

Organised Adaptation: Lecture 2

Jeremy Pitt¹, Alexander Artikis^{1,2}

¹Imperial College London, UK

²National Centre for Scientific Research “Demokritos”, Greece

Perada Summer School, Rimini, September 6-12, 2008

Programme

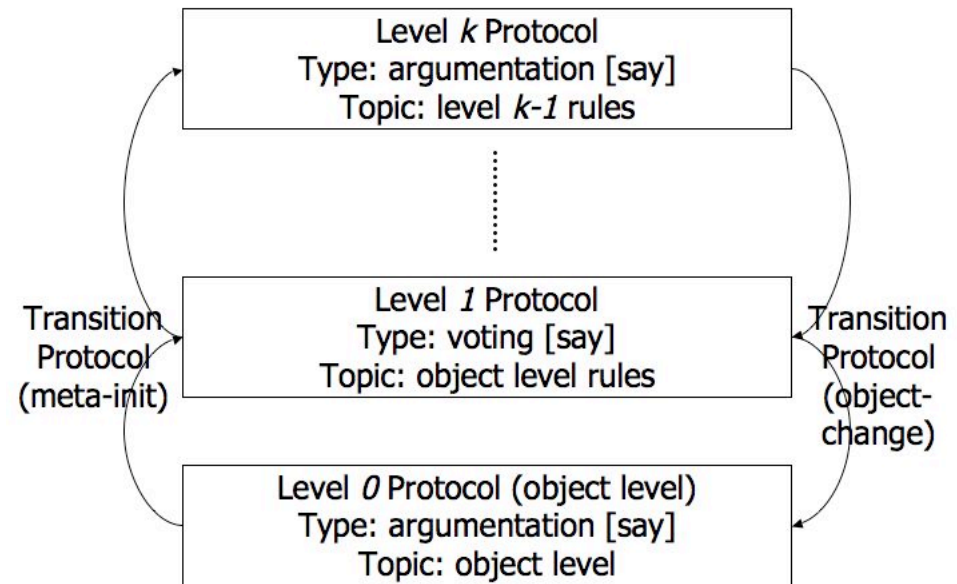
- Lecture one: Organization (what to adapt)
 - Definitions
 - Building blocks
- Lecture two: Adaptation (how to adapt)
 - Mechanism for adaptation (not the only one)
 - Evaluation of adaptation
 - (Not covered: when/why to adapt)
- Lecture three: Tutorial

Lecture 2: Abstract

- Lecture 1 gave a definition of organised adaptation, agent societies, and what can be adapted
- Lecture 2 discusses mechanisms for adaptation: how/when to adapt, and how to evaluate adaptation
- We will briefly look at various technologies for addressing each question
- We will examine in detail one mechanism for addressing the question of how to adapt
- For this, we will use the Event Calculus to formalise a voting protocol
- Lecture 3 will use a tutorial to apply the approach of organised adaptation

How to Adapt

- Protocol Stack
- Level 0: the object level
 - the basic agent society specification
 - also identify: ‘core’ rules (irreplaceable), replaceable rules, adaptable parameters
 - meta-level: rules for changing these rules



Example 1: Ad Hoc Networks

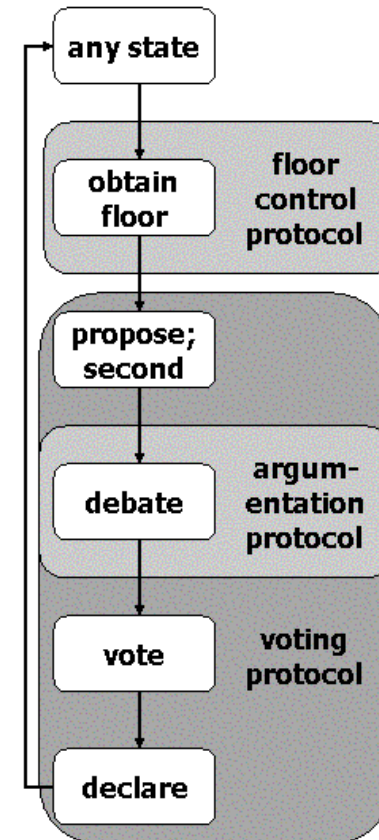
- A transient association of mobile nodes which inter-operate largely independently of any fixed support infrastructure.
 - It may be short-lived, supporting spontaneous rather than long-term interoperation
 - If longer-term operation is required, want to create a backbone (for efficient packet routing)
 - Who gets to be a vertebra?
 - How do you do resource sharing and network provisioning?
- One way to decide these matters is to take a vote . . .

