Structural Biology Metaphors Applied to the Design of a Distributed Object System

Ladislau Bölöni, Ruibing Hao, Kyunkoo Jun, and Dan C. Marinescu

Department of Computer Sciences

Purdue University

Acknowledgments

- The work reported in this paper was partially supported by a grant from the National Science Foundation, MCB-9527131, by the Scalable I/O Initiative, and by a grant from the Intel Corporation

CS concepts/algorithms inspired by biological models

- Neural Networks (alternative to von Neumann architecture) <= neurology.
- Genetic Algorithms (used to solve optimization problems) <= genetics.
- Mutation Analysis (used in Software Engineering for testing and verification) <= genetics.

Biological metaphors for building complex systems

- Protein structure. Nature uses composition to build complex systems.
- Genetics. Cells carry genetic information.
- Immunology. An antibody recognizes the binding site of a virus.
- Signalling.

Protein structure

- 20 aminoacides, the basic building blocks of life.
- Primary structure. The aminoacid sequence of the peptide chain of a protein.
- Secondary structure. Alpha helices, beta strands.

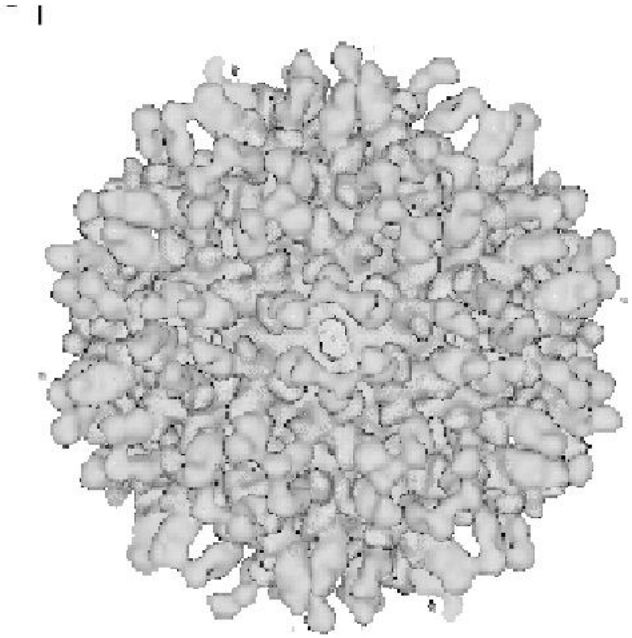
Protein Structure (cont'd)

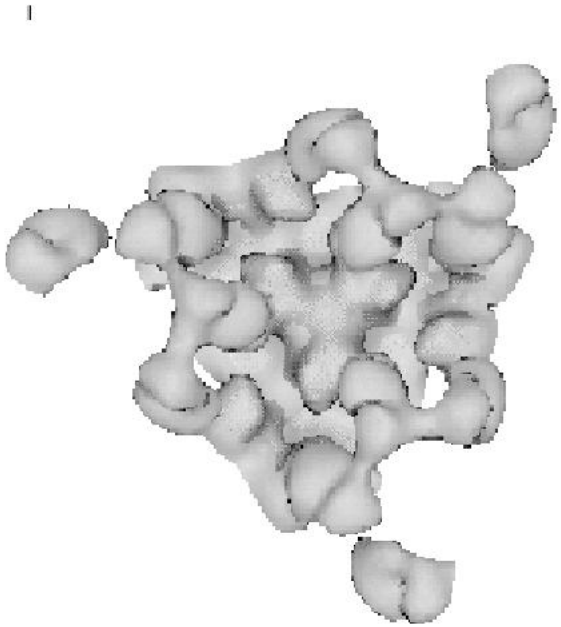
- Tertiary structure. Formed by packing alpha helices and beta strands into globular units called domains.
- Quaternary structure. Several polypeptide chains.

Active Site

- By formation of tertiary and quaternary structures proteins far apart in the sequence are brought together in three dimension to form a functional region, an active site.
- Example: the Ross-River virus with antibody fragments.

Active site





Principles

- The three dimensional structure of a protein determines its function.
- The disposition in space and the type of atoms provide a lock that can be recognized by other proteins that may bind to it provided that they have a key.
- Genetic economy.

