Group Abstractions for
Distributed and Grid Computing Systems

José C. Cunha (jcc@di.fct.unl.pt)
CITI – Centre for Informatics and Information Technologies
Dep. Informática, Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa

prepared for the NeSC event on Agent-based Grid Computing
Edinburgh, 19 February 2007
Contents

1. Need for new abstractions and models
2. Benefits of group-based models
3. A research agenda for exploiting groups in distributed and grid computing environments
4. The GroupLog approach, its instances, and applications
5. Conclusions
6. Annex: a summary of the TransGRID project at CITI
Distributed and Grid Computing Systems

- Increasing levels of interaction among components
- New forms of dynamic behavior
  - Due to mobility
  - Due to more frequent changes in system and application configurations
  - Due to changes in interaction and behavior
- Increasing scale in terms of system and application components
Application Characteristics

- Large volumes of data, requiring
  - Efficient and intelligent management and search
  - Parallel and distributed processing
- Dynamic, distributed, and mobile application entities, requiring appropriate management of
  - Structure, interaction, and coordination
- Integration of distributed, heterogeneous components in highly dynamic and interactive environments
- Dynamic organisation of small, medium, or large scale collections of distributed, intelligent agents
Research dimensions

- New solutions are required to provide
  - Abstractions and models for distributed application design and development
  - Associated tools and environments
  - Support infrastructures

- Ongoing developments in distributed - grid computing
  - Towards more robust, stable and standard infrastructures and support architectures
  - Setting up the basis to build higher level abstractions, models and environments
Abstractions and Models

- Design Patterns
- Dynamic Groups
- Distributed Intelligent Agents

How the above can be combined to allow systems to be modeled as groups of agents, which may sometimes exhibit well-identified patterns of structure, interaction, and behavior.

How this approach can contribute to ease the tasks of specifying, composing, developing, understanding dynamic, distributed, large-scale applications.
Design Patterns

- As a way to abstract commonly occurring structures and behaviors in distributed and grid applications
- And how they can be integrated in software development and execution support environments
- Allowing their manipulation during software development, e.g. to ease the building of PSEs
- And during application execution, to support coordination and autonomic behavior

- But sometimes new patterns of behavior emerge dynamically and need to be identified, and requiring decisions to be made dynamically...
Distributed Intelligent Agents

- How agent abstractions can be exploited to support reasoning, planning, intelligent decision support and intermediation
  - Between the user and a PSE
  - And to support reactive and autonomic behavior at the application and at system levels
Dynamic Groups

- As organisation and cooperation paradigm to support
  - Scale, dynamism, and mobility, eg for local or ad-hoc communities in mobile worlds, and for dynamic environments, eg, grid
  - Appropriate forms of interaction and coordination in small, medium, or large scale organisations, possibly hierarchical. Exploit forms of shared knowledge, and information, and trust relationships among group members, and for specialisation of services and cooperation
  - As units of system or application composition to help build and manage complex and dynamic organisations
Benefits from using Groups

- Geographical location / proximity
- Local and spontaneous communities in mobile worlds
- Structuring units in hierarchies
- More efficient forms of interaction
- Trust relationships
- Specialisation of services
- Cooperation:
  - parallel / load balancing / fault tolerance
  - access to a shared logical state
Groups for structure and organisation

- Collections of agents which share common attributes
- Common logical characteristics shared by group members
  - Common computational or communication behaviors
  - Common goals in a society of agents
  - Need of sharing common resources and information
  - Cooperation towards providing common service functionalities with specific constraints
    - Performance
    - QoS
    - Cost parameters
Groups for scalability(2)

- By allowing hierarchies of entities where a group member can be an individual entity or another group
- Important in large-scale and complex organisations
- Allowing confinement of local and global policies
- And more flexible and efficient forms of communication and information dissemination
Groups for modelling dynamic systems (3)

- By providing consistency of views among the group members
- By supporting forms of cooperation among group members, including a shared group state
- Or to manage components with common properties
- By allowing dynamic change of group membership
Groups as units of system composition

- Groups can appear at distinct abstraction levels:
  - At application level
  - At programming level
  - At system level

- Groups can be considered as programming units and used to build hierarchies
  - From its outside, a group can be viewed as an object, an agent, or a service, through an well-defined interface (like a set of methods, or ports), and with an internal behavior, hidden from the outside
  - Separation between the group interface and its internal behavior allows implementing local policies, internal to a group, in a transparent way
distributed processes within a group

internal interaction via
messages or
via shared space

group interface
Groups as units of system composition

- A group can support a reactive or a pro-active, and goal-oriented behavior.
- It is possible to organise a distributed application or system in terms of collections of multiple groups, each responsible for a local service and policy, and globally managed by having global coordination and policies for overseeing and deciding on global strategies.
Groups as structuring units may include processes or other groups.
Another perspective

- Many distributed applications require the ability to capture and identify common attributes and their changes related to distributed and dynamic entities evolving in large-scale environments (like the Grid).

