

Nuovi paradigmi di sviluppo del software: la progettazione e programmazione agent-oriented

Massimo Cossentino, Carmelo Lodato, Salvatore Lopes,
Patrizia Ribino, Luca Sabatucci

Consiglio Nazionale delle Ricerche (CNR)
Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
Palermo (Italy)

[cossentino, lodato, lopes, ribino, sabatucci]@pa.icar.cnr.it

Contenuti del seminario

- Parte 1: Ingegneria del Software Orientata agli Agenti
 - M. Cossentino
- Parte 2: Jason, un framework per lo sviluppo di agenti BDI
 - S. Lopes
- Parte 3: Progettare e sviluppare applicazioni Jason, esempi
 - P. Ribino
- Parte 4: La progettazione di un sistema di workflow per una realtà aziendale di gestione documentale
 - L. Sabatucci
- Parte 5: Proposte di tesi e tirocinio
 - M. Cossentino

In breve, di cosa parleremo?

- Dell'attività di ricerca svolta presso il gruppo AOSE (*Agent-Oriented Software Engineering*) dell'ICAR-CNR di Palermo
- Partecipano a questa attività:
 - M. Cossentino (CNR),
 - C. Lodato (CNR),
 - S. Lopes (CNR),
 - P. Ribino (CNR),
 - L. Sabatucci (CNR),
 - V. Seidita (Unipa)

Argomento di ricerca

- L'attività di ricerca che verrà presentata rientra nel settore della *Ingegneria del Software orientata agli agenti*
- Cos'è l'ingegneria del software?
 - Software Engineering is the “multi-person construction of multi-version software”
(D.L. Parnas. *Some Software Engineering Principles. In Structured Analysis and Design, State of the Art Report. Infotech International, 1978*)
- Che vuol dire “orientata agli agenti”?
 - Per distinguerla da quella orientata agli oggetti (o classica)
 - Il concetto di agente verrà definito nel seguito

PARTE 1

Ingegneria del Software Orientata agli Agenti

An Introduction to Agents

Source

- Many slides come from the following book:
 - “An Introduction to MultiAgent Systems”
by Michael Wooldridge, John Wiley & Sons, 2002.
<http://www.csc.liv.ac.uk/~mjw/pubs/imas/>

Programming progression...

- Programming has progressed through:
 - machine code;
 - assembly language;
 - machine-independent programming languages;
 - sub-routines;
 - procedures & functions;
 - abstract data types;
 - objects;

to *agents*.

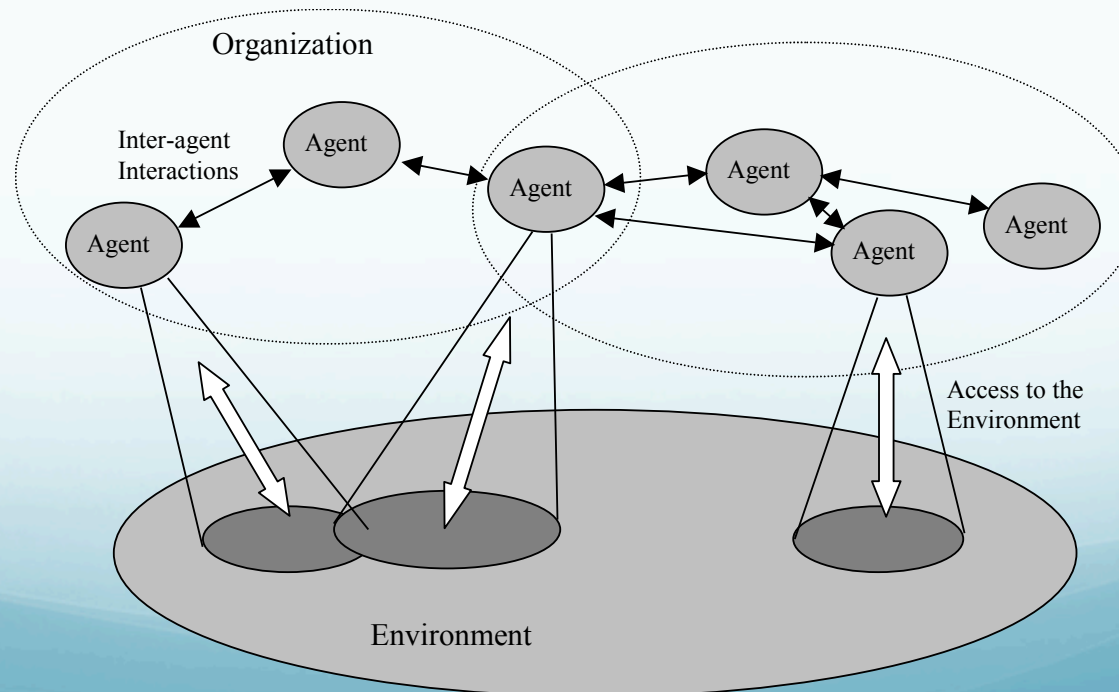
What is an Agent?

