



Calcolo Distribuito e Coordinazione

Università di Camerino

Corso di Laurea Specialistica in Informatica
12 CFU

I periodo didattico

Emanuela Merelli
emanuela.merelli@unicam.it

Anno accademico 2004-05

a.a. 2004-05



Struttura del Corso

Il corso consiste in

42 ore di lezione
21 di laboratorio

2 ore settimanali di ricevimento studenti

Il corso viene valutato in

12 CFU (Crediti Formativi Universitari)

Collaboratore didattico

Diletta.Cacciagrano@unicam.it

Sono previsti seminari monografici
(su argomenti specifici)



Modalità d'esame

Per sostenere l'esame è necessario

1. Progettare e sviluppare un'applicazione distribuita in un moderno ambiente di calcolo;
2. Tenere un seminario che mostri, tramite i risultati ottenuti nel progetto, le conoscenze e le competenze sul calcolo distribuito e sui modelli di coordinazione.

Per sostenere l'esame è necessario

isciversi alla pagina web: web.unicam.it/matinf



Appelli d'esame

I periodo

Martedì 30 novembre 2004 ore 14:30

II Periodo

Lunedì 4 aprile 2005

III Periodo

lunedì 6 giugno 2005
lunedì 11 luglio 2005

Sessione Recupero

su richiesta



Orario Lezioni

Il corso si terrà dal 11 ottobre al 30 novembre 2004

lezione

**Martedì, mercoledì e giovedì
dalle 15:00 alle 17:00**

laboratorio

**Lunedì dalle 10:00 alle 13:00
dalle 10:00 alle 13:00**



Orario Ricevimento

**Lunedì
dalle 17:00 alle 19:00**

**2° Piano, stanza no.9
*Polo Informatico***



Materiale del corso

Pagina Web

<http://dmi.unicam.it/merelli/calcolo04>



Obiettivi dell'Unità Formativa

- Analizzare le principali caratteristiche dei sistemi distribuiti
- Progettare applicazioni software complesse in moderni ambienti distribuiti
- Utilizzare i concetti fondamentali che caratterizzano i vari modelli di calcolo distribuito e di coordinazione
- Classificare, valutare e applicare alcuni linguaggi per la coordinazione
- Affrontare in modo autonomo la progettazione di applicazioni distribuiti



Sequenza delle diverse fasi dell'attività formativa

- Introduzione al calcolo distribuito
- Modelli di calcolo distribuito e di coordinazione
- Architetture software e coordinazione
- Linguaggi di coordinazione
- Middleware per applicazioni distribuite
- Coordinazione di Agenti software



Testi di Riferimento

TR1: Andrew Tanenbaum,
Distributed Systems,
Prentice Hall, 2002

TR2: Andrea Omicini, F. Zambonelli,
M. Klisch, R. Tolksdorf,
Coordination of Internet Agents
Springer, 2001

TR3: M.L.Liu, *Distributed Computing*,
Person Addison-Wesley, 2004

*Articoli e appunti distribuiti
durante le lezioni –*

*E' in fase di costruzione una
bibliografia annotata*



Programma del Corso

Introduzione al calcolo distribuito

Paradigmi di calcolo (computing)
Paradigmi di comunicazione
Paradigmi per applicazioni distribuite

Modelli e linguaggi di coordinazione

Linda e derivati (KLAIM)

Agenti software e coordinazione

AUML
TROPOS

...

Calcolo distribuito e concorrente

Algebre di processo
Timed PA, Pi-calculus, Ambients
Petri Nets
UML Activity diagrams

...BioAmbients

Middleware per la coordinazione

HERMES
JADE
Lime
MARS
TUCSON



Seminari 2004 -- Prof. Jos Baeten

15:00 – 17:00

AB3

- **Martedì 12 ottobre** *Process Theory*
- **Mercoledì 13 ottobre** *Timed Process Algebras, Part 1*
- **Lunedì 18 ottobre** *Timed Process Algebras, Part 2*
- **Martedì 19 ottobre** *Probabilities in Concurrency*
- **Mercoledì 27 ottobre** *Non Deterministic Regular Expression*



Coordination

Coordination is a key concept for studying the **activities** of complex **dynamic systems**

“Coordination is managing dependencies between activities”
[Malone and Crowston 94]