- The need to identify such attributes and their changes can become a critical concern, for example:
  - For intelligent strategies for resource management, depending on changing cost and resource usage.
  - To dynamically form ad-hoc groups:
    - As spontaneous identification of communities of interests (e.g., geoproximity between mobile users).
    - As dynamic definition of common interests, in reaching common goals, sharing common knowledge and functionalities, and contributing to common tasks.
Another perspective(2)

- The dynamic identification of groups as emerging from dynamically identified patterns of behavior or from the intention of pursuing common goals.
- This can become a powerful mechanism to guide strategies for autonomic management of complex distributed systems and applications.
A research agenda

To investigate a group-based framework by providing a two-level approach:

- **[Group specification and management]**: for the organisation in terms of groups of entities (as objects, agents, or services), where a group is a structuring unit:
  - With a public interface
  - And well-defined internal behavior

- **[Dynamic group discovery]**: to dynamically discover and identify groups in a distributed environment, being guided by a definition of the common attributes which represent common characteristics of each group

- The two levels are orthogonal and can be developed independently
Implicitly formed groups

- On having identified which entities exhibit such common attributes, repositories can be updated with the corresponding information, and global coordination policies can then be applied, by creating the so identified groups and aggregating the corresponding entities into the newly formed groups.

- Such an approach is being currently used to design a collaborative mobile framework for multimedia applications, using the concepts of explicit and implicit groups, to support dynamic and adaptive behavior.

- Other application scenarios, for example, cooperative multi-agent applications, and Grid resource management.
Previous and ongoing work

**GroupLog**: an abstract model, defines the basic elements: agents, groups, forms of interaction

- Designed to allow distinct instances, at distinct abstraction levels
- A logic-based instance of the model: distributed Prolog based
- A Java-based instance of the model: JGroupSpace (built on top of JavaGroups)
- With distributed, cluster-based implementations
- Being extended for Grid Services.

Distributed problem solving with GroupLog:

- Examples illustrate the applicability to a large diversity of distributed applications
- Ongoing work exploits GroupLog for collaborative mobile applications
A **GroupLog system**: a collection of distributed agents, able to:
* **Communicate** through interface predicates
* **Access the Group Shared State**
* **Join groups** to participate in coordination activities
Groups, as an organisation and cooperation paradigm in distributed systems.

A large complex system organised in groups, which may be further structured forming hierarchies.

Interactions among group members are more easily managed due to its smaller scale, thus enabling more appropriate coordination paradigms.
To exploit group concepts, in order to handle scalability, dynamism and mobility.

A high-level group-oriented model:
- for the dynamic organisation of distributed agents
- Integrating point-to-point, multicast, and logical shared-memory interaction models
Agents and Groups in GroupLog

- **Agents:**
  - Computational entity (an object, an agent, or a service)
  - Internal configuration
  - Guarded communication
  - Remote entry invocation
  - One-to-one, one-to-many

- **Groups:**
  - Dynamic organisation of agents
  - Integrating direct and indirect forms of communication between group members:
    - (one-to-one, one-to-many, shared group space)
The following slides give a summary of the logic based instance of the GroupLog model.

However, the model was designed to allow distinct semantic interpretations of its constructs: agents and groups, and interactions mechanisms.

