstation deve aver sempre la precedenza nell’utilizzo di tale risorsa, rispetto al sistema DEMI. Quando un utente vuole utilizzare la “propria” workstation, gli eventuali job DEMI in esecuzione su questo computer devono essere sospesi, ed eventualmente “migrati” su un altro calcolatore. Questa procedura deve essere effettuata in maniera automatica e trasparente: l’utente non deve essere obbligato ad utilizzare un comando esplicito per “reclamare” l’uso della sua workstation, ma deve essere il sistema a verificare automaticamente la “presenza” dell’utente (ad esempio rilevando l’attività sulla tastiera, controllando quali sono i processi in esecuzione, ecc...).

3.3 Metodologia di sviluppo

Considerando una qualsiasi applicazione software che deve essere sviluppata, i requisiti e le caratteristiche che in genere sono richiesti sono molteplici [28]: affidabilità, robustezza (il software deve comportarsi in maniera adeguata anche in presenza di situazioni anomale), usabilità (l'utente deve essere in grado di utilizzare l'applicazione senza particolari difficoltà), verificabilità (è necessario essere in grado di dimostrare che il sistema realizzato effettivamente risponde alle caratteristiche e alle funzionalità richieste), manutenibilità, riusabilità, interoperabilità (l'applicazione sviluppata deve poter facilmente integrarsi con altre applicazioni, che realizzano particolari funzionalità), ecc...

È chiaro, quindi, che per soddisfare tutti questi requisiti è necessario utilizzare una precisa metodologia nello sviluppo del software, soprattutto nel caso in cui il sistema che deve essere implementato sia particolarmente complesso.

Nella realizzazione del sistema DEMI si è utilizzata la metodologia chiamata *ciclo di vita a cascata* [28]. Questo modello prevede che lo sviluppo dell'applicazione si articoli in una serie di fasi, poste in cascata: ogni fase riceve ingressi dalla fase precedente, e produce risultati che vengono poi usati nella fase successiva. Una volta che sono stati ottenuti i risultati di una fase, e questi sono stati convalidati, si passa alla fase successiva. Il numero e i tipi di fasi in cui si articola lo sviluppo di un sistema non sono definiti a priori, in quanto dipendono da vari fattori: le caratteristiche e la complessità del sistema da realizzare, il ruolo del "produttore" dell'applicazione (può coincidere oppure no con il committente), l'utilizzo che si dovrà fare dell'applicazione (uso esclusivo da parte del committente, o prodotto che potrà essere utilizzato anche da altri utenti), ecc...
Nello sviluppo del sistema DEMI si sono considerate le seguenti fasi iniziali:

- analisi preliminare del problema;
- analisi e specifica dei requisiti;
- progettazione;
- implementazione.

3.3.1 Analisi preliminare del problema

Per definire in maniera chiara e non ambigua il dominio del problema, si è provveduto innanzittutto alla stesura di un **glossario**, in cui vengono definite le "parole fondamentali". Durante lo sviluppo del sistema, tale glossario è stato continuamente aggiornato con i nuovi termini introdotti.

Il glossario è riportato in appendice A.

Si sono poi analizzati alcuni sistemi e applicazioni già esistenti, che realizzano (almeno parzialmente) alcune funzionalità richieste nel sistema considerato. Queste applicazioni sono descritte e analizzate nel capitolo 5 e in appendice B.

3.3.2 Analisi e specifica dei requisiti

Si sono poi definiti in dettaglio le funzionalità e i requisiti dell'applicazione da realizzare, definendo "cosa" deve essere fornito dal sistema.

