EPSRC Engineering and Physical Sciences Research Council

ORCHID

Exploring Obligation and Prohibition as a means to support Flexible Autonomy

Daniela Dybalova (dxd@cs.nott.ac.uk)
Agents Lab / Mixed Reality Lab
School of Computer Science
University of Nottingham

General Approach

We explore the ways in which flexible autonomy can be supported using models based on the expression of obligations and prohibitions.

This is done in the context of a disaster scenario. The work involves developing a system that links with existing atomic orchid and allows the obligations and prohibitions for each agent to be expressed using 2OPL annotation language for describing constraints.

A tuple space (JINI JavaSpaces) is used to link these to the atomic game allowing these to drive the actors within the games. The tuple space contains complete state of the game with its history and also all obligations and prohibitions.

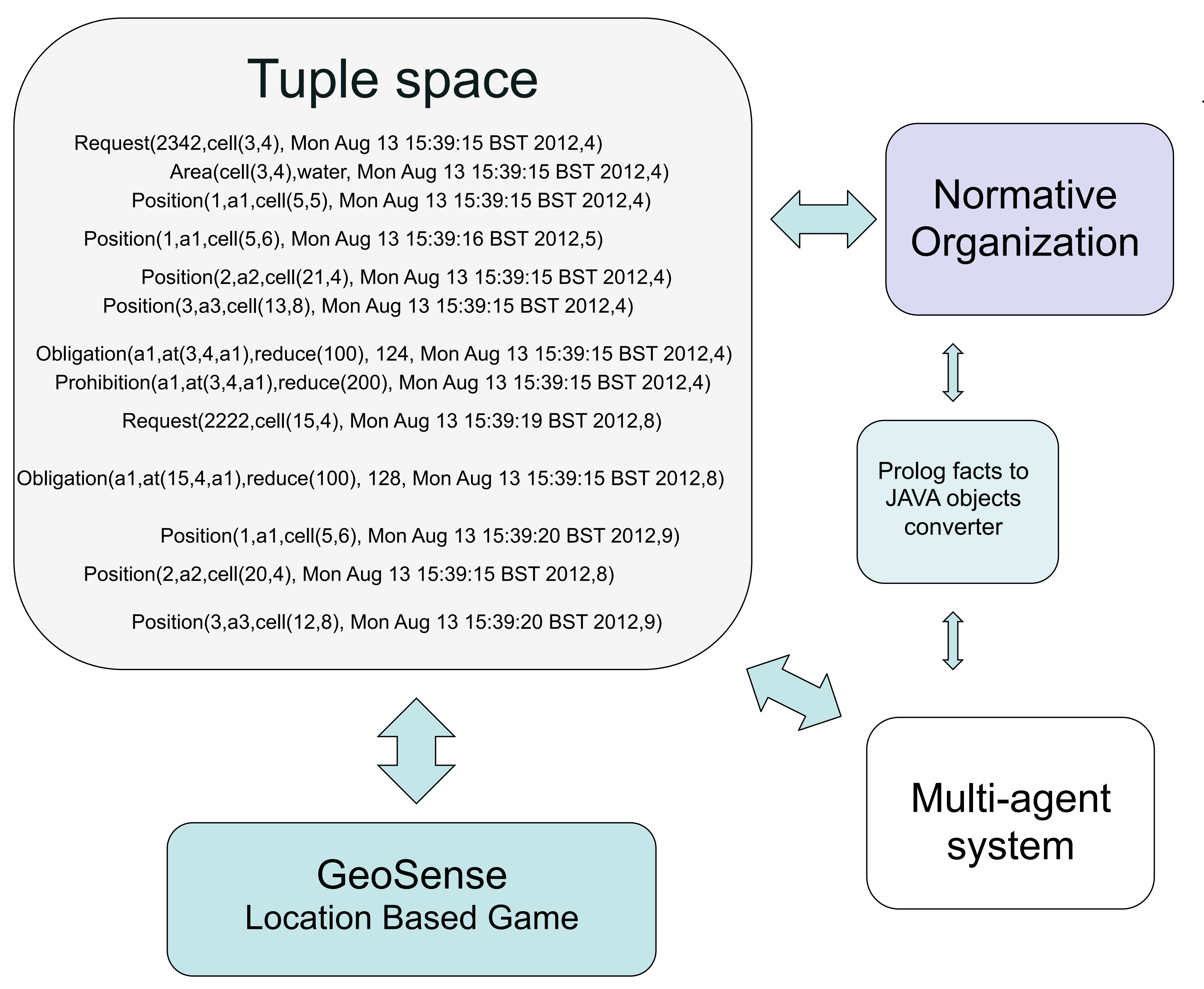
Normative Systems

Normative agent-based system is organized by means of mechanisms to represent, communicate, distribute, detect, create, modify, and enforce norms, and mechanisms to deliberate about norms and detect norm violation and fulfilment.