Ongoing work: to adapt the model abstractions to:
- Collaborative mobile applications
- Grid environments: an higher level of system organisation is required, allowing groups of agents and services to be orchestrated in order to provide adequate solution to scale, dynamism, and mobility issues; this can be built on top of standard layers, for example, for service-oriented architectures.
L₁ - Dynamic structuring units of program entities (Agents)

Structuring communication and synchronisation

- **Name**
  - Its type

- **Clause Context**
  - Local knowledge

- **Interface Context**
  - A well-defined interface

- **Agent Behaviour**
  - Models actions when it interacts with other agents

- **Agent Creation**
  - A clause models the initial actions and configuration
Agent Behaviour

Each interface is defined by a set of

**Interface Rule**

**Current configuration:**
the configuration of the agent

**Interface:**
the signature of the interface predicate

**Pre-Actions:**
the actions that the agent needs to execute *before* change to new configuration

**New Configuration:**
the new agent configuration after executing the guarded actions

**Post-Actions:**
the actions that the agent needs to execute *after* change to new configuration
L₂ - Dynamic grouping of agents (Groups)

Structuring the set of agents and supporting their cooperation

Group

Name
Clause Context
Interface Context
Group Behavior
Group Creation

Group Shared State: A multiset of atoms shared by the group members

Group Membership: The agents and groups belonging to the group
Group Membership

The group is a composition of agents and groups, which changes dynamically.

- A new entity can join a group
- An entity can leave a group

The members of the group are hidden from the outside.

This isn’t allowed

The communication is through the group interface predicates.
Shared Group State

The members of the group may coordinate by accessing the shared group state.

The group table allows the coordination of the philosophers through the shared state.

Dyning Philosophers

These philosophers suspend until one of the others frees the forks.

These predicates are blocking.
The group notion is important to specify:

* a group of agents, with a shared knowledge
* the dynamic evolution of the system
* the coordination of the agents in a group
* hierarchical organisations
Annex
TransGrid: a CINTI Research Project
coordinated by José C. Cunha

- To improve parallel and distributed environments for complex problem solving, in computational clusters/grids.
- Multiple research streams:
  - Parallel and Distributed Processing
  - Multimedia and Graphics
  - Human Language Technology
  - Geological, Materials, Environmental Sciences
- Dimensions:
  - Applications
  - Abstractions and Models
  - Tools and Environments
  - Distributed Execution
1 - Application Classes

- Parallel Text Mining;
- Collaborative Mobile Multimedia;
- Distributed Simulation, Visualisation and Steering;
- Distributed Intelligent Agent Systems.
Application Characteristics

- large volumes of data (text or images):
  - efficient search, parallel processing and input/output
- dynamic, distributed, mobile application entities:
  - appropriate structuring, interaction, coordination
- integration of distributed heterogeneous components in a highly interactive environment:
  - supporting dynamic reconfiguration of components, and execution at a small or large scale
- organisation, management, coordination in a distributed agent system:
  - dynamic organisation and intelligent agents.
2 - Abstractions and Models

- Design Patterns
- Dynamic Groups
- Distributed Logic Agents
Design Patterns for Grid Workflows

Cecília Gomes, Omer Rana, and José Cunha

- Patterns as first-class entities
- To use patterns to abstract commonly occurring structures and behaviours in distributed dynamic environments
- To integrate them into grid environments
GroupLog
José Cunha, Fernanda Barbosa, Carmen Morgado, Jorge Custódio, Nuno Correia

- Groups, as an organisation and cooperation paradigm in distributed systems.
- A large complex system organised in groups, which may be further structured forming hierarchies.
- Interactions among group members are more easily managed due to its smaller scale, thus enabling more appropriate coordination paradigms.
To exploit group concepts, in order to handle scalability, dynamism and mobility.

A high-level group-oriented model:
- for the dynamic organisation of distributed agents
- Integrating point-to-point, multicast, and logical shared-memory interaction models
To develop distributed computing models and architectures for logic programs based on tabling.

A basis for:

- supporting reasoning, planning, intelligent decision support, and intermediate between the user and the system levels.
3 - Tools and Environments

- To observe application behaviour for resource management and for system/application dynamic reconfiguration;
- To support integrated testing and debugging;
- To support flexible infrastructures for tool integration
4 - Execution infrastructure

To evaluate infrastructure support for the selected applications, namely concerning:
- parallel i/o and file systems
- support for group abstractions
Conclusions on Challenging Requirements

- Higher degrees of user interaction, increased flexibility in observation, control, or modification of application components.
- Multidisciplinary applications, interactions between distinct sub-models, and distributed user collaboration.
- Dynamic applications and environments, as new application components or system resources are dynamically generated, made unavailable, or mobile.
- Spatial distribution of application components and system resources, at small, medium or large scales.
Conclusions

- Main challenges:
  - New concepts
  - Models, tools and support environments

- They are driving significant research and development efforts that will have great impact upon many areas