- The main point about agents is they are *autonomous*: capable of acting independently, exhibiting control over their internal state
- Thus: *an agent is a computer system capable of autonomous action in some environment in order to meet its design objectives*



Multiagent Systems, a Definition

- A multiagent system (MAS) is one that consists of a number of agents, which *interact* with one-another
- Usually agents in a MAS adopt social structures like groups, societies or organizations



What is an Intelligent Agent?

- *An intelligent agent is a computer system capable of **flexible** autonomous action in some environment*
- By *flexible*, we mean:
 - *reactive*
 - *pro-active*
 - *social*

Reactivity

- In the real world things change, information is incomplete. Many (most?) interesting environments are *dynamic*
- A **reactive** system is one that maintains an ongoing interaction with its environment, and **responds to changes that occur** in it (in time for the response to be useful)

Proactiveness

- Reacting to an environment is easy (e.g., stimulus → response rules)
- But we generally want agents to *do things for us*
- Hence *goal directed behavior*
- *Pro-activeness* = generating and attempting to achieve goals; not driven solely by events; taking the initiative
- Recognizing opportunities
- Anticipating changes

Social Ability

- The real world is a *multi*-agent environment: we cannot go around attempting to achieve goals without taking others into account
- Some goals can only be achieved with the cooperation of others
- *Social ability* in agents is the ability to interact with other agents (and possibly humans) via some kind of *agent-communication language*, and perhaps cooperate with others

Objects do it for free...

- *agents do it because they want to*
- *agents do it for “money”*

BDI Agents

What is a BDI agent?

- A BDI agent is a particular type of rational agent
 - A rational agent is an intelligent agent that directs its activity towards achieving goals
- A BDI agent has specific mental attitudes: Beliefs, Desires and Intentions (BDI).

BDI = Belief, Desire, Intention

- **Beliefs** represent the informational state of the agent, in other words its beliefs about the world (including itself and other agents).
- **Desires** represent the motivational state of the agent. They represent objectives or situations that the agent would like to accomplish or bring about.
 - Examples of desires might be: find the best price, go to the party or become rich.
- **Goals:** A goal is a desire that has been adopted for active pursuit by the agent.
 - Usage of the term goals adds the further restriction that the set of active desires must be consistent.
 - For example, one should not have concurrent goals to go to a party and to stay at home – even though they could both be desirable.

BDI = Belief, Desire, Intention /2

- **Intentions** represent the deliberative state of the agent – what the agent has chosen to do. *Intentions are desires to which the agent has committed.*
- **Plans:** Plans are sequences of actions that an agent can perform to achieve one or more of its intentions.
 - Plans may include other plans: my plan to go for a drive may include a plan to find my car keys.
- **Events:** These are triggers for reactive activity by the agent. An event may update beliefs, trigger plans or modify goals.
 - Events may be generated externally and received by sensors or integrated systems.
 - Additionally, events may be generated internally to trigger decoupled updates or plans of activity.

Research on AOSE (at ICAR-PA)

- Design processes/Methodologies
 - Design tools
- Metamodeling and model driven engineering techniques
- Modeling languages
- Adaptive workflows
- Simulations
- Cognitive Robotics

SMART INFORMATION SYSTEMS

END of Part 1

Next: JASON (a framework for developing BDI agents)

Parte 5

Ricerca, Tirocini e Tesi di laurea

Possibili Tesi di Laurea/ Attività di Tirocinio

Lo studente/laureato in Informatica vs il CNR

- Lo studente presso il CNR può:
 - Svolgere il suo tirocinio
 - Svolgere tesi di laurea
- Il laureato in (Ingegneria) Informatica può collaborare con il CNR mediante:
 - Contratti di lavoro co.co.co.
- Il laureato può iniziare la carriera di ricerca presso il CNR mediante:
 - Assegni di Ricerca
 - Punto di arrivo:
 - Posizione di ricercatore CNR (difficilissimo!!!!)