Autonomy in this system is as an authority of an agent to perform an action in current circumstances without seeking human approval. Norms in the form of obligations and prohibitions are used to constrain agent's autonomy.

Norm-aware agents programmed in N-2APL are able to reason about norms that are in a form of obligations and prohibitions with assigned fixed numerical priorities and corresponding sanctions. Such a coordination mechanism of multi-agent systems allows the autonomy of agents to be dynamically adjusted with the use of norms created by the organization.

Overall Architecture



```
Request(2342,cell(3,4), Mon Aug 13 15:39:15 BST 2012,4)
Area(cell(3,4),water, Mon Aug 13 15:39:15 BST 2012,4)
Position(1,a1,cell(5,5), Mon Aug 13 15:39:15 BST 2012,4)
Position(1,a1,cell(5,6), Mon Aug 13 15:39:16 BST 2012,5)
Position(2,a2,cell(21,4), Mon Aug 13 15:39:15 BST 2012,4)
Position(3,a3,cell(13,8), Mon Aug 13 15:39:15 BST 2012,4)
Obligation(a1,at(3,4,a1),reduce(100), 124, Mon Aug 13 15:39:15 BST 2012,4)
Prohibition(a1,at(3,4,a1),reduce(200), Mon Aug 13 15:39:15 BST 2012,4)
Request(2222,cell(15,4), Mon Aug 13 15:39:19 BST 2012,8)
Obligation(a1,at(15,4,a1),reduce(100), 128, Mon Aug 13 15:39:15 BST 2012,8)
Position(1,a1,cell(5,6), Mon Aug 13 15:39:20 BST 2012,9)
Position(2,a2,cell(20,4), Mon Aug 13 15:39:15 BST 2012,8)
Position(3,a3,cell(12,8), Mon Aug 13 15:39:20 BST 2012,9)
```

Normative Programming

2OPL (organization) [1]

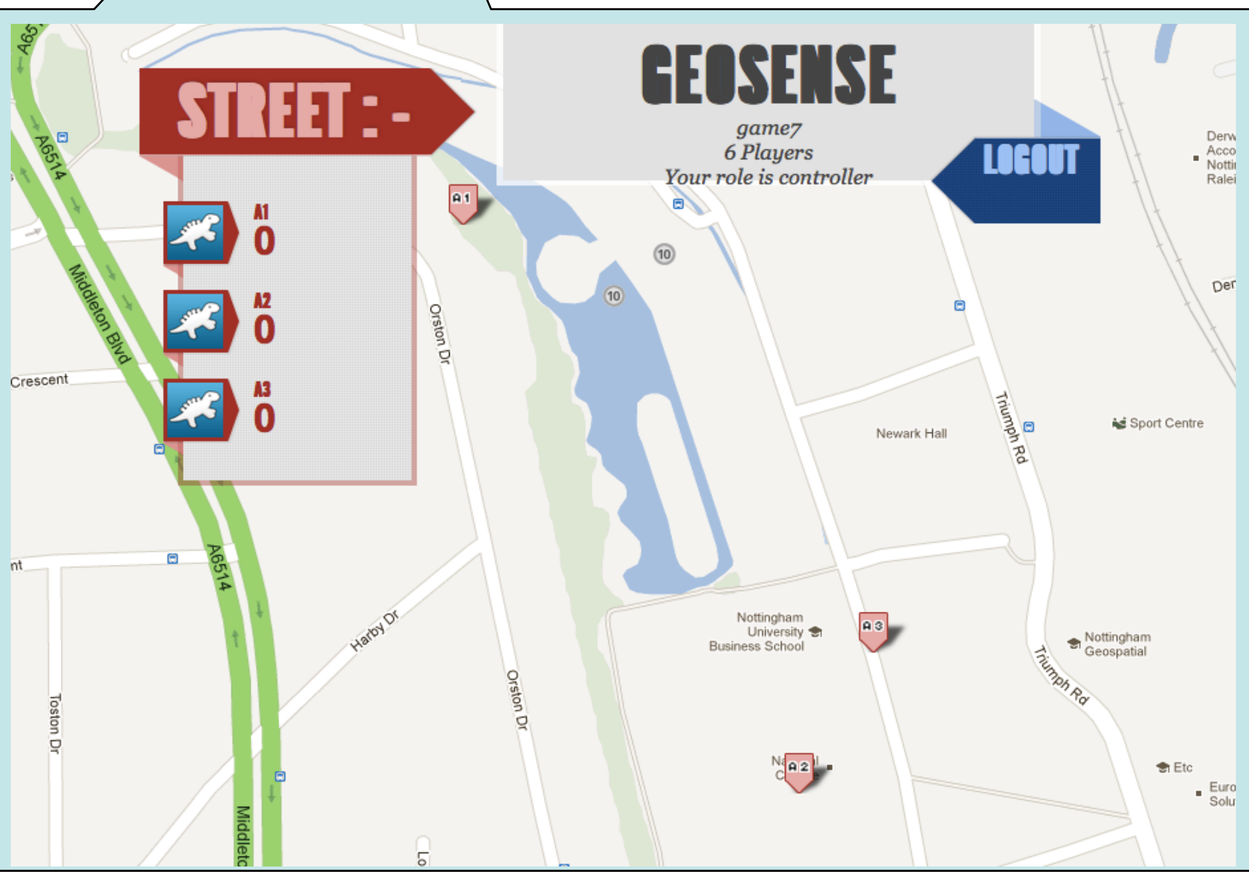
```
Facts:
norm(
    pick_up_coin(Agent),
    Agent, // the subject agent
    ( at(X,Y,Thing), coin(Thing), agent(Agent), clock(Now),
      Deadline is Now + 120 ), //precondition
    obligation([at(X,Y,Agent)], Deadline, [reduce(Agent,100)])
    ).

norm(
    avoid_water(Agent),
    Agent, // the subject agent
    ( water(X,Y), agent(Agent) ),
    prohibition([at(X,Y,Agent)], [reduce(Agent,200)])
    ).
```