Such a definition implies that all instances of coordination include **agents** performing **activities** that are **interdependent**

Due to its fundamentality, this notion covers a lot of facets, for instance in distributed artificial intelligence, robotics, **biology**, and organisational sciences

“Coordination is the process of building programs by gluing together active pieces”
[Carriero and Gelernter 92]

Active pieces here can mean processes, objects with threads, agents, or whole applications

programming = coordination + computation



Programming = Coordination + Computation

In programming, coordination is expressed by:

- **Coordination models**

Triple (E, L, M)

- E** – entities being coordinated (agents, processes, tuples)
- L** – media used to coordinate the entities
(channels, shared variables, tuple spaces)
- M** – semantics framework of the model
(guards, synchr. constraints)

- **Coordination languages**

“The Linguistic embodiment of a coordination model”



Programming model

Modello di programmazione completo è costituito da:

1. *Modello di computazione:*

Permette al programmatore di creare attività singole

2. *Modello di coordinazione:*

Permette di creare un'attività unica legando attività separate



Integrazione o separazione?

Integrazione → Linguaggio di programmazione completo:

Linguaggio di calcolo + Linguaggio di coordinazione

Unico linguaggio (Kahn e Miller)

Separazione → Sistema di programmazione completo:

Linguaggio di computazione, Linguaggio di coordinazione

Linguaggi separati (Gelernter e Carriero)



Ensemble :

“Computazione strutturata esplicitamente, vista come un insieme di processi comunicanti”

Esempi: Applicazioni parallele, sistemi distribuiti, sistemi operativi

Ensemble asincrono:

“Collezione di attività asincrone comunicanti”

↓
Programma, processo, thread o agente capace di simulare una Macchina di Turing. Potrebbe essere una persona o un altro ensemble.



Vantaggi della Separazione

Costruire un ensemble equivale a coordinare computazioni separate.

1. *Ortogonalità:* trattare la coordinazione indipendentemente dalla computazione nella fase di costruzione dei programmi.
2. *Generalità:* possibilità di definire linguaggi di coordinazione general-purpose, applicabili a qualsiasi tipo di ensemble.



What is a Coordination Model?

A coordination model is the formal basis (semantics) for a coordination language;

It proved that it is possible to “think in a coordinated way” abstracting from low level mechanisms for concurrent, parallel, or distributed programming

A *coordination model* is an **abstract** (semantic) **framework** useful to study and understand problems in designing concurrent and distributed programs

“A coordination model is the **glue** that binds separate activities into an ensemble” [CarGel92]

Historically, Linda was introduced as a new model for parallel programming, more flexible and high level wrt its competitors (“Linda Is Not aDA”)



What is a Coordination Model? Cont.

In other words, a coordination model provides a minimum **framework** in which the interaction of individual agents can be expressed

Coordination framework covers the issues of **dynamic agent creation and destruction**, **control** of communication flows among agents, control of spatial distribution and **mobility of agents**, as well as **synchronization** and **distribution** of actions over time



What is a coordination language?

A **Coordination language** is the linguistic embodiment of a coordination model, and consists of some coordination mechanisms that are added to an *host* (sequential) language

In practice, a coordination language includes clearly defined mechanisms for **communication**, **synchronization**, **distribution**, and **concurrency** control.



References

[Malone and Crowston '94]

T. Malone and K. Crowstone, *The interdisciplinary study of coordination*, ACM Computing Survey, vol.26, no.1, pp.87-119, 1994

[Gelernter and Carriero '92]

N. Carriero and D. Gelernter, *Coordination languages and their significance*, Communication of ACM, vol.35, no.2, 1992

[G.Cabri et al. 2000]

G. Cabri, L. Leonardi and F. Zambonelli, *Mobile-Agent coordination models for Internet Applications*, IEEE Computer, vol.33, no.2, pp.82-89, feb. 2000



Digital Library

- ACM
 - www.acm.org
 - User-id: emerelli
 - Pwd: costanza
- IEEE
 - www.computer.org
 - User-id: miaweb
 - Pwd: memacu04



Internet and Coordination

The Internet can be enhanced by **coordination middleware** to a programmable platform offering support for

- large-scale groupware,
- agent-based applications,
- high-performance computational services (eg. the GRID), and
- activity-based applications



