Negotiating Agents for Business Process Management

Nick Jennings
Dept of Electronics and Computer Science
University of Southampton.
nrj@ecs.soton.ac.uk
http://www.ecs.soton.ac.uk/~nrj
Talk Outline

I. The Basics of Business Process Management

II. The Case for an Agent-Based Approach to BPM

III. The ADEPT Approach

IV. Negotiation Techniques in ADEPT

V. Conclusions and Future Work
A Business Process

Market-centred description of an organisation’s activities, implemented as information processes (automated and partially automated tasks that create, process, manage, and provide information) and/or material processes (assembly of physical components or the delivery of physical products)

Medina-Mora et al.
Components of a Business Process
(from Workflow Management Coalition)

- **Business Process**
  - (what is intended to happen)
  - is defined in a process definition

- **Process Definition**
  - (representation of what should happen)

- **Activities**
  - Manual
  - Automated
  - may be composed of

- **BPM System**
  - (controls automated aspects of BP)
  - is managed by a BPM system
  - used to create & maintain

- **Process Instances**
  - (representation of what is actually happening)
  - include 1 or more activity instances

- **Activity Instances**
  - represented by work items
  - which include invoked apps

- **Work Items**
  - (tasks allocated to participants)

- **Invoked Apps**
  - (supporting computer tools)

During exec., may be used to create & maintain.
Managing Business Processes

♦ Successful organisations do it well!
  • activities completed on time, within specified constraints

♦ Increasing awareness of benefits of appropriate computer-based support in:
  • scheduling
  • executing
  • monitoring
  • coordinating

business process activities
Corporate-Wide Business Processes

♦ Process is dynamic and unpredictable
  • impossible to give complete \textit{a priori} specification of all activities to be performed and how they should be ordered.

♦ Physical distribution & high degree of concurrency

♦ Multiple organisations involved

♦ Within organisations:
  • decentralised ownership of tasks, information and resources
  • different groups are relatively autonomous
Traditional Workflow Approaches

- Describe entire business process from centralised perspective
  - need to provide:
    - complete list of all activities and all paths
    - criteria for following a particular path
    - ordering constraints on the actions
  - e.g: InConcert, Staffware, Floware, Action Workflow
Shortcomings of Traditional Methods

They lack:

- **reactivity**: require *a priori* representation of BP and all potential deviations;

- **context**: do not make decisions based on nature of information generated;

- **resource management**: do not control resourcing, rely on process being fully dimensioned beforehand;

- **heterogeneity**: take a centralised view with single workflow management engine
Talk Outline

I. The Basics of Business Process Management
II. The Case for an Agent-Based Approach to BPM
III. The ADEPT Approach
IV. Negotiation Techniques in ADEPT
V. Conclusions and Future Work
A New Perspective

Devolve responsibility for enacting specific business process activities to constituent components

♦ Components become active
  • activities assigned to number of problem solving entities
    ■ responsible for ensuring activity is fulfilled within specified constraints
    ■ means left to responsible entity to determine
Ramifications

More sophisticated management system

- decisions traditionally made in process description, moved to execution system:
  - which activities should be performed?
  - how much resource should each activity consume?
  - who should perform the activities?
  - when should the activities be performed?
  - how should the interdependencies be resolved?
An Agent-Based Approach

♦ Make each responsible entity an autonomous agent

♦ Such agents have specific goals to achieve and interact with one another to manage their interdependencies

♦ Group agents to reflect the desired organisational structure
Agent

“encapsulated computer system, situated in some environment, and capable of flexible autonomous action in that environment in order to meet its design objectives” (Wooldridge)

♦ control over internal state and over own behaviour

♦ experiences environment through sensors and acts through effectors

♦ reactive: respond in timely fashion to environmental change
♦ proactive: act in anticipation of future goals
Definitional Malaise

“My guess is that object-oriented programming will be what structured programming was in the 1970s. Everybody will be in favour of it. Every manufacturer will promote his product as supporting it. Every manager will pay lip service to it. Every programmer will practice it (differently). And no one will know just what it is.” (Rentsch, 82)

“My guess is that agent-based computing will be what object-oriented programming was in the 1980s. Everybody will be in favour of it. Every manufacturer will promote his product as supporting it. Every manager will pay lip service to it. Every programmer will practice it (differently). And no one will know just what it is.” (Jennings, 99)
Agent Interactions

♦ Interaction between agents is inevitable
  • to achieve individual objectives and to manage inter-dependencies

♦ Conceptualised as taking place at knowledge-level
  • which goals, at what time, by whom, what for

♦ Flexible run-time initiation and responses
  • cf. design-time, hard-wired nature of extant approaches

paradigm shift from previous perceptions of computational interaction
Organisations

Agents act/interact to achieve objectives:

- on behalf of individuals/companies
- part of a wider problem solving initiative

underlying organisational relationship between agents
Organisations

Since this organisational context:

• influences agents’ behaviour
  ■ relationships need to be made explicit
    – peers
    – teams, coalitions
    – authority relationships

• is subject to ongoing change
  ■ provide computational apparatus for creating, maintaining and disbanding structures
A Canonical View of Agent-Based Computing

Environment

Sphere of influence

Agent

Interaction

Organisational relationships

(see also: Castelfranchi, Ferber, Gasser, Lesser, …..)
Why Use an Agent-Based Approach?