Mapping the metaphors

- Introspection and reflections for composition.
- Metaobjects and agents to emulate lock and key mechanisms.
- Metobjects and genetic information.
- Agent signalling. Sets of rules and facts exchanged by agents with inference capabilities.

Lock & Key mechanism for building complex systems

- Network Objects
  - Hardware platforms, communication links, programs, data, libraries, services
- Metaobjects
  - Provide knowledge about network objects

Metaobjects

- Collections of the relevant properties of a network object
- Extend the notion of "*reflection*" from languages like Java
- Contain only the relatively stable properties
- There is a loose coupling between metaobjects and the corresponding network objects
- Some inconsistency is sometimes unavoidable

The Bond distributed object system

Bond

- Message-oriented middleware
- Every Bond object speaks the KQML agent communication language
- Object-oriented Java implementation
- Implements the lock & key mechanism using metaobjects and software agents
- **`http://bond.cs.purdue.edu`**

Method Invocation vs. Message Passing

- Efficent
- Restrictive
- Proper for performance-critical applications

- Less Efficient
- Less Restrictive
- Proper for agent-based programming

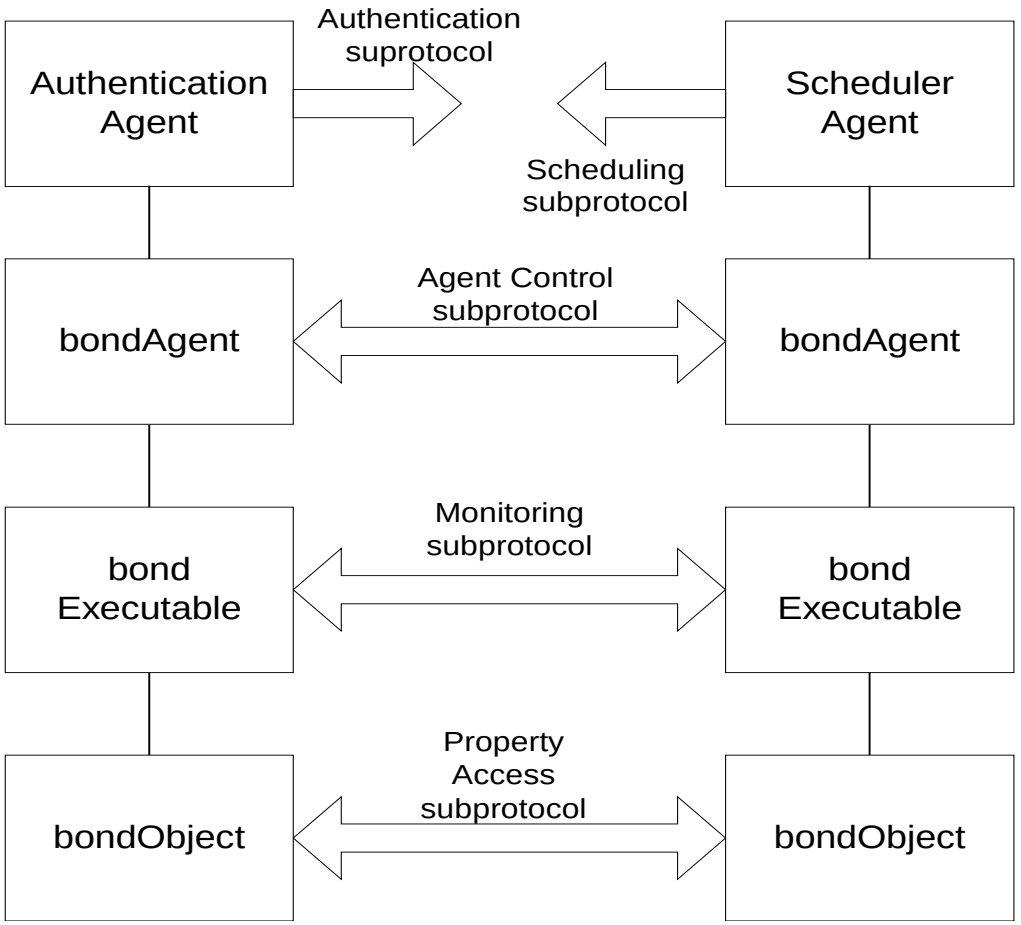
Bond Messages

- Use the format of the KQML agent communication language
- Consists of performative and a set of parameters, e.g. sender, receiver, contents
- e.g.
  - (tell :content get :sender A :receiver B)