A case study

- Distributed Applications in Bioinformatics Domain
- Activity-based applications
- Tool integration distributed middleware
- Agent-based coordination model



The Tool Integration Problem in Activity-Based Applications

Problem: To *integrate and coordinate multiple software tools* for retrieving and integrating heterogeneous, distributed and frequently redundant data

Objective: To *integrate and coordinate several software tools* in order to provide a uniform way and an high level of abstraction for users

Aim: To define an *integrated environment* freeing the user from the need to know details on data repository and to coordinate the intermediate steps of an experiment (tasks)

Proposed Approach: To define an application as a workflow of tasks; to coordinate the execution of cooperative tasks by using software agent tools



Activity-Based Application in Bioinformatics Domain

Problem: Given genetic sequence, e.g., *atggag ... tga*, find the crystallographic structure of the 10 proteins more similar to the protein corresponding *atggag ... tga*

Objective: To use several Bioinformatics Software Tools available on Internet in order to find the wanted result

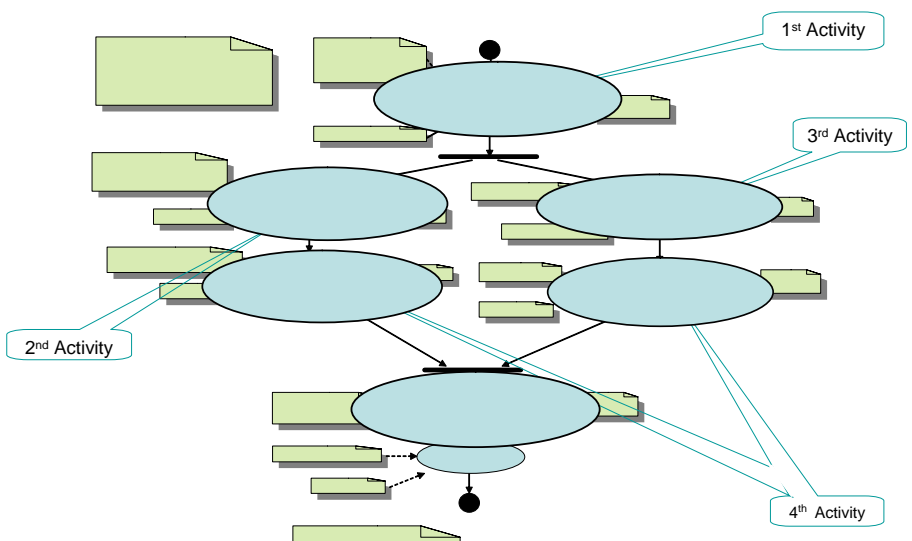
1. Select the 10 proteins more similar to the given sequence *atggag ... tga*
 - by using **BLASTn** in **GenBank** at **NCBI** 1st Activity
2. Search for the **PDB ID** (crystallographic structure identifier) of each selected proteins,
 - by using **BLASTp** in **SWISS-PROT** at **EMBL** **EBI** 2nd Activity
 - by retrieving from **PubMed** via **Entrez Retrieval System** at **NCBI**, abstracts containing PDB-ID information
3. Search for the **Crystallographic Structure** of any selected **PDB ID**
 - find 3-D biological macromolecular structure in **Protein DataBank** repository 3rd Activity