N-2APL (agent) [2]

```
event(obligation(Goal,Deadline,Sanction),space) <- true |
{
   adoptObligation(Goal, Deadline, Sanction)
}
event(prohibition(State,Sanction),space) <- true |
{
   adoptProhibition(State,Sanction)
}
```

```
⟨Agent_Prog⟩    =  [ "Beliefs:" { ⟨belief⟩ } ],
                   [ "Goals:" ⟨goals⟩ ],
                   [ "Plans:" ⟨plans⟩ ],
                   [ "PG-rules:" { ⟨pgrule⟩ } ],
                   [ "PC-rules:" { ⟨pcrule⟩ } ]
                   [ "PR-rules:" { ⟨prrule⟩ } ]
⟨goals⟩         =  ⟨goal⟩ { "," ⟨goal⟩ } ;
⟨goal⟩          =  ⟨atom⟩ ":" ⟨deadline⟩;
⟨pgrule⟩        =  ⟨goalquery⟩ "<-" ⟨belquery⟩ "|" ⟨plan⟩ ":" ⟨duration⟩;
⟨goalquery⟩     =  ⟨goalquery⟩ "and" ⟨goalquery⟩ | ⟨goalquery⟩ "or" ⟨goalquery⟩ | "(" ⟨goalquery⟩ ")" | ⟨atom⟩ ;
⟨belquery⟩      =  ⟨belquery⟩ "and" ⟨belquery⟩ | ⟨belquery⟩ "or" ⟨belquery⟩ | "(" ⟨belquery⟩ ")" | ⟨literal⟩ ;
⟨plan⟩          =  ⟨atomic-plan⟩ | ⟨non-atomic-plan⟩;
⟨atomic-plan⟩   =  "["⟨non-atomic-plan⟩"]";
⟨sanction⟩      =  ⟨atom⟩;
⟨deadline⟩      =  ⟨time⟩;
⟨duration⟩      =  ⟨int⟩;
```

Figure 1: EBNF syntax of N-2APL.



Future Plans

The next step in the development is to extend the capabilities of the system so that it matches requirements of the current atomic orchid scenario. That includes mainly introducing a notion of group obligations and prohibitions to the agents.

Another research interest is to explore in which other scenarios this architecture (normative multi-agent system + tuple space) could be used.

[1] Tinnemeier, N. a. M., Dastani, M. M., Meyer, J.-J. C. & Torre, L. V. D. (2009), `Programming Normative Artifacts with Declarative Obligations and Prohibitions', 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology pp. 145-152.

[2] Alechina, N., Dastani, M. & Logan, B. (2012), Programming Norm-Aware Agents, in `Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)', pp. 1057-1064.

BAE SYSTEMS
ACFR AUSTRALIAN CENTRE FOR FIELD ROBOTICS
SECURE
UNIVERSITY OF Southampton