DIESECT: A Distributed Environment for Simulating E-commerce Contracts

Damian Wood, Özgür Kafalı and Kostas Stathis

Department of Computer Science
Royal Holloway, University of London, UK
{damian.wood.2010, ozgur.kafali, kostas.stathis}@cs.rhul.ac.uk

August 27, 2013
Outline

1. Introduction
2. Technical Background
3. DIESECT
4. Experiments
5. Conclusions
Electronic Contracts

- Contract: agreements describing the terms of how parties should act in their interactions
- Example delivery scenario:
  - buyer logs on to the online market to buy a book
Electronic Contracts

- Contract: agreements describing the terms of how parties should act in their interactions
- Example delivery scenario:
  - buyer logs on to the online market to buy a book
  - the terms of the contract describes how the buyer should pay as well as the deadline for delivery
Electronic Contracts

- Contract: agreements describing the terms of how parties should act in their interactions
- Example delivery scenario:
  - buyer logs on to the online market to buy a book
  - the terms of the contract describes how the buyer should pay as well as the deadline for delivery
  - contract not binding until buyer pays
Electronic Contracts

- Contract: agreements describing the terms of how parties should act in their interactions
- Example delivery scenario:
  - buyer logs on to the online market to buy a book
  - the terms of the contract describes how the buyer should pay as well as the deadline for delivery
  - contract not binding until buyer pays
  - fulfilled if seller delivers on time, violated otherwise
Electronic Contracts

- Contract: agreements describing the terms of how parties should act in their interactions
- Example delivery scenario:
  - buyer logs on to the online market to buy a book
  - the terms of the contract describes how the buyer should pay as well as the deadline for delivery
  - contract not binding until buyer pays
  - fulfilled if seller delivers on time, violated otherwise
- In open environments where agents collaborate to do e-business
  - contracts provide a basis to regulate agents’ actions
  - fulfillment of a contract depends on the participating agents’ behaviours
Motivation

- Execution and monitoring of e-contracts important to detect exceptions
- Several verification tools proposed in the literature
Motivation

- Execution and monitoring of e-contracts important to detect exceptions
- Several verification tools proposed in the literature
- Pros:
  - detect violations
  - diagnose cause of exceptions
  - predict future problems
Execution and monitoring of e-contracts important to detect exceptions
Several verification tools proposed in the literature
Pros:
- detect violations
- diagnose cause of exceptions
- predict future problems
Cons:
- mostly centralised, assume a monitoring entity
- ignore agent autonomy, work on a fixed trace of execution
Contributions

- We build upon previous work on agent contracts to provide an agent-based distributed environment *DIESECT*
  - integrate agent autonomy with contract execution in order to provide a simulation environment for electronic contracts
  - provide a practical implementation based on the widely-used JADE agent development framework
  - evaluate our system’s performance via experiments
- We combine the strengths of
  - object-oriented programming to provide the infrastructure and network operations for distribution and communication of agents
  - logic programming to provide a declarative and efficient way to deal with agent reasoning for contracts
Running Example

1. place order

contract [deliver in 3 days after payment]
Running Example

1. place order

2. send payment

contract [deliver in 3 days after payment]
Running Example

1. place order

contract [deliver in 3 days after payment]

2. send payment

Bank

3. notify payment

Seller

Buyer
Running Example

1. place order

contract [deliver in 3 days after payment]

2. send payment

3. notify payment

4. send delivery
Running Example

1. place order
2. send payment
3. notify payment
4. send delivery
5. deliver

contract [deliver in 3 days after payment]
JADE Agent Platform

  - provides reliable agent communication and documentation support
  - consists of a runtime environment, a library of classes to develop agents, and a set of graphical tools to allow agent administration and monitoring
  - use the FIPA [http://www.fipa.org/](http://www.fipa.org/) specification for agent communication via messages
  - provides a set of behaviours to describe agent tasks
  - provides a yellow page service for publish & subscribe type services
**JADE Agent Platform**

  - provides reliable agent communication and documentation support
  - consists of a runtime environment, a library of classes to develop agents, and a set of graphical tools to allow agent administration and monitoring
  - use the FIPA [http://www.fipa.org/](http://www.fipa.org/) specification for agent communication via messages
  - provides a set of behaviours to describe agent tasks
  - provides a yellow page service for publish & subscribe type services
- Provide a logic-based reasoning component for agents
- Implement agent contracts in JADE
Commitments

- Represents a contract between two agents
- $C_{\text{Debtor, Creditor}}^{\text{State}}$ (Antecedent, Consequent)
- $C_{\text{store, customer}}^{c}$ (paid, delivered)
- Changes state during the course of agents’ actions

![Diagram of Commitment States]

- $C^c(Q, P)$
  - Conditional
  - $C^a(Q, P), Q$
    - Active
    - $C^v(Q, P), Q, \neg P$
      - Violated
  - $C^f(Q, P), Q, P$
      - Fulfilled
Reactive Event Calculus (*REC*)

- Extends the Event Calculus to monitor commitments in run-time
- The reasoner component of the agent
- Uses three types of input:
  - Commitment theory
  - Protocol description
  - Event trace
% create as conditional
initiates (E, status(C, conditional), T):− ccreate(E, C, T).