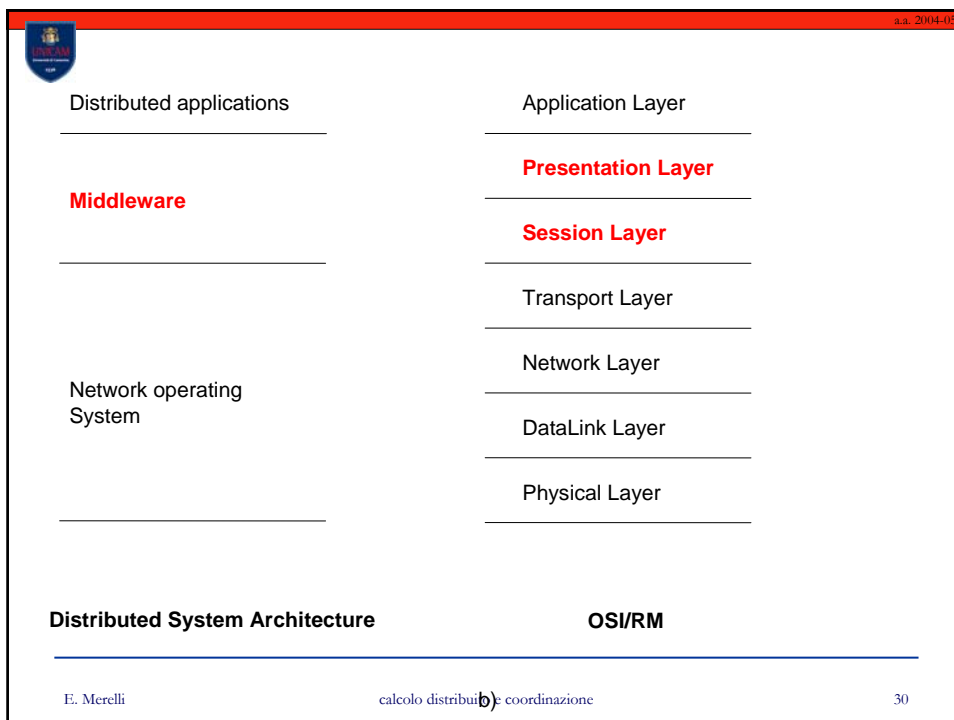
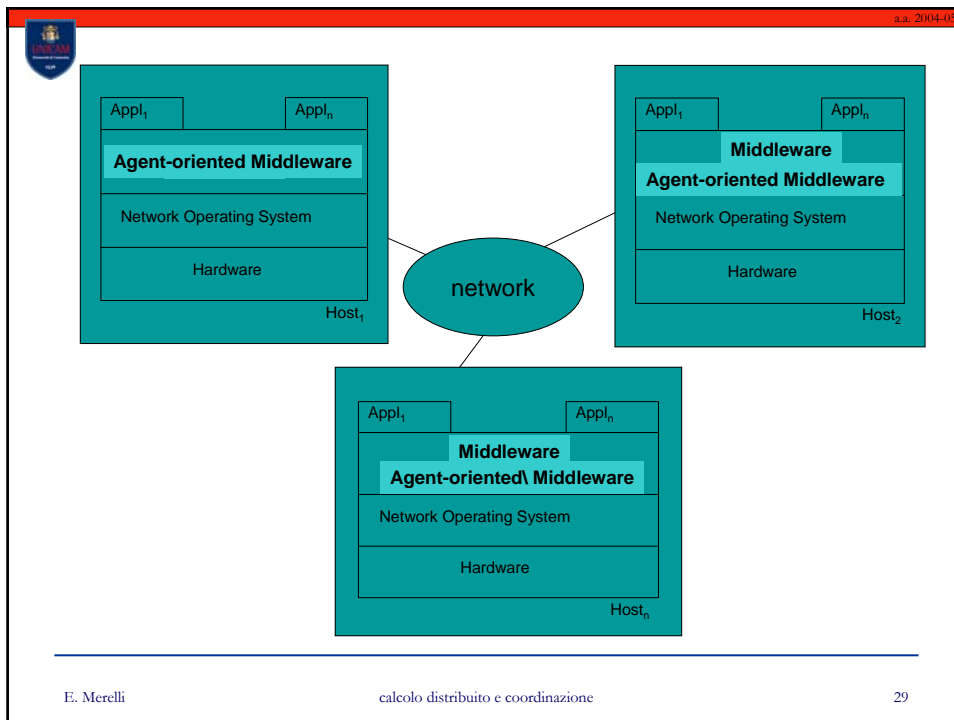
Aim: the automatic execution of the activities relies on the **coordination of several Bioinformatics tools**, freeing the Bioscientist from the need to know details on data and to continuous interact with remote sites.

By defining an **integrated environment** supporting the coordination of all activities in order to automatically execute an experiment



Coordination of a workflow of activities







Coordination of software components

Coordination is essential to control autonomy

Consider three types of autonomy:

- Design autonomy, reflects the independence of designers and vendors to build their software components.

Design autonomy forces coordination about schemas and ontologies

- Configuration autonomy, reflects the independence of system administrators to set up individual hosts and networks.

