7
Dynamic Contextual Protocol Information Provision

The methodology proposed in this thesis can be similarly applied to contextualize, concretize, represent and compose contextual protocol information. A *protocol* can be defined as a set of directives; and a *directive*, in turn, can be defined as an action prescription that should be followed in order to fulfill the conditions of a goal.

The DynaCIP solution (meaning *Dynamic Contextual Protocol Information Provision in Open MAS*) is presented in the following subsections.

7.1.
DynaCIP

In [Felicissimo, 2008c], the seminal ideas of DynaCIP are proposed. In [Rocha, 2008], such ideas evolved in order to support the development of a framework for service negotiation simulation in the domain of next generation wireless networks [Berezdivin *et al.*, 2002]. Both works follow an *action* and *effect* approach, *i.e.*, agents aware of the valid protocols in a given context are more likely to perform correctly, and so, being able to achieve their goals faster.

Figure 39 illustrates the DynaCIP created ontology in which the *Protocol* concept is the main asset. In the ontology, the *Role* concept encompasses the instances of all roles of the system and each role instance has associations with its protocols (via the *hasProtocol* property) and with its organization (via the *isPlayedIn* property). The *Organization* concept encompasses the instances of all organizations and each organization instance has associations with its protocols, with itself (via the *hasMainOrganization* property for representing its main organization) and with its environment (via the *isIn* property). The *Environment* concept encompasses the instances of all environments and each environment instance has associations with its protocols and with itself (via the *belongsTo* property for representing its owner environment). The *Protocol* concept encompasses the instances of all protocols and each protocol instance has associations with its di-

rectives (via the *hasDirective* property). The *Directive* concept encompasses the instances of all directives of the system.

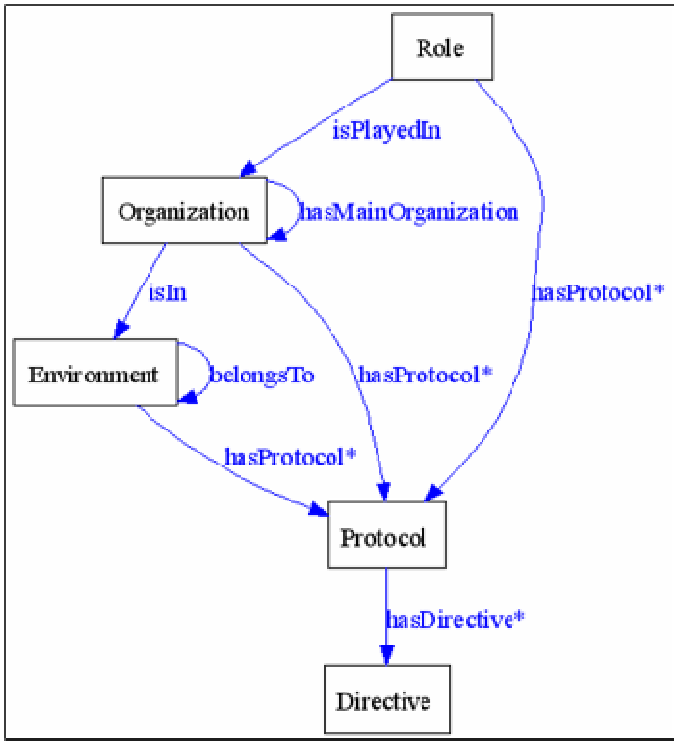

Figure 39 – The DynaCIP ontology

7.2. DynaCIP at Work

As a typical example of the applicability of DynaCIP in MAS, two universities, in two different countries, *PUC-Rio* in *Brazil* and *Paris VI* in *France*, for instance, are considered. Both organizations have a location service capable of determining the position of agents inside them (such as the MoCA's Location Inference Service [Rubinsztejn *et al.*, 2004]). The information is given in terms of a symbolic location that associates a name with each of the smallest areas, distinguishable by the location service. Examples of atomic spaces may be organizations, laboratories, seminar rooms, offices, corridors, etc., and they are populated by agents that have some association with those spaces (*e.g.*, professors, students, researchers, administrative staff members, etc.). For instance, when two organizations have some sort of cooperation, a member of one institution may be a temporary visitor at another one.