Subprotocols

- Closed set of KQML messages
- Solve the problem of semantic understanding
- Subprotocols are inherited from parents
- Examples
  - Property Access Subprotocol
  - Agent Control Subprotocol

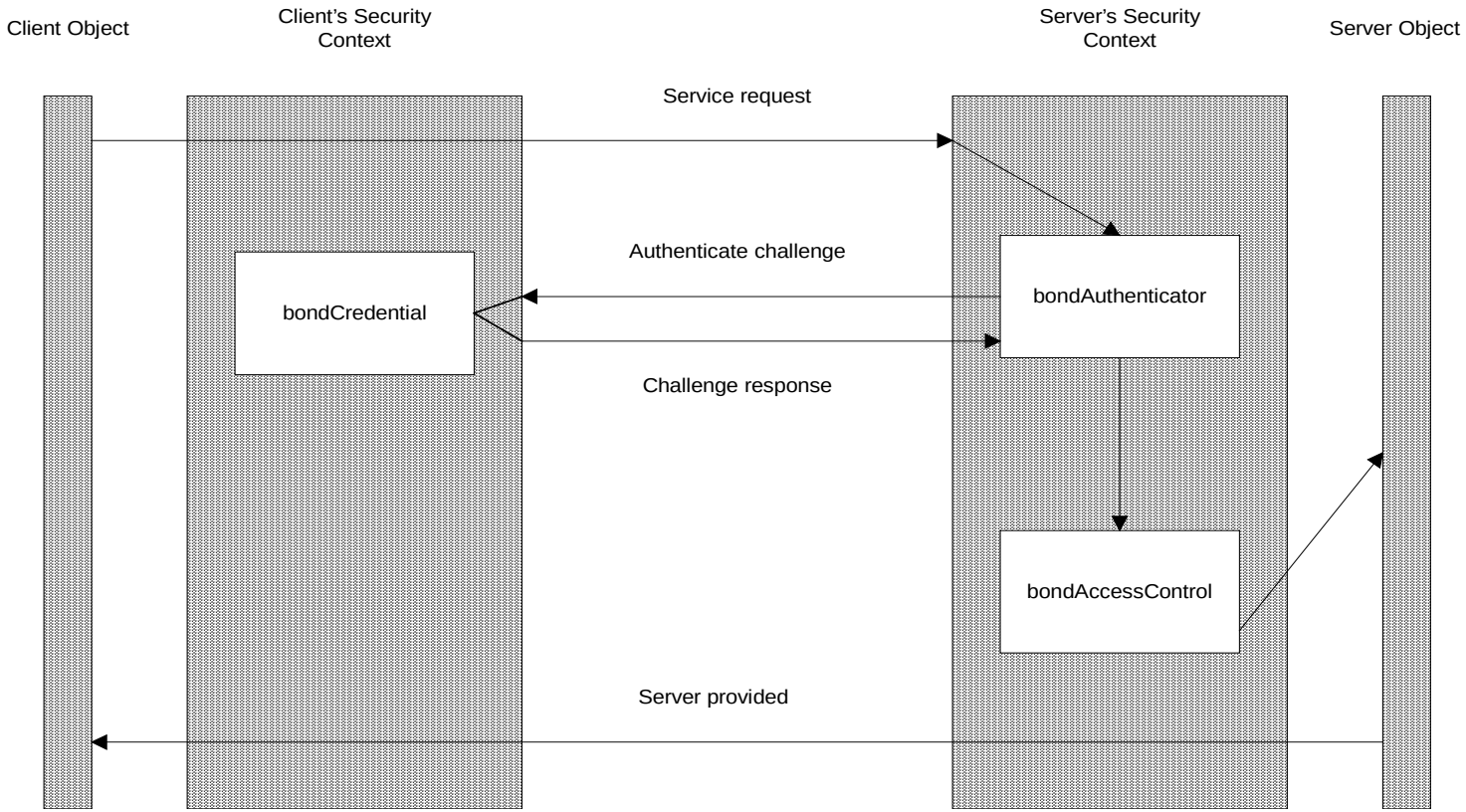
Subprotocol inheritance



Other Bond Concepts

- Shadows
  - communication abstraction
  - extend the notion of a proxy
- Probes
  - enable aspect-oriented programming

