

normHACing – a framework for programming norm-aware multi-agent systems

Daniela Dybalova (dxd@cs.nott.ac.uk)

Agents Lab / Mixed Reality Lab
School of Computer Science
University of Nottingham

normHACing framework

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

Normative MAS are 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.

N-2APL norm-aware agents

Norm-aware agents programmed in N-2APL are able to reason about norms that are in a form of obligations and prohibitions with assigned 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 deployed by the organization.

Implementation of group norms

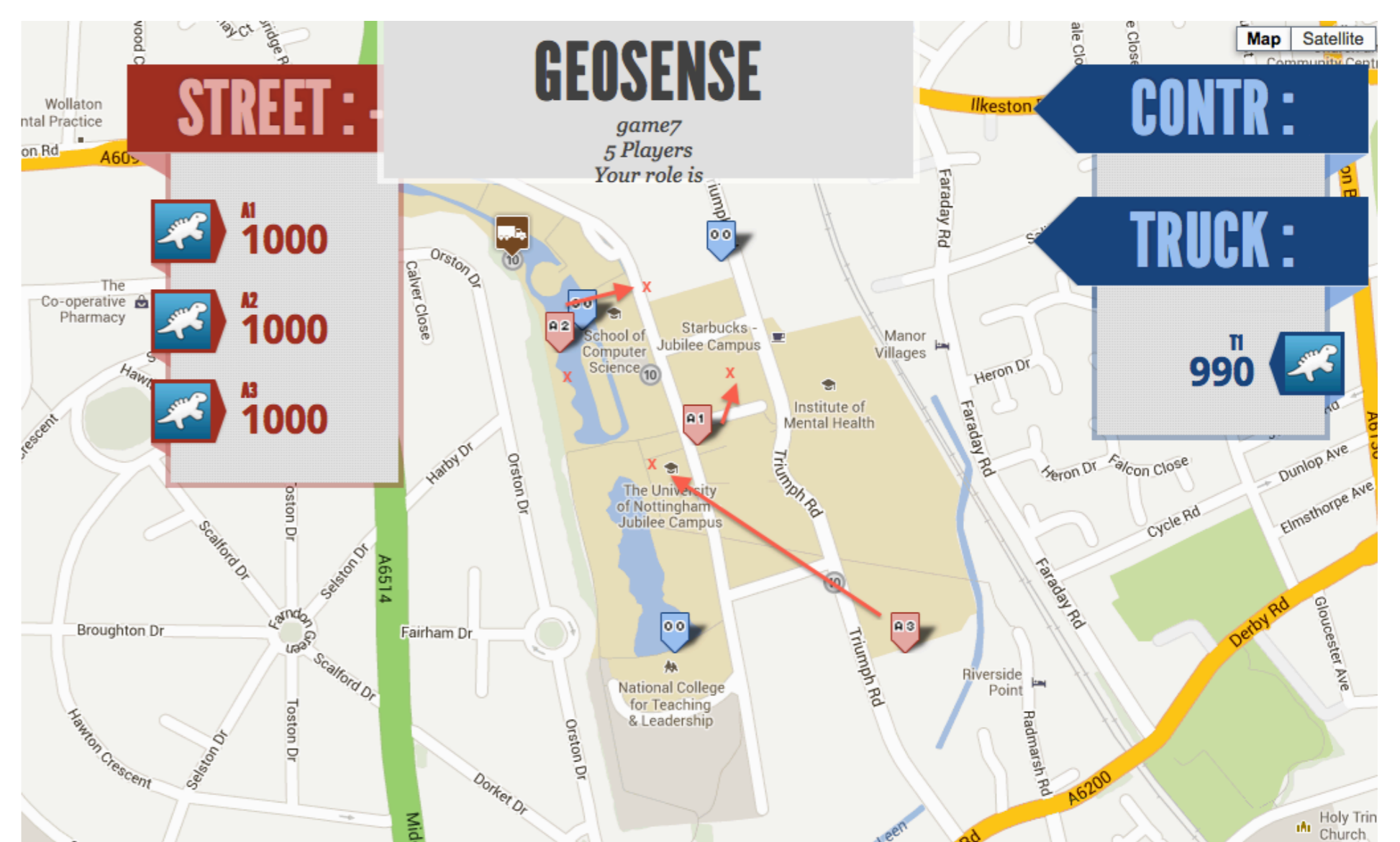
In this framework agents could be regulated not only with obligations aimed at them individually but also obligations that are issued to a of group agents. Implementation of such an obligation requires the agents to have a plan how to participate in the joint action and to be able to infer what is the sanction going to be depending on the actions of other agents.