% conditional to active
terminates (E, status(C, conditional), T):− detach(E, C, T).
initiates (E, status(C, active), T):− detach(E, C, T).

detach(E, c(X, Y, property(e(T1, T2), Q), P), T):−
  conditional(c(X, Y, property(e(T1, T2), Q), P), T),
  T >= T1, T <= T2, initiates(E, Q, T).

% payment
initiates (exec(pay(Customer, Bank, Item)), paid(Item), _).

% verification of payment
initiates (exec(verify(Bank, Customer, Item)), verified(Item), _).

% commitment for payment
create (exec(pay(Customer, Bank, Item)), Bank,
  c(Bank, Customer, property(e(Ts, Te), verified(Item))), Ts):−
  Te is Ts + 3.
Technical Background

REC Output

c(1, hsbc, federico, paid(card), delivered(card), rel(4, 7))

1. conditional detached

paid(card)

1. true

c(1, hsbc, federico, true, delivered(card), abs(8, 11))

1. active fulfilled

delivered(card)

1. true

exec(start) 1, exec(offer) 2, exec(accept) 3, exec(pay) 4, exec(hsbc) 5, exec(offer) 6, exec(pay) 7, exec(hsbc) 8, exec(offer) 9, exec(deliver) 10, exec(deliver) 11, exec(deliver) 12, exec(deliver) 13, exec(deliver) 14, exec(deliver) 15
Multi-agent Architecture

$Agent_i$  $Agent_j$
Multi-agent Architecture

Agent\textsubscript{i} and Agent\textsubscript{j} exhibit different behaviours.
Multi-agent Architecture

behaviours

Agent\textsubscript{i}

Agent\textsubscript{j}
Multi-agent Architecture

behaviours

Agent_i

Agent_j
Multi-agent Architecture

behaviours

Agent_i

Agent_j
Multi-agent Architecture
Multi-agent Architecture

Agent\_i

REC engine

behaviours

Agent\_j

JADE message
Multi-agent Architecture

Agent $i$ and Agent $j$ communicate via JADE message. The REC engine manages the behaviors of agents.
Simulation Profile

```xml
<simulation>
  <agents>
    <customer name="bob" eagerness="0.3" lateness="0.0">
      <wanteditems><product name="ipad"></product></wanteditems>
    </customer>
  </agents>

  <store name="ebay" eagerness="0.0" lateness="0.0">
    <inventory>
      <product name="ipad" deliveryCost="5" price="450"/>
      <product name="iphone" deliveryCost="5" price="350"/>
    </inventory>
  </store>

  <bank name="hsbc" eagerness="0.0" lateness="0.0"/>

  <courier name="ups" eagerness="0.0" lateness="0.2"/>
</agents>
</simulation>
```
DIESECT Interface

Actions

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Target(s)</th>
<th>Action</th>
<th>Start</th>
<th>Duration</th>
<th>ETA</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>bob</td>
<td>ebay, amazon</td>
<td>search(bob, ebay, amazon, [pad])</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>bob</td>
<td>ebay, paypal</td>
<td>buy(bob, ebay, [pad])</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>bob</td>
<td>ebay, order123456</td>
<td>verify(bob, ebay, order123456)</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>bob</td>
<td>ebay, order123456</td>
<td>verify(bob, ebay, order123456)</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>bob</td>
<td>ebay</td>
<td>search(bob, ebay, [pad])</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>bob</td>
<td>parforce</td>
<td>deliver(parforce, bob, order123456)</td>
<td>4</td>
<td>7</td>
<td>11</td>
<td>RUNNING</td>
</tr>
<tr>
<td>bob</td>
<td>ebay, cibay, bob, property(order123456)</td>
<td>deliver(cibay, bob, property(order123456, [pad]), order123456)</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>COMPLETE</td>
</tr>
</tbody>
</table>