♦ Inherent distribution of data, problem solving capabilities and responsibilities
  • conforms to basic model of distributed, encapsulated, problem solving entities

♦ Existing organisational structure must be maintained
  • appeals to autonomous nature of agents and explicit organisational representations
Why Use an Agent-Based Approach? (cont.)

♦ Interdependencies depend on context
  • requires complex social skills with which agents are endowed

♦ Solution cannot be prescribed from start to finish
  • be responsive to environmental changes and to unpredictability in BP
  • proactively take opportunities when they arise
I. The Basics of Business Process Management
II. The Case for an Agent-Based Approach to BPM
III. The ADEPT Approach
IV. Negotiation Techniques in ADEPT
V. Conclusions and Future Work
An Organisation as a Community of Negotiating, Service-Providing Agents

- Agency
- Responsible Agent
- Information Sharing
- Negotiation
- Services
- Service Level Agreements (SLAs)

- subsidiary of
- peer of
Services

♦ Activities managed by agents
  • conceptual unit of problem solving activity
  • what agents offer to their acquaintances (API)

♦ Service Lifecycle

<table>
<thead>
<tr>
<th>CREATION (manual)</th>
<th>PROVISIONING (automated)</th>
<th>MANAGEMENT (automated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Class</td>
<td>Service Instance</td>
<td>Manage constraints</td>
</tr>
<tr>
<td>SLA Class</td>
<td>SLA Instance</td>
<td></td>
</tr>
</tbody>
</table>

23
Service Creation

(service Basic Template
  name: Prepare_Table
  inputs:
  (Guests guests cli man
   TableAndChairs accom ser man
   Cutlery cutlery any man
   Crockery crockery any man
   Glasses glasses any opt)
  outputs (Seat_Allocation)
  guard "( <=guests.number accom.number)"
  body (unspecified> )
)

sequence:Meal( Service Body
  must-para:organise (cond:have_friends
    (not (empty-set service::friends))
    Plan_Menu (restrictions =
      service::friends),
  ) ..> (and have_friends Plan_Menu)
  can-para:prepare (Prep_Food (food =
    Plan_Menu::menu),
    Prep_Table (guests =
      Plan_GuestList::choice)
  ) ->(and Prep_Food Prep_Table),
  Eat_Meal (<unspecified>),
} ->(and organise prepare Eat_Meal)
Service Provisioning

(Faratin, Sierra and Jennings, 98)

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Cost</th>
<th>Start Time</th>
<th>End Time</th>
<th>Quality of Service</th>
<th>Penalty for Violation</th>
</tr>
</thead>
</table>

Does not know A2’s:
- reservation values
- resource constraints
- outside options
Why Negotiation?

♦ De facto means of coming to agreement between independent entities
  • cannot instruct an autonomous agent!

♦ Arrange inter-dependent activities (required services) according to prevailing situation:
  • short vs. long time to reach agreement
  • scarce vs. plentiful resource availability
Service Management

♦ Scheduling services
  • Associate task/service instance with particular SLA
    - Sequential provisioning: for on-demand services
      - All constituent components scheduled before first execution
      - Resources blocked off for duration of agreement
    - Look-ahead provisioning: for one-off services
      - As execute current step, provision next component

♦ Executing services
  • Parse service description and invoke activities according to specified logic
  • Monitor exceptions and try and deal with them:
    - by re-invoking different instances
    - by rescheduling activities within agreed SLA
    - by renegotiating SLA
ADEPT Architecture

Agency

Self & Acquaintance Models

- Service Execution
- Situation Assessment
- Interaction Management

Domain Tasks

Subsidiary Agents

Responsible Agent

Comms Module

Agency

Agency

Peers

Built on CORBA platform (DAIS)

CLIPS (for rule-based inference)

C (for integrating with DAIS)
Talk Outline

I. The Basics of Business Process Management

II. The Case for an Agent-Based Approach to BPM

III. The ADEPT Approach

IV. Negotiation Techniques in ADEPT

V. Conclusions and Future Work
BT’s Provide Customer Quote Process

(Jennings et al., 96)
Service-Oriented Negotiation

♦ Two agents must agree about conditions under which service will be executed

♦ Range of negotiation behaviour required:
  • competitive
  • accommodative
  • conciliatory

♦ Variety of contexts:
  • peer vs. subsidiary
  • external organisation vs. internal organisation
  • short time to reach agreement vs. plenty of time
Negotiation Characteristics