N-2APL agents have individual plans to enact their part in a group obligation. Agents also know the maximum and sanction they can incur in case the norm is violated.

When an agents deliberate about which plan to adopt it can take into account either the maximum or the minimum value depending whether agent is optimistic or pessimistic.

Agent is sanctioned with the maximum value in case it was solely responsible for the violation and similarly with the minimum in case when none of the responsible agents did their part.

This is done in the context of a disaster scenario. The work involves developing a system that links with a version of existing AtomicOrchid and allows the obligations and prohibitions for each agent to be expressed using 2OPL language for describing constraints. A tuple space is used to link these to the AtomicOrchid allowing these to drive the actors within the game. The tuple space contains complete state of the game with its history and also all obligations and prohibitions^[1].



Taxonomy of group norms^[2]

Actorship / Responsibility	Individual	Representative	Collective
Individual	(1,1) Individual norm specified in a generic (role-based) way: "When a coin is place four pursuer agents should surround the place from north, east, south and west."	(1,2) Individual action, appointed blame: "Leading pursuer agent is responsible for coordinating agents to surround a coin."	(1,3) Individual action, collective blame: "To win the game pursuers must surround the truck from at least three directions."
Representative	(2,1) Appointed action, individual blame: "After surrounding a coin agents have to make reading otherwise pursuers loose 200 points each."	(2,2) Appointed action, appointed blame: "A reading needs to be taken at a request of coordinating agent. Leading pursuer agent is responsible for the reading to taken by a pursuer agent."	(2,3) Appointed action, collective blame: "The agent who reaches the capturing location first has to investigate and capture the truck otherwise pursuers loose the game."
Collective	(3,1) Collective action, individual blame: "Two pursuer agents need to carry a load together to dispose of it."	(3,2) Collective action, appointed blame: "Agents must not be further apart than 5 fields from the closest agent otherwise the leading pursuer agent gets sanctioned."	(3,3) Group action, group blame: "Agents must not be further apart than 5 fields from the closest agent or their team lose the game."

Rules of the game are described as conditional norms with deadlines and sanctions. In the example below is a basic group obligation in 2OPL notation telling pursuers to surround a coin, which was placed by a coordinator.

```
norm(surround_coin(Pursuers),
    Pursuers,
    (pursuers(Pursuers), coin(X,Y,Agent),
      West is X-1, East is X+1, North is Y+1, South is Y-1),
    obligation(
      Pursuers,
      [at(East, Y, A1), at(West, Y, A2), at(X, North, A3), at(X, South, A4)],
      now+15,
      [reduce_score(Pursuers, 300)]
    )
  ).
```

Future Work

- Implementation of a distributed monitoring mechanism, which will support monitoring of norms that require collective actorship with higher level of coordination.
- Evaluation of the framework with a mix of human and agent players. Agents will be taking roles as advisors to players, regular team members and team leaders.

[1] Dybalova, D., Testerink, B., Dastani, M., & Logan, B. (2013). A Framework for Programming Norm-Aware Multi-Agent Systems. In International workshop on Coordination, Organizations, Institutions, and Norm in Multi-Agent Systems.

[2] Huib Aldewereld, Virginia Dignum, and Wamberto Vasconcelos. We ought to; They do; Blame the management! In Coordination Organizations Institutions and Norms in Agent Systems, 2013.