Security probe



Metaobjects
and
Software Agents

- Metaobjects provide synthetic information about network objects

- Agents use metaobjects to access and manipulate network objects

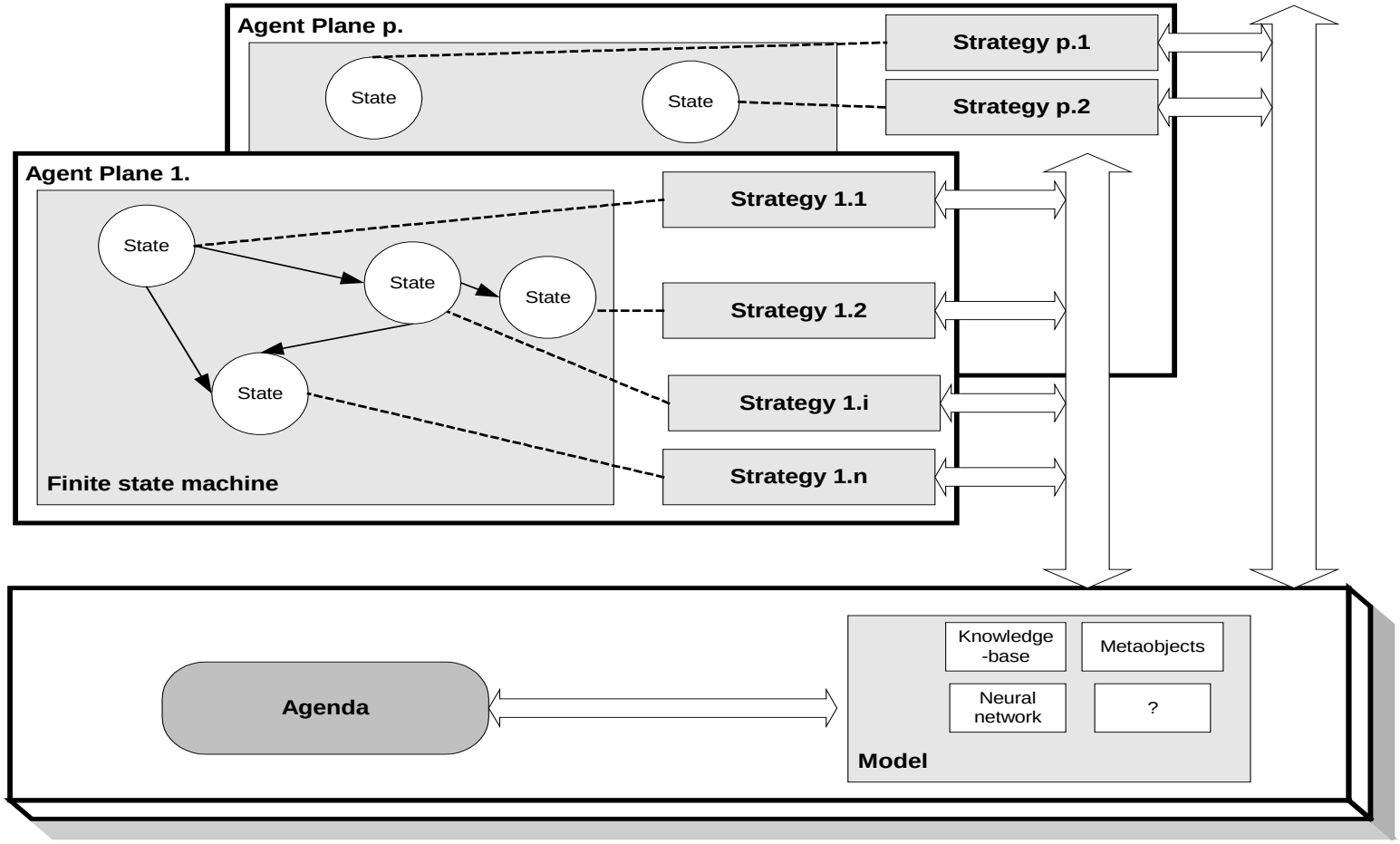
Metaobjects in Bond

- Properties are added dynamically as our knowledge improves
- `set()` and `get()`
- Properties can be read and written remotely using the property access subprotocol