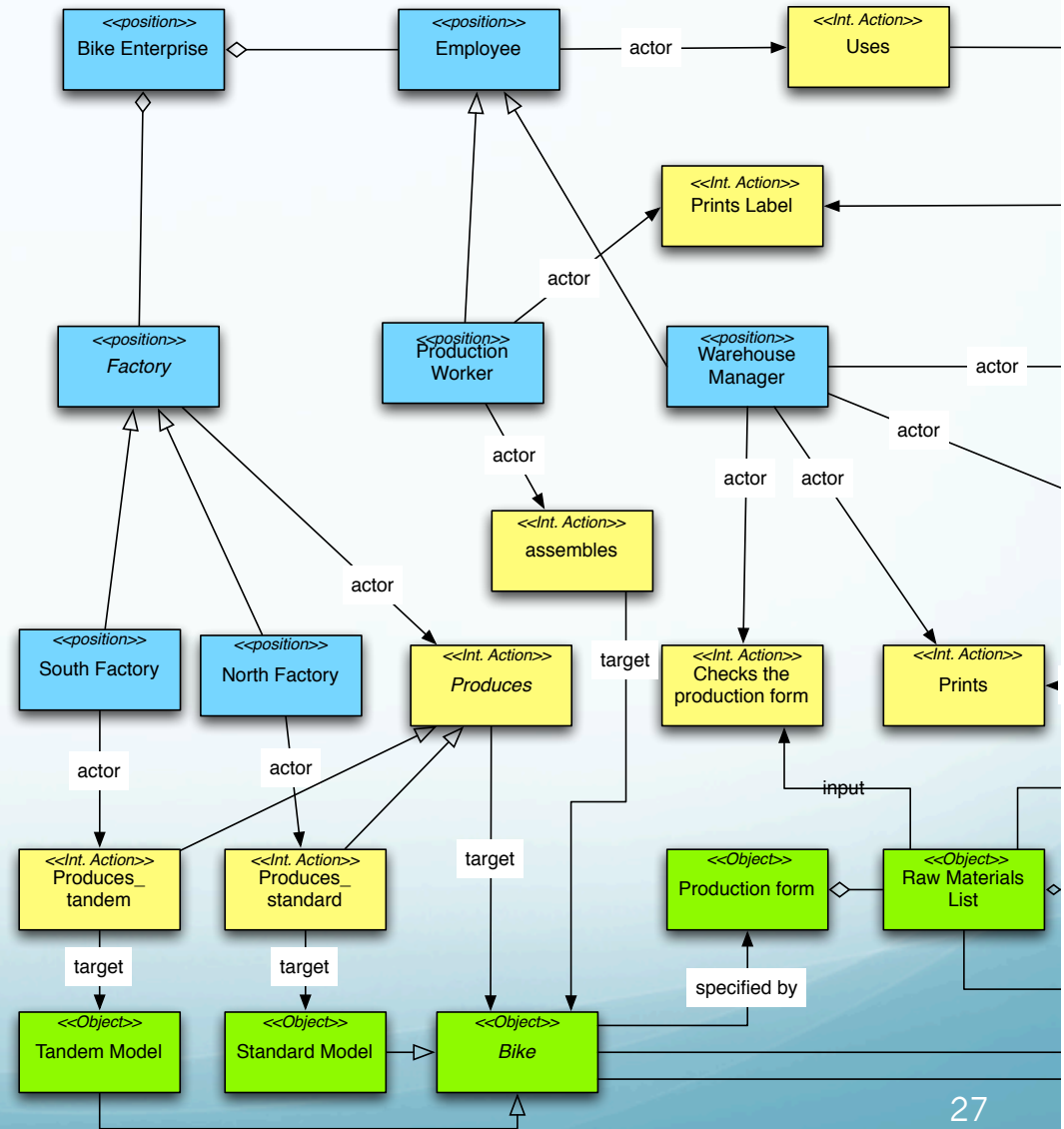
Tesi/tirocini di Ingegneria del Software orientata agli agenti

- Fare una tesi di laurea (o attività di tirocinio) presso un Ente Pubblico di Ricerca come il CNR vuol dire:
 - Studiare problematiche attualissime
 - Imparare come utilizzare tecnologie di frontiera
- In genere tirocini e tesi di laurea legati alla ricerca sono:
 - Impegnativi
 - Premianti per l'esame di laurea
 - Qualificanti per l'inserimento nel mondo del lavoro