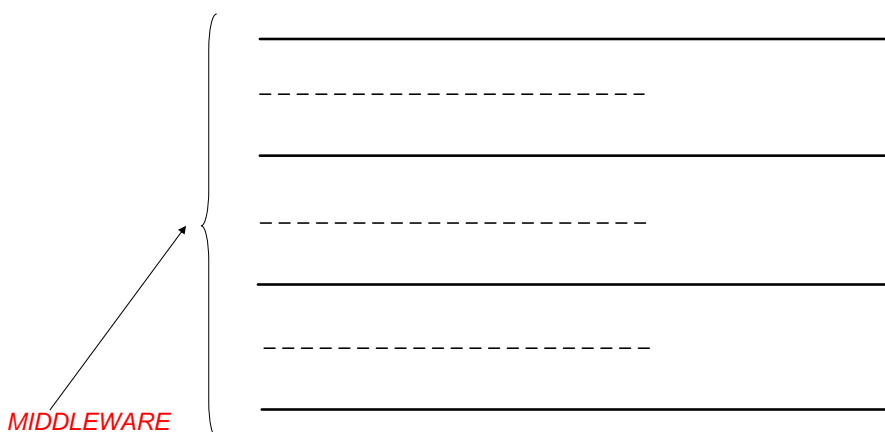
Configuration autonomy forces coordination for resources discovery

- Execution autonomy, reflects the independence of end users to act as they prefer.