In the example, the following motivating scenario for scheduling a meeting is proposed:

Carolina, a student from PUC-Rio, arrives at Paris VI for an internship of one year. Carolina wants to schedule a meeting with Amal, a professor at Paris VI, that will be her advisor during the internship. Carolina does not know how to proceed, i.e., if, in Paris VI, it is correct to contact directly a Professor to ask for a meeting or if the local protocol is to send an email with the request; moreover, if it is allowed to speak English; etc.

For the example, the following environment protocols were created:

An Environment Protocol for Communicating: In all countries, a national language is defined for communication among people.

Concrete Environment Protocols for Communicating: the national language for communication among people is (a) in *Brazil*, *Portuguese*; and, (b) in *France*, *French*.

Figure 40 illustrates the contextual protocols for communication concretized in the *Brazil* and *France* environments. The *Brazilian* protocol is represented by the *BrazilianProtForCommunicating* protocol instance and the *French* one by the *FrenchProtForCommunicating*.

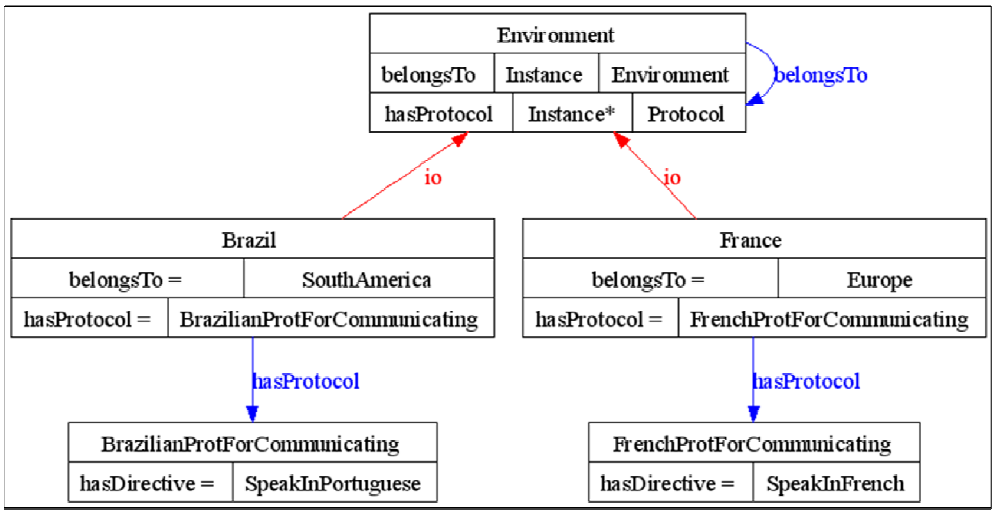

Figure 40 – Concrete environment protocols for communicating

The following role protocols were also created for the example:

A Role Protocol for Scheduling Meetings: Professors should inform the correct manner they want to be contacted for scheduling meetings.

Concrete Role Protocols for Scheduling Meetings: in order to schedule meetings, (a) professors of *Paris VI* should be contacted by email; and, (b) professors of *PUC-Rio* should make public 30% of their agenda.

Thus, now, Carolina is informed by DynaCIP that, for scheduling a meeting in *Paris VI*, she should send an *email*, in *French*, to professor Amal.

The following contextual protocols were also created in the example in order to illustrate the other basic contexts for a DynaCIP MAS.

An Organization Protocol for Answering Emails: Universities should define a deadline of less than *one week* for academic emails to be answered.

A Concrete Organization Protocol for Answering Emails: (a) In *PUC-Rio*, academic emails should be answered in less than *three days*.

An Interaction Protocol for Scheduling Meetings: in order to schedule meetings, PhD students should contact their advisors or advisors' secretaries.

A Concrete Interaction Protocol for Scheduling Meetings: in order to schedule meetings, (a) in *PUC-Rio*, *Prof.Lucena's PhD students* should call his secretary.

Figure 41 illustrates the *ProtForSchedulingMeetingsWith* (a *Protocol* sub-concept) and its instance (*LucProtForSchedulingMeetingsWithPhDStds*), both created to concretize the interaction protocol between Lucena (represented by an instance of the *PUCRioProfessor* role sub-concept) and his PhD students (represented by the *LucenaPhDStudents* instance of a *PhDStudent* role sub-concept). The figure also illustrates the *PUCRio* organization instance and its *PUCRioProtForAnsweringEmails* protocol instance.