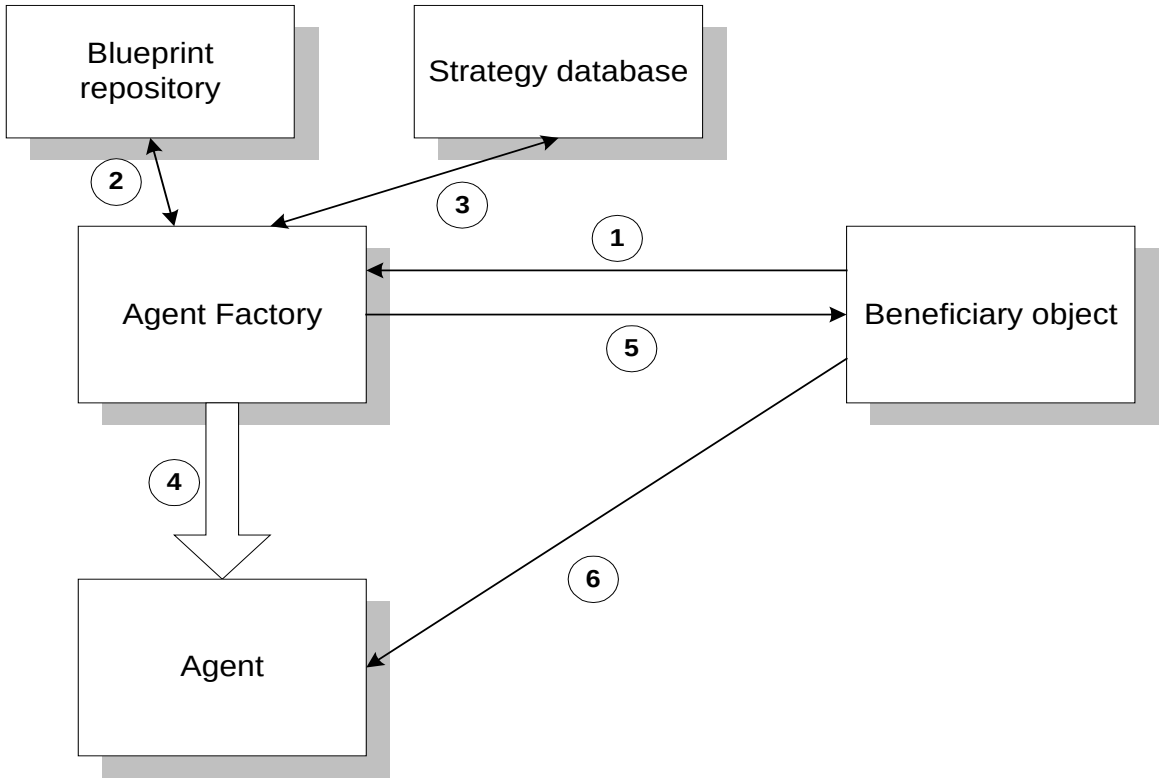
Software Agents in Bond

- Finite State Machine
  - Each state has an associated strategy
- Model of world
  - Collection of the information an agent has about the world
- Agenda
- Blueprint language - describes the structure of an agent
- Agent Factory - creates agents based on a blueprint

The agent model



Agent assembly



Advantages of Software Agents

- Reusability of the strategies
  - Strategy database
  - Simple strategies can be smoothly replaced with increasingly complex ones
- Mobility

Summary

- Biological metaphors from protein structure, genetics and immunology useful for designing complex systems.
- The Bond system uses metaobjects and software agents to implement the metaphors.