Execution autonomy forces coordination through interaction protocols

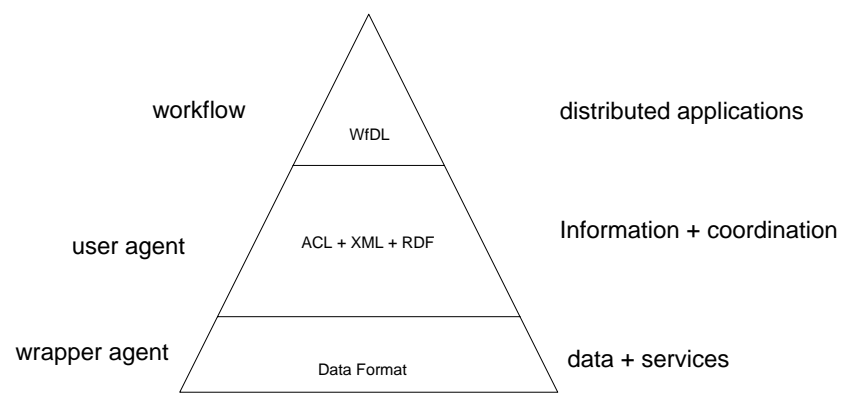


Software Architecture

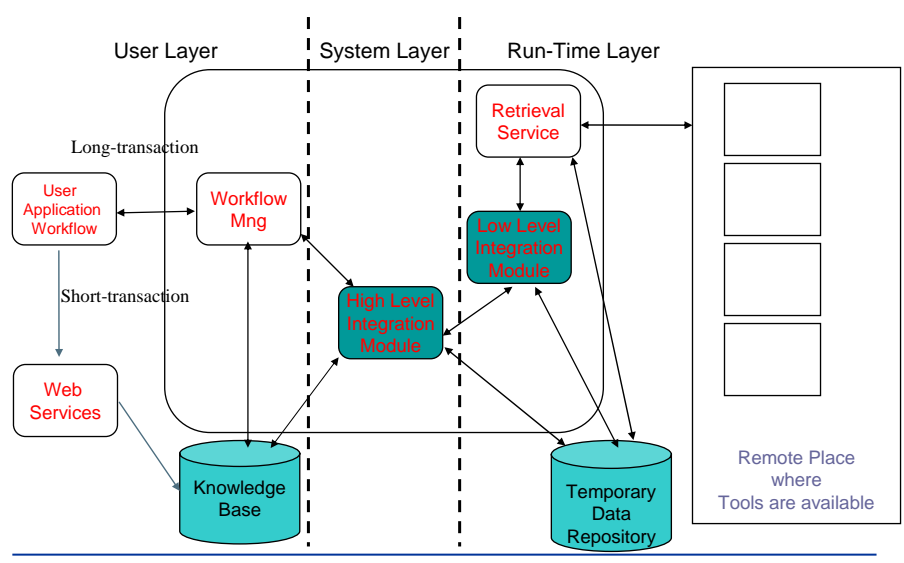


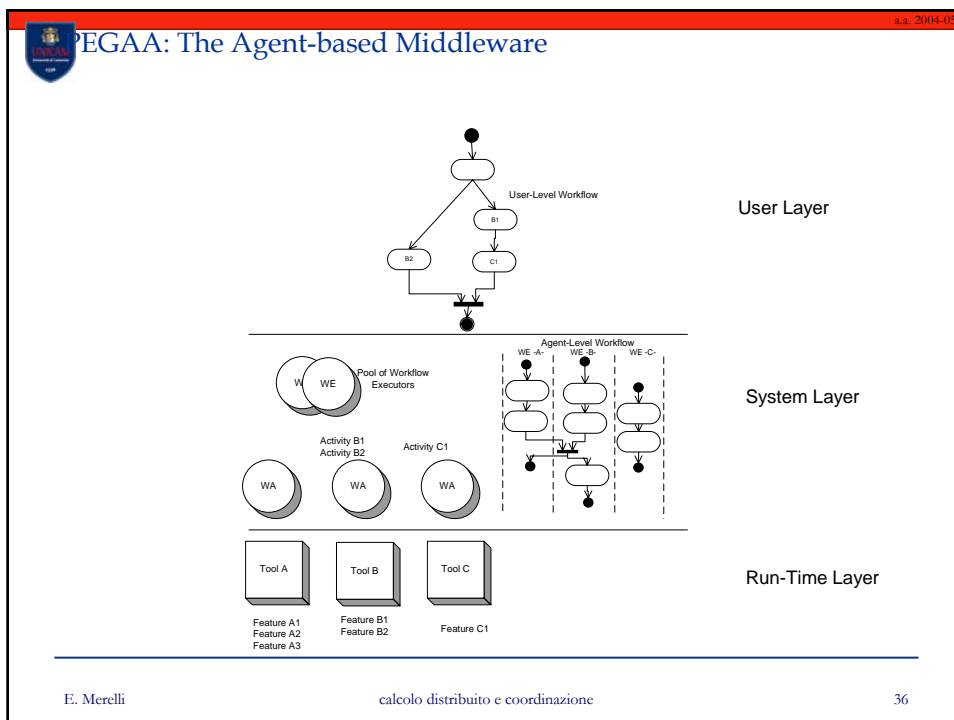
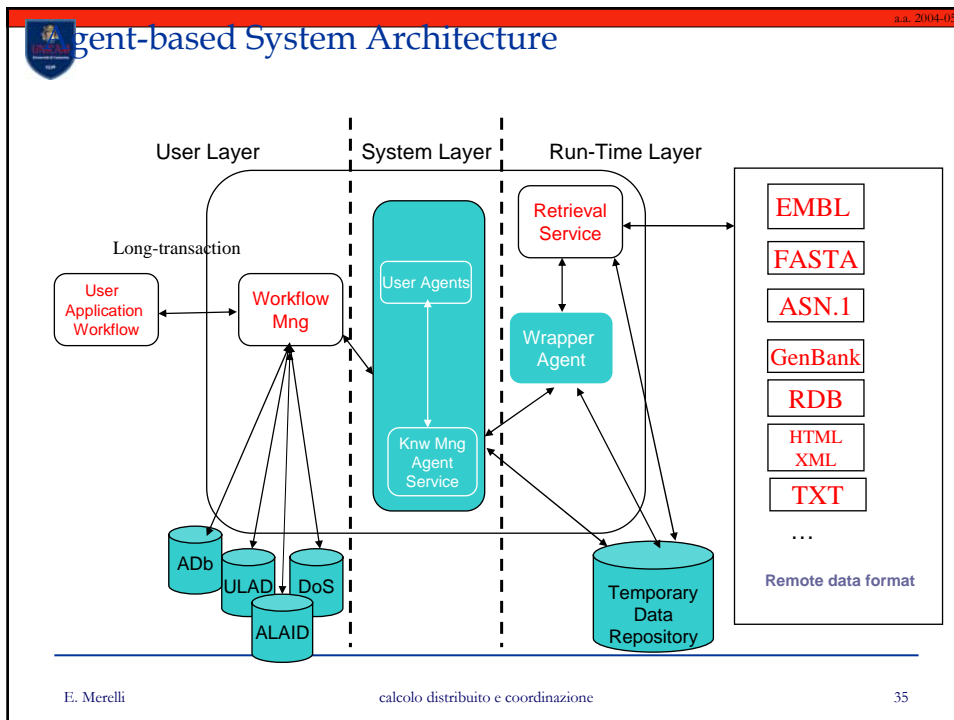


