

Web Services Negotiation in an Insurance Grid

Shamimabi Paurobally¹, Chris van Aart², Valentina Tamma¹, Michael Wooldridge¹,
Peter van Hapert³

¹Department of Computer Science University of Liverpool Liverpool L69 3BX, U.K.

²Y'All B.V., Grotestraat 182, 5141 HD Waalwijk, The Netherlands

³Boyd International, B.V., Stationsstraat 11, Tilburg, NL-5000AE, The Netherlands

¹{sha,valli,mjw}@csc.liv.ac.uk, ²chris@y-all.nl, ³pvanhapert@boydinternational.nl

ABSTRACT

There are an increasing number of initiatives for the migration of agents research towards new Internet technologies such as the semantic web, Grid, and web services. On the one hand, service oriented Grid architectures need to support dynamic cooperation, negotiation and coordination between web services controlling Grid resources, for efficient resource and task allocation and execution. On the other hand, the Grid can facilitate agent communication, life-cycle management, and access to resources for agents. The insurance sector is one such area that would benefit from the automation brought by multi-agent systems techniques in handling claims and detecting fraudulent cases. In this paper, we discuss our work on facilitating dynamic and adaptive negotiation between web and grid services. We describe our deployed approach in an InsuranceGrid, which manages businesses involved in dealing with car damage claims for a number of insurance companies.

General Terms

Experimentation

Keywords

web services, insurance grid, negotiation

1. INTRODUCTION

The long-term Grid vision involves the development of “large-scale open distributed systems, capable of effectively and dynamically deploying and redeploying computational (and other) resources as required, to solve computationally complex problems” [5]. But the Grid community is by no means the only research community with such a set of goals: research in multi-agent systems addresses a very similar set of issues [4]. Thus, research in the architecture of the Grid has focussed largely (though by no means exclusively) on the development of a software middleware with which complex distributed systems, (often characterised by large datasets

and heavy processing requirements) can be engineered. On the other hand, research in multi-agent systems has addressed the issue of how independently acting (often called “autonomous”) computing elements (termed agents) can cooperate to solve complex problems that are beyond the capabilities of any of the individual computing elements. In fact, current web services would benefit from existing research in negotiation between parties with heterogeneous information needs. In Grid systems, the effective allocation of limited resources in an open environment to satisfy customers’ requirements can be complex. Thus, a key challenge for the Grid in the coming decade is adaptability to varying requirements, and availability of distributed resources. Resource management systems are moving from localised resources and services towards virtual organisations (VOs) sharing heterogeneous resources across multiple organisations and domains [9], [1]. The virtual organisations and usage models include a variety of owners and consumers with different usage, access policies, cost models, varying loads, requirements and availability. In current Grid applications, heterogeneity and dynamic provisioning are limited, and dynamic virtual organisations are restricted to those parties with a priori agreements to common policies and practice.

Another area with heterogeneous parties that would benefit from agent negotiation is the insurance sector. Currently, the insurance market mostly relies on traditional ways of handling claims, which can be slow and costly because of the inter-dependency between the multiple parties involved in expediting a claim. All involved parties exchange large amounts of data and maintain long-term relationships. The exchange of information is time consuming and is subject to errors. The reason why this cross-process and cross-company data exchange is not yet automated is because human nodes make the necessary translation from one domain (e.g. from a financial to legal domain) to another. There have been previous work on agents in the insurance field [11]. Thus, automating the various steps in claims handling can help save costs and time.

To this end, we propose to facilitate the automation of a number of the steps in insurance claim settlement, in the *InsuranceGrid*. An InsuranceGrid looks after and controls all businesses involved in dealing with car damage claims for a number of insurance companies. The goal of the InsuranceGrid is to enhance the quality and efficiency of the total damage claims handling process between consumer, damage repair companies and insurance companies. In this paper, we integrate an agent negotiation and a web services approach towards automating some of the steps in insurance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS’07 May 14–18 2007, Honolulu, Hawaii, USA.

Copyright 2007 IFAAMAS.

claims settlement. More particularly, we deploy negotiation mechanisms in the InsuranceGrid business case. Such an approach integrates chains of services, exchanges substantial data via secure, reliable and robust channels, and encourages interactions between insurance companies which would otherwise not have trusted each other. The various actors are considered as being part of a grid and offering grid or web services. Our aim is to successfully deploy negotiation and interaction mechanisms between an InsuranceGrid services in the same way that autonomous agents negotiate in a collaborative or competitive distributed system.