Identificazione di goal da ontologia

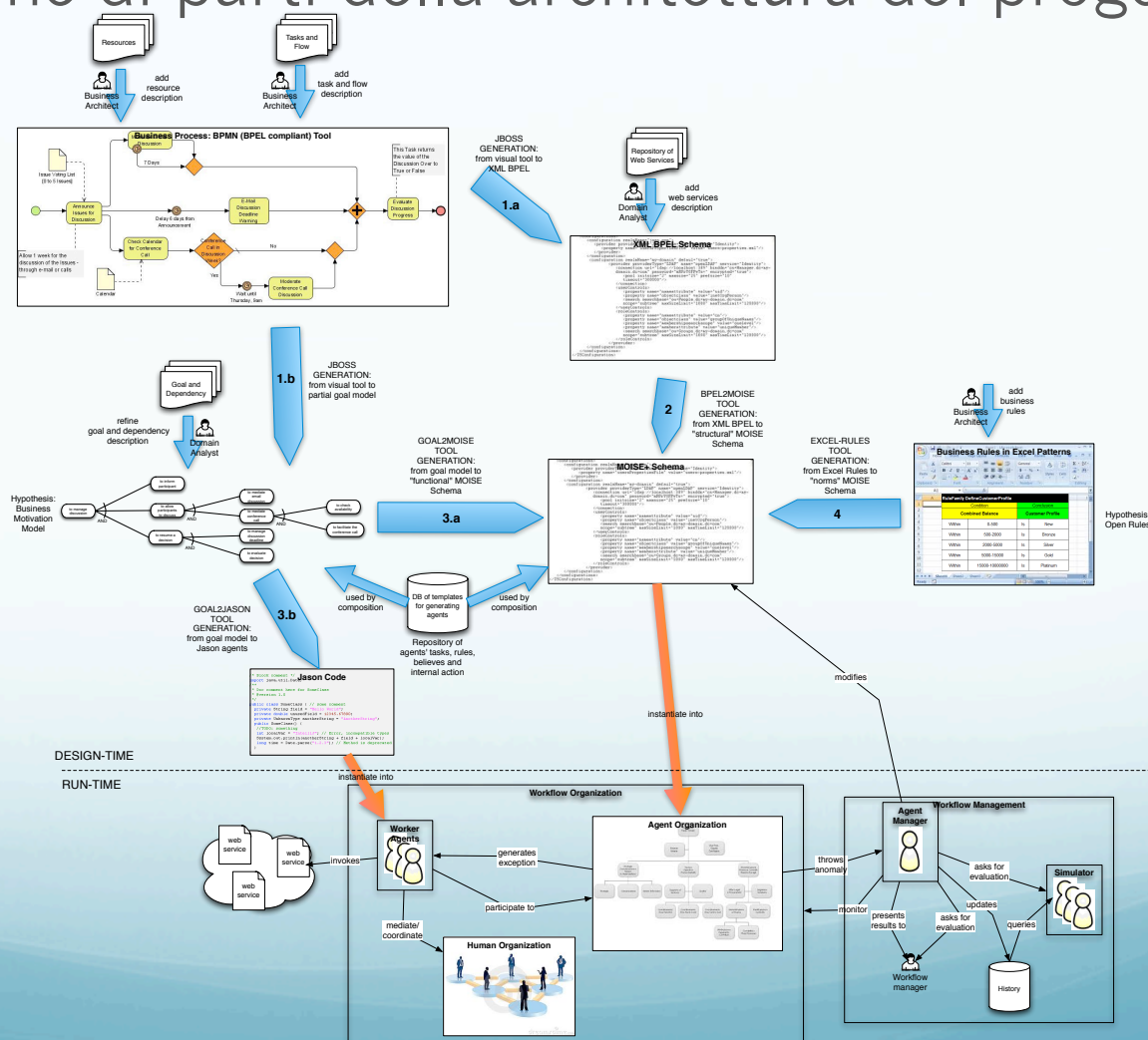
- Generazione codice ontologia da diagramma Metaedit
- Estrazione automatica di goal da ontologia RDF