From Data to Knowledge and vice versa



A general system's architecture

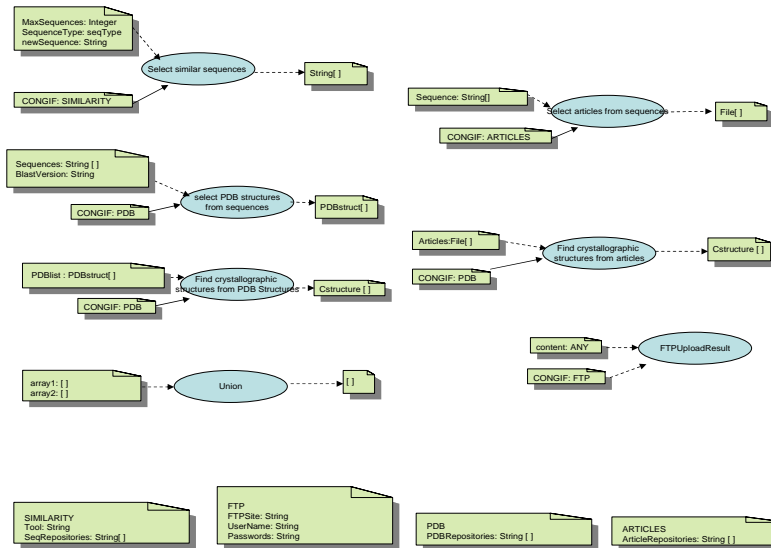






Activity Database (ADb)