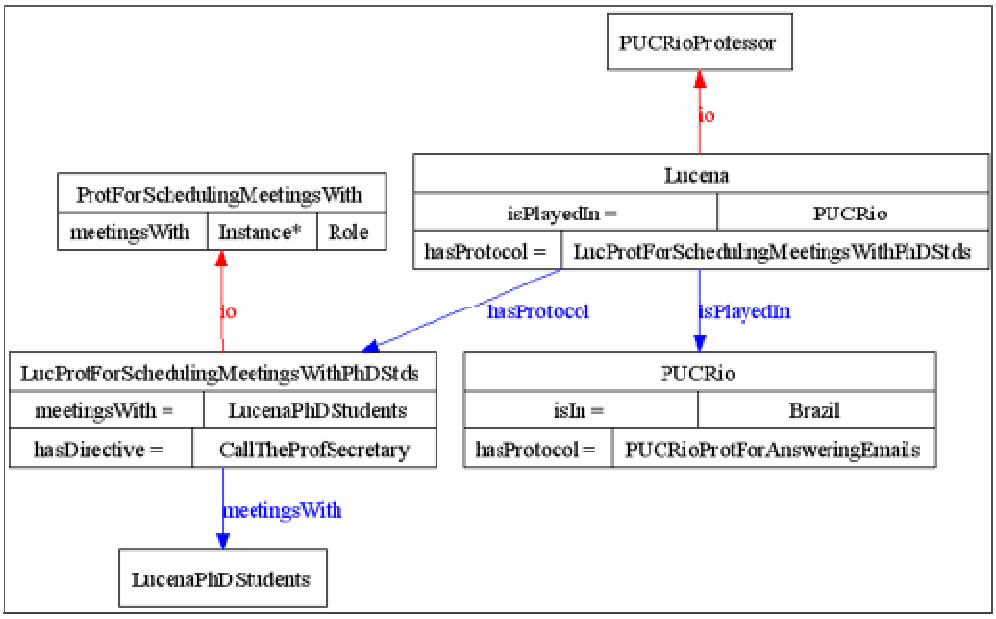

Figure 41 – Concrete contextual protocols of PUC-Rio

In order to compose contextual protocols, the system developer should write rules, following a similar syntax of the DynaCROM rules for norm composition. Code 34 presents the basic rules wrote for a DynaCIP MAS.

Code 34. DynaCIP rules to compose contextual protocols

```
(1)  [DynaCIPRule_EnvWithOEnvProtocols:
(2)    hasProtocol(?Env,?OEnvProtocols)
(3)     <- hasProtocol(?OEnv,?OEnvProtocols),
(4)        belongsTo(?Env,?OEnv)]

(5)  [DynaCIPRule_OrgWithMOrgProtocols:
(6)    hasProtocol(?Org,?MOrgProtocols)
(7)     <- hasProtocol(?MOrg,?MOrgProtocols),
(8)        hasMainOrganization(?Org,?MOrg)]

(9)  [DynaCIPRule_OrgWithEnvProtocols:
(10)   hasProtocol(?Org,?OrgEnvProtocols)
(11)    <- hasProtocol(?OrgEnv,?OrgEnvProtocols),
(12)       isIn(?Org,?OrgEnv)]

(13) [DynaCIPRule_RoleWithOrgProtocols:
(14)   hasProtocol(?Role,?OrgProtocols)
(15)    <- hasProtocol(?Org,?OrgProtocols),
(16)       isPlayedIn(?Role,?Org)]
```

For instance, the *DynaCIPRule_RoleWithOrgProtocols*, presented in Code 34 (lines 13 to 16), composes the protocols of a role with the protocols of its organization. More precisely, considering *Lucena* as an example of the given role instance, the following process is executed, according to the domain ontology instance illustrated in Figure 41: in (16), the '*?Org*' variable is instantiated with the

PUCRio inferred value, when the '*?Role*' variable is instantiated with the *Lucena* given value; in (15), the '*?MOProtocols*' variable is instantiated with the *PUCRio-ProtForAnsweringEmails* inferred value; and in (14), the inferred protocol is added as a new protocol of *Lucena*. The result of the process is that professor Lucena should answer his emails in less than *three days*.

For the process of composing protocols and, consequently, composing directives, the same inference rule engine of DynaCROM (illustrated in Figure 10) is used. The only difference is that, instead of composing normative contexts, the rules written for DynaCIP compose contextual protocols.