Goal Information		
NAME: [to Produce Bike]	STATE: [Bike is produced]	WHO: Factory
DEPENDENCIES: [to Produce Bike] depends on [to Assemble Bike], [to Produce Tandem Model] and [to Produce Standard Model]		
NAME: [to Produce Tandem Model]	STATE: [Tandem Model of Bike is produced]	WHO: South Factory
DEPENDENCIES:		
NAME: [to Produce Standard Model]	STATE: [Standard Model of Bike is produced]	WHO: North Factory
DEPENDENCIES:		
NAME: [to Assemble Bike]	STATE: [Bike is assembled]	WHO: Production Worker
DEPENDENCIES:		
NAME: [to Set in Print]	STATE: [to be setting in print]	WHO: Production Worker
DEPENDENCIES: [to Set in Print] depends on [to Create Label]		
NAME: [to Send To Raw Material Supplier]	STATE: [sending To Raw Material Supplier]	WHO: Warehouse Manager
DEPENDENCIES:		
NAME: [to Plan Raw Material Supply]	STATE: [Raw Material Supply is planned]	WHO: Warehouse Manager
DEPENDENCIES:		
NAME: [to Record]	STATE: [to be recording]	WHO: Software System
DEPENDENCIES: [to Set in Print] depends on [to Create Label]		
NAME: [to Create Label]	STATE: [Label is created]	WHO: Software System
DEPENDENCIES: [to Create Label] is depended on by [to Set in Print]		



Sistemi di Gestione Documentale

- Realizzazione di parti della architettura del progetto POR IDS



Altre tesi

- SMartcity – interfacciamento con Google Maps, simulazione traffico cittadino
- Design Pattern per Jason
- Valutazione delle prestazioni di sistemi Jason+Moise
- Animazione diagrammi Metaedit, simulazione dei diagrammi fatti come strumento di debugging

FINE

Grazie per l'attenzione

[cossentino,lodato,lopes,ribino,sabatucci]@pa.icar.cnr.it

Sviluppare un software è come costruire un palazzo?

- Il software è un'opera dell'ingegno, come gli edifici e le altre opere dell'ingegneria civile
- Il software è capillarmente diffuso, è un elemento costante della nostra esperienza quotidiana, come gli edifici in cui abitiamo, lavoriamo, svolgiamo altre attività
- Eppure...

**Se facessimo gli edifici come
facciamo il software saremmo
estinti sotto le macerie!**

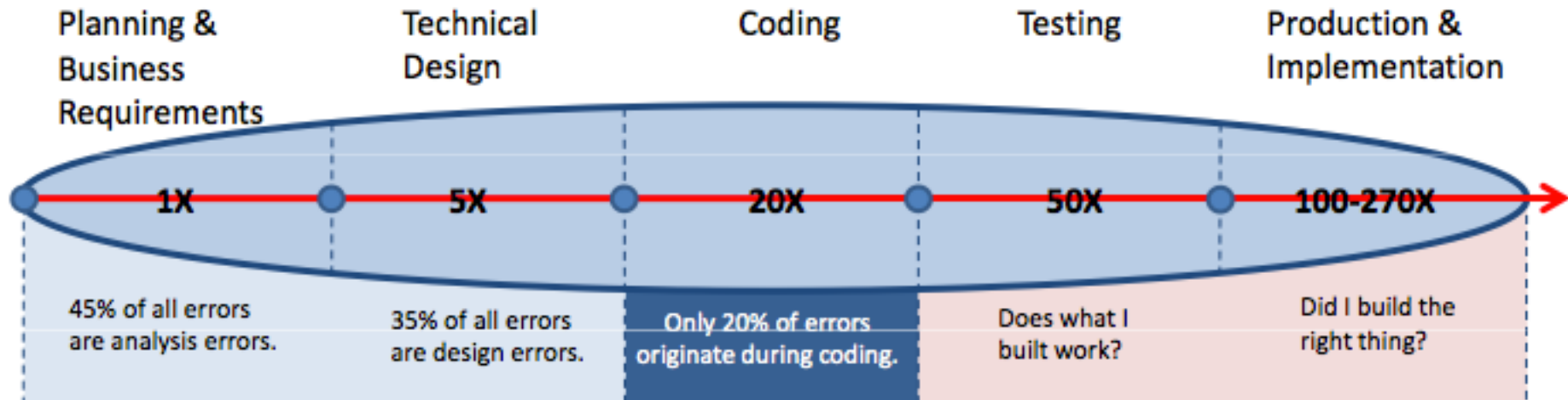
Sviluppare un software è come costruire un palazzo? (2)

COSTRUZIONI CIVILI	SOFTWARE
I risultati sono facilmente misurabili e osservabili	Le metriche del software danno indicazioni di difficile interpretazione e non sono facilmente ottenibili
Servono molte maestranze a bassa professionalità	La maggior parte del personale coinvolto è laureato
Per diminuire i tempi spesso basta aumentare il personale e/o i macchinari impiegati	L'aumento di personale non ha un impatto diretto sui tempi (soprattutto per prodotti medio-piccoli). Le attrezzature sono quasi influenti.