Esistono diversi modelli per la specifica formale dei requisiti [28]: il modello entità-relazioni, i diagrammi di flusso dei dati, gli automi a stati finiti, le reti di Petri, la logica temporale del primo ordine, ecc... Queste specifiche formali consentono di definire le funzionalità richieste senza ambiguità, e inoltre possono essre "manipolate" automaticamente per valutarne la completezza, per analizzare le relazioni che esistono tra i vari concetti definiti, ecc... Tuttavia per il sistema DEMI si è preferito considerare una specifica di tipo informale, scritta in linguaggio naturale, sostanzialmente per garantire la leggibilità anche da parte di persone che non hanno specifiche competenze in materia.

Il risultato della fase di analisi e specifica dei requisiti del sistema DEMI, è il cosiddetto documento URD (*User Requirement Document*), presentato in [1]. I requisiti del sistema DEMI sono comunque descritti nel capitolo 4.

3.3.3 Progettazione

All'analisi e specifica dei requisiti è seguita la fase di progettazione, cioè le attività che consentono di realizzare effettivamente l'applicazione. Oltre a definire l'architettura software del sistema, si è data particolare enfasi al problema della gestione dei dati.

La progettazione del sistema DEMI è presentata nei capitoli 6 e 7 .

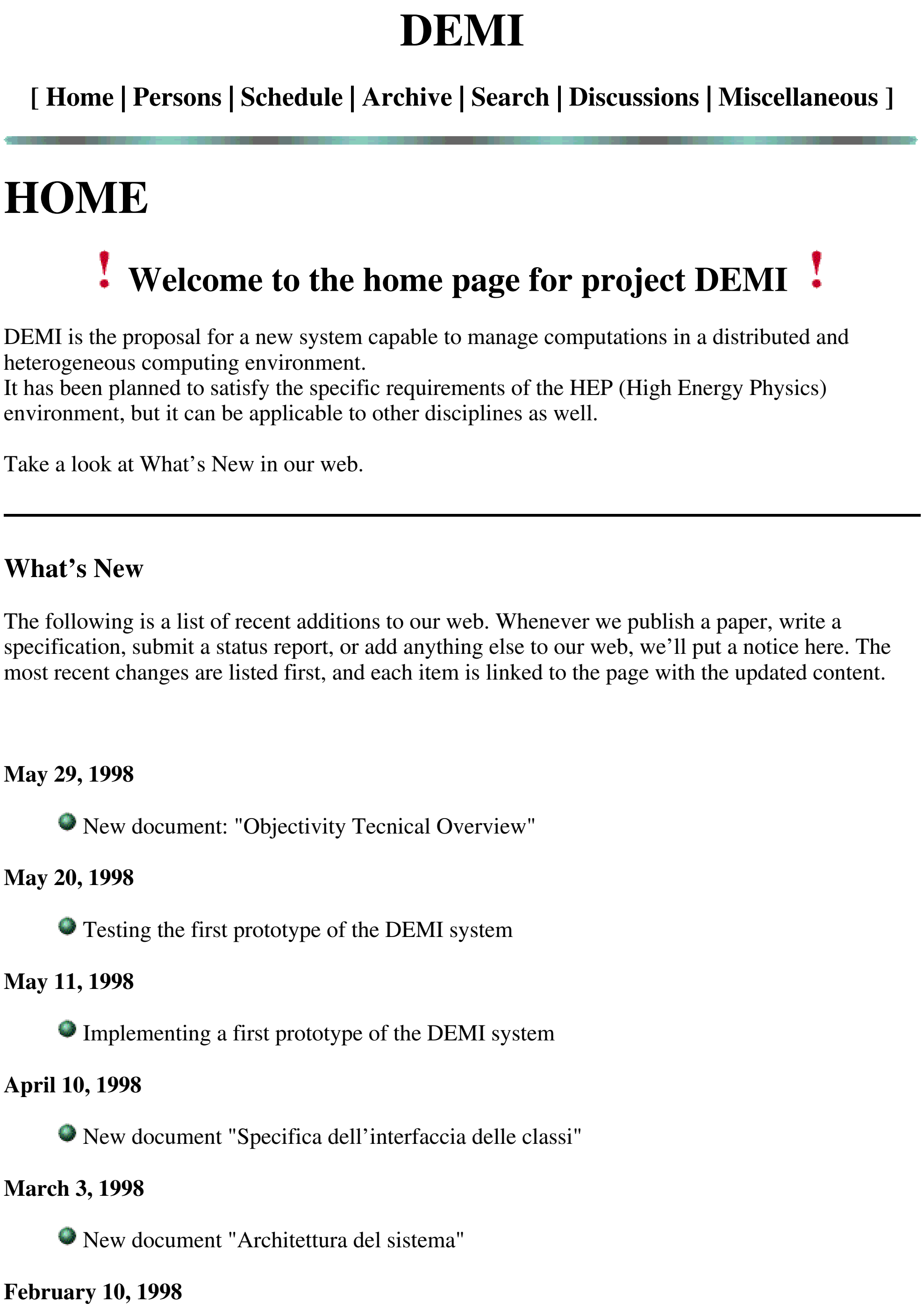
3.3.4 Implementazione

Alla progettazione è seguita la realizzazione di un primo prototipo, e la conseguente verifica della sua funzionalità.

Questa fase dello sviluppo del sistema DEMI è descritta nel capitolo 8.

3.4 Gestione della documentazione

I vari documenti che sono stati prodotti durante lo sviluppo del sistema DEMI sono stati organizzati in un server web [1], in cui sono state inserite anche altre informazioni: discussioni relative alla realizzazione del progetto, link ad altri progetti correlati, bibliografia, ecc... Nelle figure 3.3 e 3.4 sono presentate alcune di queste pagine ipertestuali.



DEMI

[Home | Persons | Schedule | Archive | Search | Discussions | Miscellaneous]

HOME

! Welcome to the home page for project DEMI !

DEMI is the proposal for a new system capable to manage computations in a distributed and heterogeneous computing environment.