a.a. 2004-05



E. Merelli

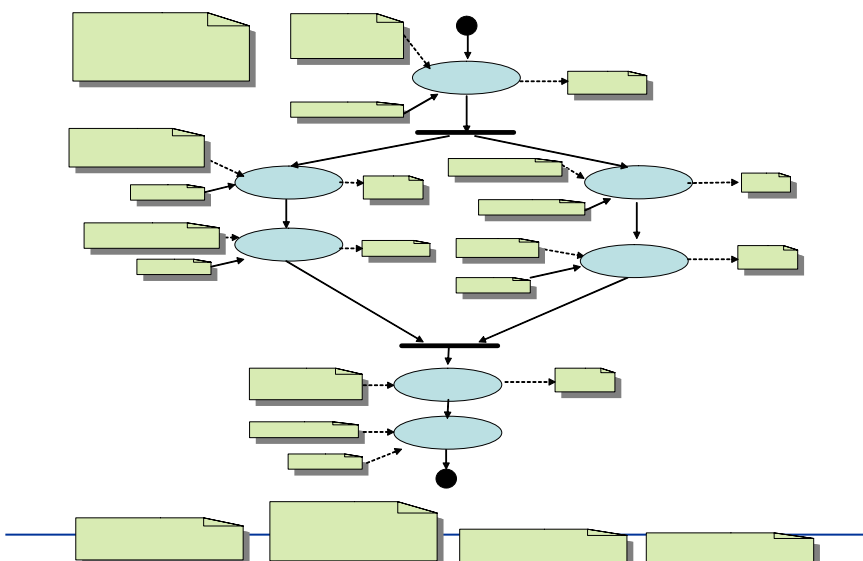
calcolo distribuito e coordinazione

37



User Level Workflow

a.a. 2004-05

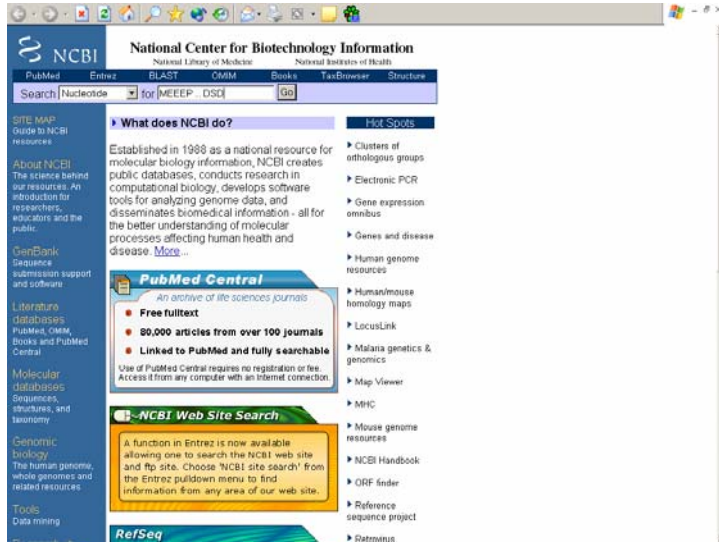


E. Merelli

calcolo distribuito e coordinazione

38

NCBI - Home page

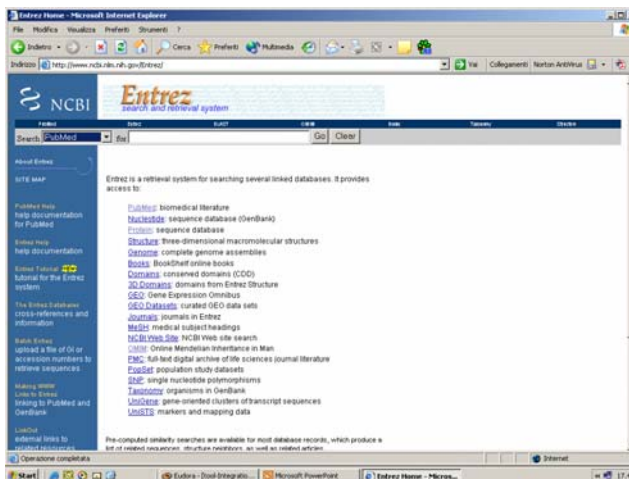


E. Merelli

calcolo distribuito e coordinazione

41

NCBI - Entrez



E. Merelli

calcolo distribuito e coordinazione

42



PDB - Home page

E. Merelli

calcolo distribuito e coordinazione

43



NCBI - formats

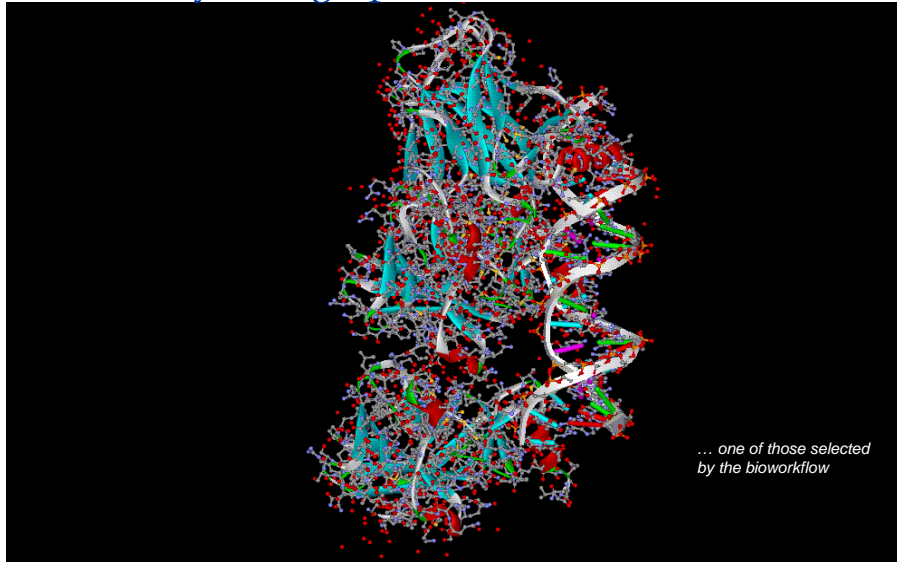
E. Merelli

calcolo distribuito e coordinazione

44



P53 Crystallographic Structure



Assegnazione progetti

max 2 studenti per gruppo

- 1) Dominio biologico (system biology):
 - 1) modello di coordinazione dei componenti di una cellula
 - 2) Simulazione del processo di invasione del parassita della malaria
- 2) Dominio industriale: modello di coordinazione dei componenti di un sistema di distribuzione
- 3) Dominio biomedico: modello di coordinazione dei componenti di un workflow



- 25 ottobre: marco vita "AOSE e coordinazi
- 18 ottobre: chiara ercoli "middleware e coordinazione"

- 8 novembre