Exampe 2 Virtual Enterprises

- Transient, inter-organizational, cross-border, ICT-enabled collaboration between legally independent entities
- Requirements and Mechanisms for formation, run-time management and dissolution, given that there is no hierarchical structure
 - applications to join
 - action selection: deciding on individual and collective action
 - mapping of tasks to business processes
 - appointment to 'roles' (e.g. controller for protection of personal data or IPR)
 - dispute resolution
- Ostensibly fair way to reach consensus between independent peers is to take a vote ...

Deliberative Assemblies

- A mechanism for adaptation in Ad Hoc Networks and Virtual Organizations is voting
- Leverage work in online deliberative assemblies using RONR (cf. ZENO)
- Coheres several protocols studied in the literature



What's the Problem with e-Voting?



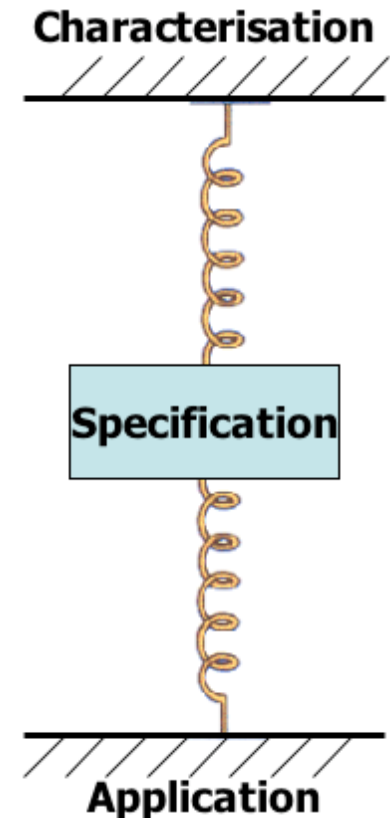
- ACM Statement on eVoting
 - voting systems should enable each voter ... to verify that his or her vote has been accurately cast and to serve as an independent check on the result produced and stored by the system

Enfranchisement = 'Right' + 'Entitlement'

- 'Right' Aspect
 - having the power (being empowered) to vote;
 - denying anyone else the power to object to 'appropriate' exercise of this power; and
 - subjecting inappropriate removal of this power to sanction;
- 'Entitlement' Aspect
 - being entitled to access the 'voting machinery';
 - being entitled to have the vote counted correctly; and
 - being entitled to a 'fair' outcome (i.e. the result is declared according to the way the votes were cast with respect to the standing rules of the committee)

Specification

- Specify a voting protocol as defined in RONR
- Respects 'characterisation' requirements
- Supports 'application' requirements
- Use norm-governed approach to systems specification as described in Lecture 1
- Reminder: formal base is Event Calculus
 - (Note minor notational variation from Lecture 1)



Voting Protocol for Organised Adaptation

- Informal specification of a decision-making procedure according to Robert's Rules of Order (Newly Revised) (RONR)
 - a committee meets and the chair opens a session
 - a member requests and is granted the floor
 - that member proposes a motion
 - another member seconds the motion
 - the members debate the motion
 - the chair calls for those in favour to cast their vote
 - the chair calls for those against to cast their vote
 - the motion is carried or not, according to the standing rules of the committee
- The motions concern adaptable components of the agent specification

An Event Calculus Specification

- Basic Items: Fluents and Actions
- Institutional Powers
- Status of Motions
- Roles & Role Assignment
- Voting and Counting Votes
- Permission and Obligation
- Sanctions and Objections

Fluents

Fluent	Range
<i>sitting</i> (<i>S</i>)	boolean
<i>status</i> (<i>M</i>)	{ <i>pending, proposed, seconded voting</i> (<i>T</i>), <i>voted, resolved</i> }
<i>votes</i> (<i>M</i>)	$N \times N$
<i>voted</i> (<i>Ag, M</i>)	{ <i>nil, aye, nay, abs</i> }
<i>resolutions</i> (<i>S</i>)	list of motions
<i>qualifies</i> (<i>Ag, R</i>)	boolean
<i>role_of</i> (<i>Ag, R</i>)	boolean
pow (<i>Ag, Act</i>)	boolean
per (<i>Ag, Act</i>)	boolean
obl (<i>Ag, Act</i>)	boolean
<i>sanction</i> (<i>Ag</i>)	list of integers

Actions

Action	Indicating. . .
<i>open_session(Ag, S)</i>	open and close a session
<i>close_session(Ag, S)</i>	
<i>propose(Ag, M)</i>	propose and second a motion
<i>second(Ag, M)</i>	
<i>open_ballot(Ag, M)</i>	open and close a ballot
<i>close_ballot(Ag, M)</i>	
<i>vote(Ag, M, aye)</i>	vote for or against a motion,
<i>vote(Ag, M, nay)</i>	abstain or change vote
<i>abstain(Ag, M)</i>	
<i>revoke(Ag, M)</i>	
<i>declare(Ag, M, carried)</i>	declare the result of a vote
<i>declare(Ag, M, not_carried)</i>	