Quanto costa non progettare il software?

- La risoluzione dei problemi costa tanto di più quanto più tardi essi vengono scoperti nel ciclo di sviluppo

QUALITY AND THE LIFE CYCLE

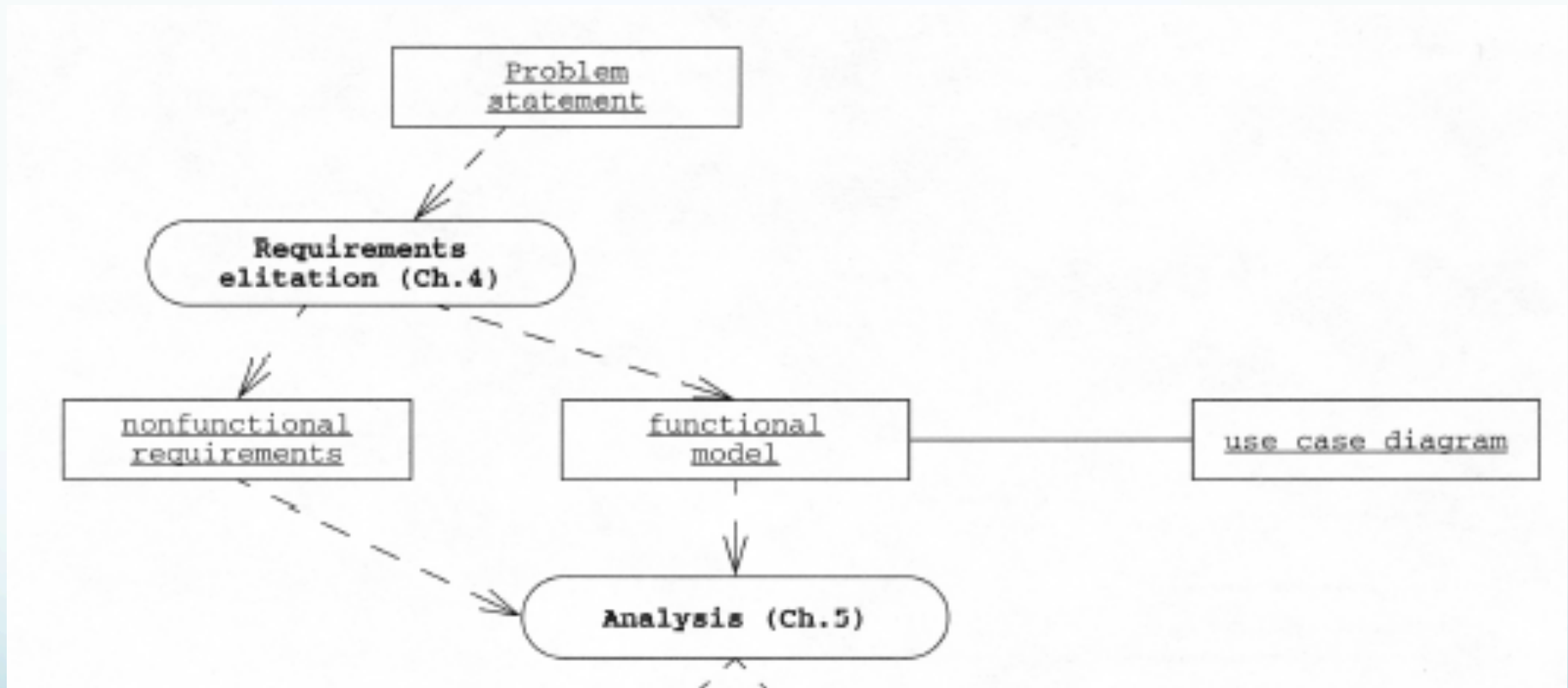


Source: Boehm, B., Software Economics, Prentice Hall

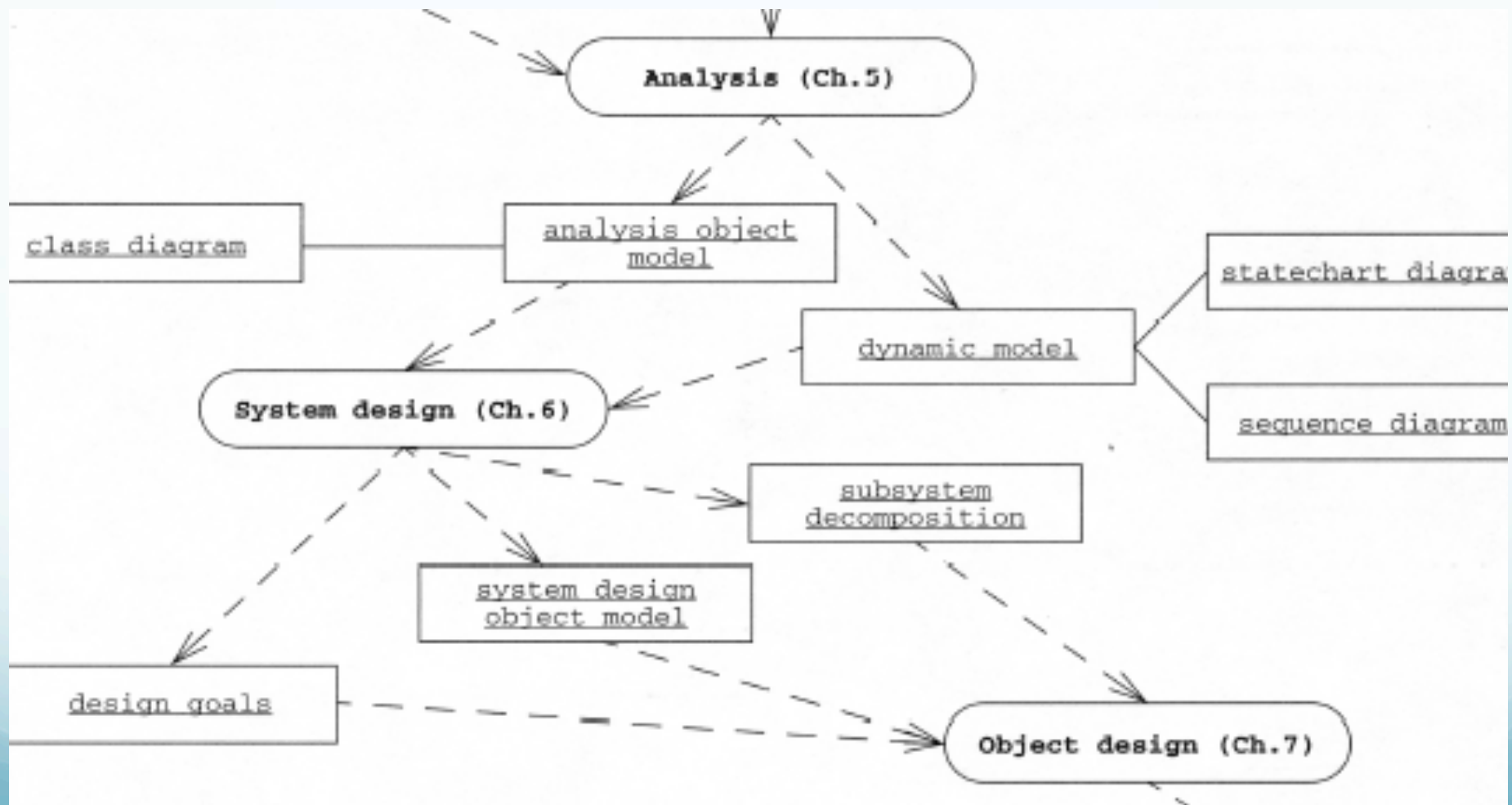
Il Corso di Ingegneria del Software

(in 3 slide)