Trace

0. T: bob; starting process of getting ipad
1. T: bob; scheduling search(bob, ebay, amazon, [pad]) to complete at 1
2. T: bob; performing search(bob, ebay, amazon, [pad])
3. T: bob; initiated a search of pad and returned product(order123456, [pad], $295.0000000)
4. T: bob; initiated a search of pad and returned product(order123456, [pad], $295.0000000)
5. T: bob; got offering of pad for $452.0000000 from amazon
6. T: bob; got offering of pad for $255.0000000 from ebay
7. T: bob; choosing to get ipad from ebay for $255.0000000
8. T: bob; performing buy(bob, ebay, [pad])
9. T: bob; created order order123456 for bob after buying ipad
10. T: bob; got invoice(order123456, [pad], ebay, bob, order123456, parforce) from abs
11. T: bob; performing payday(bob, hscb, order123456)
12. T: hscb; registered(hscb, order123456, verified(order123456))
13. T: hscb; scheduling verify(hscb, order123456) to complete at 4
14. T: hscb; scheduling verify(hscb, order123456) to complete at 4
15. T: hscb; registered(hscb, property(order123456), verified(order123456))
16. T: hscb; performing verify(hscb, order123456)
17. T: hscb; registered(cheby, bob, property(order123456, [pad], delivered(order123456)))
18. T: hscb; performing verify(order123456, hscb, order123456)
19. T: hscb; received notification that order order123456 was verified by hscb. order will be c
20. T: hscb; registered(cheby, bob, property(order123456, [pad], delivered(order123456)))
21. T: hscb; delivering(delivery(order123456, [pad], hscb, order123456))
22. T: parforce; order0123456 was dispatched by hscb, heading to bob
23. T: parforce; decided to exhibit lateness on deadlines of 5 1/2 making new deadlines
24. T: parforce; scheduling deliver(delivery(order0123456, [pad], hscb, order0123456)) to complete at 13
25. T: hscb; my prolog REC engine says status of cibay, bob, property(order123456, [pad], verified)
26. T: hscb; my prolog REC engine says status of cibay, bob, property(order123456, [pad], delivered)
27. T: bob; my prolog REC engine says status of cibay, bob, property(order123456, [pad], delivered)
28. T: bob; performing diagnostics(bob, ebay, cibay, bob, property(order123456, [pad], delivered)
29. T: bob; requesting diagnosis from ebay for cibay, bob, property(order123456, [pad], delivered)
30. T: bob; my prolog REC engine says status of cibay, bob, property(order123456, [pad], delivered)
31. T: bob; got diagnosis required from bob for cibay, bob, property(order123456, [pad], delivered)
32. T: bob; got diagnosis from ebay as MSBKH14AVOUR
33. T: bob; culprit was ebay, adding to misbehaving agent set so as to avoid in future
Sequence Diagrams in DIESECT

- T=1 search(bob, [amazon, ebay, ipad])
- T=1 search(bob, [amazon, ebay, ipad])
- T=1 buy(bob, ebay, ipad)
- T=1 pay(bob, hsbc, order22690)
- T=1 verify(hsbc, bob, order22690)
- T=4 deliver(parcelforce, bob, order22690)
- T=10 diagnose(bob, ebay, c(ebay, bob, property(e(4,9), delivered(order22690))))
- T=1 verify(fysseller(hsbc, ebay, order22690))
- T=4 dispatch(ebay, parcelforce, order22690)
JADE Sniffer Agent
Experimental Setup

- **Number of agents**
  - gradually increase the number of agents
  - 10 to 140 buyers
  - two sellers, one bank and one courier
  - see whether the results are consistent with the performance of REC reported before (with no additional simulation overhead)

- **Different agent behaviours**
  - 30 buyers, two sellers, one bank and one courier
  - change the simulation profile by assigning different behaviours to agents
  - see whether behaviour affects performance
Performance Results: Number of Agents

- Linear increase in memory
- Time requirements consistent with REC
## Performance Results: Agent Behaviours

<table>
<thead>
<tr>
<th>Agent behaviour</th>
<th>Time (s)</th>
<th>Memory (mb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer eagerness 100%</td>
<td>5.62</td>
<td>218</td>
</tr>
<tr>
<td>Customer eagerness 50%</td>
<td>4.63</td>
<td>207</td>
</tr>
<tr>
<td>Bank lateness 100%</td>
<td>3.87</td>
<td>208</td>
</tr>
<tr>
<td>Bank lateness 50%</td>
<td>4.02</td>
<td>211</td>
</tr>
<tr>
<td>Store lateness 100%</td>
<td>3.41</td>
<td>217</td>
</tr>
<tr>
<td>Store lateness 50%</td>
<td>4.30</td>
<td>213</td>
</tr>
<tr>
<td>Courier lateness 100%</td>
<td>3.98</td>
<td>208</td>
</tr>
<tr>
<td>Courier lateness 50%</td>
<td>4.27</td>
<td>210</td>
</tr>
</tbody>
</table>
Conclusions

- Presented **DIESECT**: a distributed simulation environment for contract execution and monitoring
- Autonomous behaviour of agents may lead to different contract outcomes during execution
- Shown that our system performs well with a reasonable number of agents
Conclusions

- Presented **DIESECT**: a distributed simulation environment for contract execution and monitoring
- Autonomous behaviour of agents may lead to different contract outcomes during execution
- Shown that our system performs well with a reasonable number of agents
- Future work:
  - Generic contract execution where protocols described by defining commitment templates and associated agent roles
  - Integrate model checking capability to enable verification and prediction
  - Extend performance evaluation to distinguish between the time spent on JADE and REC