♦ Given service provided by multiple agents
  • Patterns: few--few, few--many, the difficult ones!
♦ Individual agents can be both clients and servers in different negotiations.
♦ Social context and inter-relationships influence negotiation behaviour
♦ Time is important:
  • time to reach an agreement
  • deadline by when negotiated service executed
The Negotiation Protocol

Alternating offers protocol

cand ⇔ capable ⇔ propose ⇔ counterpropose ⇔ accept

1 ⇔ 2 ⇔ 3 ⇔ 4 ⇔ 5 ⇔ 6 ⇔ 7

culuiice ciuicii ciuicii ciuiciciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciuiciu
The Negotiation Issues

Based on current BP agreements

• Intermediate SLA: “offer on the table”
• Agreed SLA: “binding Contract”

Issues to be settled:

• Delivery type: one-off vs. on-demand
• Start and end times
• Volume
• Price
• Penalty
The Reasoning Model

Negotiation Drivers

- Desirability
- Time until needed
- Num. Suppliers
- Resource consumed

High
Low

Evaluation Decision:
- Accept, Reject, Counter

Strategic Choice:
- balance drivers
- determine key issues
- monitor progress
- modify stance

Tactical Enactment

Service Name
- Cost
- Start Time
- End Time
- Quality of Service
- Penalty for Violation

Pool of Tactics
- Tit-for-Tat
- Conceder
- Boulware

Pool of
Tactics
The Evaluation Reasoner

♦ Rank incoming (counter) proposal using additive scoring function \( (V_{\text{received}}) \)
  • each SLA issue assigned importance weighting
  • function computes utility for value in slot
  • value of overall SLA = weighted sum of all issue

♦ Compute offer would have returned \( (V_{\text{wanted}}) \)
  • Negotiation no longer valid : Reject proposal
  • \( V_{\text{received}} \geq V_{\text{wanted}} \) : Accept proposal
  • \( V_{\text{received}} < V_{\text{wanted}} \) : Counter propose
Strategic Reasoner

Setting Strategies

If have-time & different-organisation(opponent)
then
strategy(time=0, resource=0.2, behaviour=0.8)
convergence_speed(slow)
likely_response(competitive)

If have-time & same-organisation(opponent) & many(servers)
then
strategy(time=0.5, resource=0.5, behaviour=0)
convergence_speed(fast)
likely_response(cooperative)

Modifying Strategies

If time-running-low & convergence_speed(slow) & different-organisation(opponent)
then
strategy(inc(time), dec(behaviour))
convergence_speed(quicker)
likely_response(unsure)

If have-time & likely_response(cooperative) & exploiting(opponent) & intensifying(exploitation)
then
strategy(dec(time), inc(behaviour))
convergence_speed(lower)
likely_response(competitive)
The Tactical Reasoner

♦ Families of tactics

• Time dependent
  - concede more rapidly as deadline approaches

• Resource dependent
  - model pressure of limited resources
    - number of potential providers
    - amount of communication resource consumed

• Imitative
  - protect agents from exploitation by basing behaviour on that of negotiation opponent
Sample Rules

if have-time &
   different-organisation(opponent) &
   many(servers) &
   strategy(?time,?resource,?behaviour)
then
   offer_time=high(initial), concede(slowly)
   offer_resource=high(initial),
   concede (number-servers)
   offer_behaviour=high(initial),
   concede(last-offer, less-than-opp)
   offer= average(
      (offer_time * ?time),
      (offer_resource * ?resource)
      (offer_behaviour* ?behaviour)
   )

if have-time &
   same-organisation(opponent) &
   one(server) &
   strategy(?time,?resource.?behaviour)
then
   offer_time=high(initial), concede(slowly)
   offer_resource=low(initial),
   concede(num-messages)
   offer_behaviour=medium(initial),
   concede(last-offer, more-than-opp)
   offer=average(
      (offer_time * ?time),
      (offer_resource * ?resource)
      (offer_behaviour* ?behaviour) )
Talk Outline

I. The Basics of Business Process Management
II. The Case for an Agent-Based Approach to BPM
III. The ADEPT Approach
IV. Negotiation Techniques in ADEPT
V. Conclusions and Future Work
Conclusions

♦ Promising means of conceptualising BPM apps
  • devolve responsibility to active problem solvers that interact to resolve dependencies

♦ Clean way of constructing agent applications
  • agents interact about services
  • negotiate to manage service interdependencies
  • agreements contain terms and conditions

♦ Flexible means of implementing negotiation
  • separation of strategies and tactics
  • heuristically-driven, context dependent selection/monitoring/ modification of negotiation behaviour
Future Work

♦ Support for organisational dynamics
  • creating and disbanding structures at run-time

♦ Greater flexibility in negotiation
  • mechanism design
  • argumentation
  • issue trade-off mechanisms