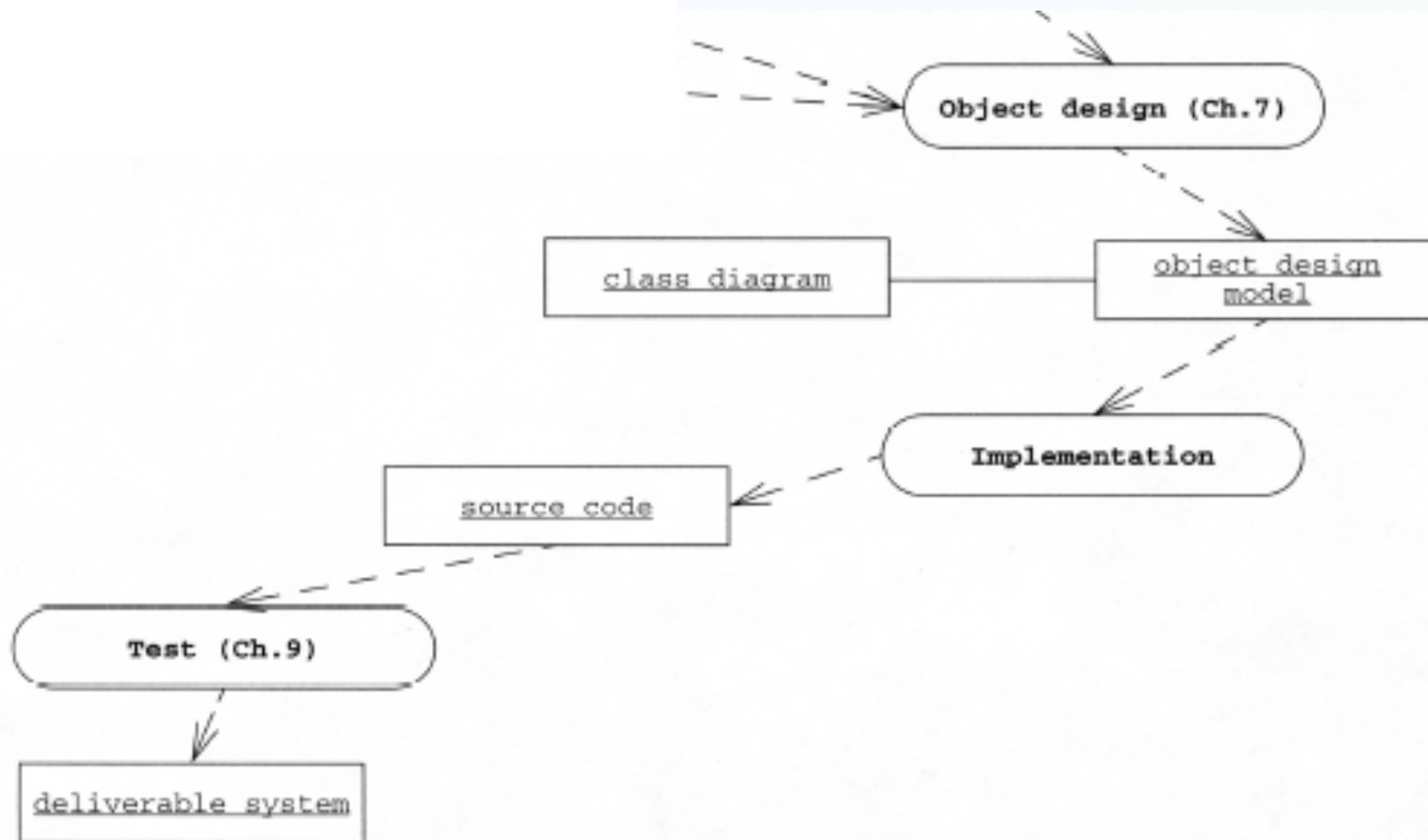
Il processo di progettazione 1/3



Il processo di progettazione 2/3



Il processo di progettazione 3/3



Tematiche di ricerca

- Metodologie per lo sviluppo del software
 - Definizione di Processi di progettazione
 - Studio delle tecniche di composizione dei processi
 - Tool per la progettazione del software
 - Design Pattern
- Sistemi auto-organizzanti o adattativi
- Tecniche di progettazione applicate alla Robotica Cognitiva

Simulatore di navigazione indoor

- Per esplorare approcci innovativi alla self-adaptation e self-organization
- AGV (Automated Guided Vehicles) guadagnano quando viaggiano carichi
- Alcune casse necessitano di due AGV
- Quando un AGV si scarica abbandona il carico
- Si può spostare o caricare una cassa abbandonata
- Quando ad un AGV conviene collaborare e quando gli conviene fare un viaggio in autonomia?