It has been planned to satisfy the specific requirements of the HEP (High Energy Physics) environment, but it can be applicable to other disciplines as well.

Take a look at What's New in our web.

What's New

The following is a list of recent additions to our web. Whenever we publish a paper, write a specification, submit a status report, or add anything else to our web, we'll put a notice here. The most recent changes are listed first, and each item is linked to the page with the updated content.

May 29, 1998

- New document: "Objectivity Tecnical Overview"

May 20, 1998

- Testing the first prototype of the DEMI system

May 11, 1998

- Implementing a first prototype of the DEMI system

April 10, 1998

- New document "Specifica dell'interfaccia delle classi"

March 3, 1998

- New document "Architettura del sistema"

February 10, 1998

Figura 3.3: Una pagina del server WWW

DEMI

[Home | Persons | Schedule | Archive | Search | Discussions | Miscellaneous]

ARCHIVE

This page contains links to documents authored by project members, to software programs and components developed by the project, and to other tools we've found useful and collected here for your convenience.

Each entry has a title, a summary, a list of authors, a date of publication or when it was last updated, and links to the material in various formats.

Documents

Introduction to the problem

The initial document that describes the most important features and characteristics required for the DEMI system.

```
By:           Massimo Sgaravatto
Last updated: December 3, 1997
Formats:      .ps (682 k), .gz (138 k)
```

Glossary

Up-to-date glossary that describes the "domain" of the problem. This document is always kept in sync with updates of the other documents.

```
By:           Massimo Sgaravatto
Last updated: November 10, 1997
Formats:      HTML
```

User Requirement Document

The document where all the requirements are specified.

```
By:           Massimo Sgaravatto
Last updated: December 3, 1997
Formats:      .ps (536 K), .gz (111 K)
```

Ambienti visuali per la programmazione parallela

Analisi di alcuni ambienti visuali per la programmazione parallela

```
By:           Massimo Sgaravatto
Last updated: November 24, 1997
Formats:      .ps (44 M)
```

Figura 3.4: Un'altra pagina del server WWW