Institutional Power

- Value of a fluent changes as a consequence of an empowered agent performing a designated act
- Agents that occupy particular roles are empowered to perform such acts
- Empowered when certain fluents have particular values

$$\mathbf{pow}(C, open_ballot(C, M)) = true \text{ holdsat } T \leftarrow$$
$$status(M) = seconded \text{ holdsat } T \wedge$$
$$role_of(C, chair) = true \text{ holdsat } T$$

The Status of Motions

- The exercise of power by performing an action changes the status of a motion
- The change of status 'switches on' and 'switches off' other powers

open_ballot(C, M) initiates $status(M) = voting(T)$ at $T \leftarrow$

pow($C, open_ballot(C, M)$) = *true* holdsat T

close_ballot(C, M) initiates $status(M) = voted$ at $T \leftarrow$

pow($C, close_ballot(C, M)$) = *true* holdsat T

declare(C, M, carried) initiates $status(M) = resolved$ at $T \leftarrow$

pow($C, declare(C, M, -)$) = *true* holdsat T

Roles & Role Assignment

- Agents 'qualified' to perform roles are assigned to roles by some dedicated role assignment protocol
- Other roles are the product of performing designated acts in the voting protocol

$propose(A, M)$ initiates $role_of(B, seconder) = true$ at $T \leftarrow$
 $\mathbf{pow}(A, propose(A, M)) = true$ holdsat $T \wedge$
 $qualifies(B, seconder) = true$ holdsat $T \wedge$
 $A \neq B$

Voting & Counting Votes

$open_ballot(C, M)$ initiates $votes(M) = (0, 0)$ at $T \leftarrow$

$\mathbf{pow}(C, open_ballot(C, M)) = true$ holdsat T

$open_ballot(C, M)$ initiates $voted(V, M) = nil$ at $T \leftarrow$

$\mathbf{pow}(C, open_ballot(C, M)) = true$ holdsat $T \wedge$

$role_of(V, voter) = true$ holdsat T

$vote(V, M, aye)$ initiates $votes(M) = (F1, A)$ at $T \leftarrow$

$\mathbf{pow}(V, vote(V, M)) = true$ holdsat $T \wedge$

$votes(M) = (F, A)$ holdsat $T \wedge$

$F1 = F + 1$

$vote(V, M, aye)$ initiates $voted(V, M) = aye$ at $T \leftarrow$

$\mathbf{pow}(V, vote(V, M, -)) = true$ holdsat T

Permission (and Sanction)

- 'Right' aspect of enfranchisement
 - Agents have the power to vote
 - Power implies permission
 - Power without permission initiates sanction: e.g. closing a ballot early

per($C, close_ballot(C, M)$) = *true* holdsat $T \leftarrow$

role_of($C, chair$) = *true* holdsat $T \wedge$

status(M) = *voting*(T') holdsat $T \wedge T > T' + 10$

close_ballot(C, M) initiates *sanction*(C) = [(103, M)| S] at $T \leftarrow$

role_of($C, chair$) = *true* holdsat $T \wedge$

per($C, close_ballot(C, M)$) = *false* holdsat $T \wedge$

sanction(C) = S holdsat T

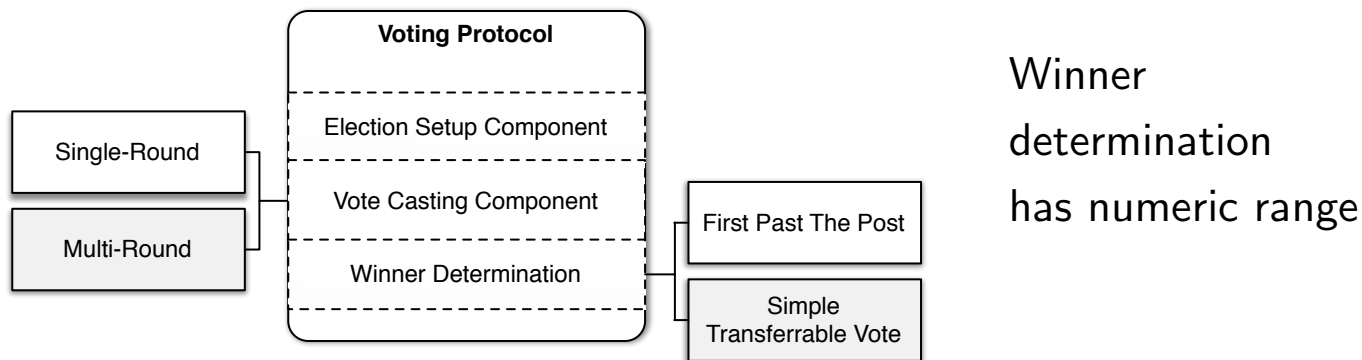
Obligation

- 'Entitlement' aspect of enfranchisement
 - 'Access' to 'voting machine' is a 'physical' issue
 - Correct vote count: as above
 - A 'fair' outcome: obligation to declare the result correctly: e.g. a simple majority vote

$$\begin{aligned} \text{obl}(C, \text{declare}(C, M, \text{carried})) = \text{true} \text{ holdsat } T \leftarrow \\ \text{role_of}(C, \text{chair}) = \text{true} \text{ holdsat } T \wedge \\ \text{status}(M) = \text{voted} \text{ holdsat } T \wedge \\ \text{votes}(M) = (F, A) \text{ holdsat } T \wedge \\ F > A \end{aligned}$$

Evaluating Adaptation

- Introduced modifiable aspects of an agent society in two forms:
 - **Parameters**, variables that specify some procedural characteristic of a norm creation rule (conceptual protocol).
 - **Replaceable Components**, sets of rules that represent the semantics of part or whole of a protocol and can be replaced at runtime by the society's agents.
- Add fluents and actions to allow modification within those constraints



Modelling Dynamic Specifications

- In the original ('static') framework, we prove properties of the system using narratives and 'animating' the specification
- In the extended framework, need to model the 'meta specification'
- We can consider a 'dynamic' specification as a sequence of 'static' specifications, which differ in the values of numeric parameters and replaced components
- By quantifying the parameter values and enumerating the protocol components we create an n-dimensional space determining all the degrees of freedom of the specification
- We can analyse the resultant specification space using the idea of Metric Spaces (Bryant)

Metric Spaces

- A Metric Space is a set with a notion of distance between members of the set
 - think euclidean 3-D space with distance given by straight line
 - think specification space where we define our own metrics for 'distance'
- A specification space with $r + q$ degrees of freedom is defined as:

$$X = \underbrace{\{0, 1, \dots, n_0\} \times \dots \times \{0, 1, \dots, n_{q-1}\}}_{q \text{ terms}} \times \underbrace{[a_0, b_0] \times \dots \times [a_r, b_{r-1}]}_{r \text{ terms}}$$

- The formal definition of a metric:

$$M(k, l) = \sum_{i=0}^{q-1} \frac{|l_i - k_i|}{n_i} + \sum_{i=0}^{r-1} w_i \frac{|l_i - k_i|}{|l_{max_i} - k_{max_i}|}$$

Example Metric Space: the Voting Protocol

- Given a numeric parameter (majority) and the replaceable winner determination component, and a second numeric parameter for timeout (between 3 and 100 time units)

$$X_{vp} = [3, 100] \times [0.5, \frac{2}{3}] \times \{0, 1\}$$

- Define a metric giving 'distance' between the 'points' in X_{vp}

$$d(a, b) = w_x \frac{|x_b - x_a|}{|100 - 3|} + w_y \frac{|y_b - y_a|}{|\frac{2}{3} - 0.50|} + w_z \frac{|z_b - z_a|}{1}$$

- The 'importance' of the shifts in all variables is given by setting the weights, w_x, w_y, w_z

Summary & Conclusions

- There are certain applications for which specification as an agent society is appropriate
- Specification can include object-level and meta-level components
- We require
 - a mechanism for how to adapt, e.g. voting
 - a mechanism for evaluation of adaptation, e.g. metric spaces
 - mechanisms for when to adapt (not covered here)
 - * e.g. machine learning, case-based reasoning
 - * 'fit for purpose': society and its norms

Further Reading

- J. Pitt, L. Kamara, M. Sergot and A. Artikis. Voting in Multi-Agent Systems, the Computer Journal, 49(2): 156-170, Oxford Journals, 2006.
- D. Kaponis and J. Pitt. Dynamic Specifications in Norm-Governed Open Computational Societies. Proceedings ESAW 2006, LNCS4457, Springer-Verlag, pp265-283, 2006.
- H. Roberts, W. Evans, D. Honemann and T. Balch. Robert's Rules of Order Newly Revised, Tenth Edition, Perseus Publishing, 2000.
- A. Artikis., M. Sergot and J. Pitt. An Executable Specification of a Formal Argumentation Protocol, Artificial Intelligence Journal, 171(10-15):776-804, 2007.