The remainder of this paper is structured as follows. Section 2 provides an overview of the need for negotiation and web services negotiation in the insurance domain and an overview of the InsuranceGrid. Section 3 describes two negotiation protocols and their strategies that we deploy in the InsuranceGrid, namely the contract net protocol (CNP) and the English auction with deadlines. Section 4 gives more technical details about the deployment of negotiation in the InsuranceGrid in Tomcat and GT4. Section 5 provides an evaluation of our approach, and section 6 concludes.

2. THE INSURANCE BUSINESS CASE

The Insurance field largely relies on very traditional procedures for handling claims. Every aspect of a claim will often be dealt with by a different specialized person working in a different department of an insurance company. The input, processing and distribution of data is treated by each component in their own traditional way. As a consequence, claims processing is very costly, and the insurance market is increasingly looking for ways to reduce the cost of claims handling. Because the process of claims handling involves many different parties (such as the victim(s), witnesses, surveyors/experts, lawyers, insurance companies, and doctors), there is a growing need for chain integration. More information about the insurance field can be found in the International Insurance Factbook:

<http://www.internationalinsurance.org/>.

2.1 The InsuranceGrid

An InsuranceGrid (see also www.insurancegrid.org) is built for an imaginary company called DamageSecure, which manages businesses involved in car damage claims on behalf of insurance companies. The goal of DamageSecure is to enhance the quality and efficiency of the total damage claims handling process between consumer, damage repair companies and insurance companies. Every year around 100,000 damages are reported to DamageSecure, of which 40% are repairs and 60% replacements. If an InsuranceGrid can work without human intervention, it could potentially save 172M euros.

An InsuranceGrid reasons about repairs of damaged cars which are insured at insurance companies. The various parties in the car repair claims handling have different preferences. When a car is damaged a customer wants to get its car repaired as soon as possible. An insurance company needs to pay for the damages (claim settlement) under the best circumstances (lowest prices, highest quality, as soon as possible, as close as possible, etc). Currently, insurance companies have long term contracts (e.g. over a year) which are negotiated manually with customers and repair services. Our aim is have repair services representing repair compa-

nies negotiate for repairs. This has two advantages: (1) it is potentially more efficient than the current claim settlement process, which is manual, and (2) repair prices from repair companies will drop and the quality of repairs will increase.

2.2 Repair Grid Scenario

We illustrate the interactions when handling claims for car repairs. In figure 1, a customer is insured at a company offering insurance services. Car repair services are carried out by the damage repair company and a surveyor may provide expert advice services on the quality and cost of repairs. DamageSecure is a company that provides services for liaising between insurance settlement services.

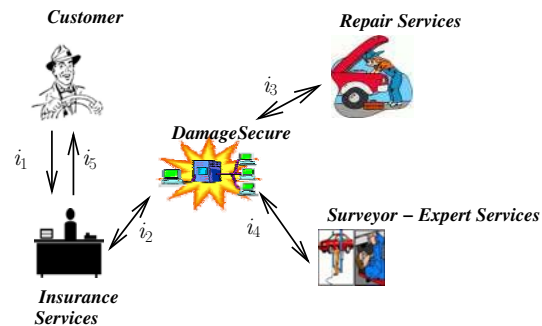


Figure 1: Members and Interactions in the InsuranceGrid VO

Before any repair claims are received, the repair service can negotiate a contract with the DamageSecure service and insurance companies for bidding on repair jobs from insured customers. The terms of a contract may include price of material and labor, speed and quality of expediting repair jobs. All parties to the contract can re-negotiate their contracts if they are not satisfied with the repairs or payments. Customers buy services from insurance companies which negotiate with the DamageSecure service on the best price and quality for their insured customers. After an accident, the following interactions to handle a claim occur between the insurance settlement services:

- i_1 : A customer files a damage report with an insurance company.
- i_2 : Based on a description of the damage, the insurance service requests DamageSecure to select the most appropriate repair service representing a repair company that will physically repair the car.
- i_3 : Before any negotiation is carried out, DamageSecure selects a number of repair services based on the competences of the repair companies they represent. For example, if the brand of the damaged car is Volvo, then only repair companies skilled in repairing Volvos are selected for participation in the negotiation. DamageSecure ask the selected repair services to provide an offer. The repair services analyse the damages and make a proposal to DamageSecure, who decides which proposal to accept or reject. This step may resemble to a contract net protocol (CNP) [6], where DamageSecure makes a call for proposal (cfp) to repair services,

who responds with proposals, out of which DamageSecure selects the best proposal.

- i_4 : DamageSecure can employ the services of an expert surveyor to analyse the quality of repairs on the damages and the charged costs. If a party is not satisfied, the contract between DamageSecure and repair services may be revised.
- i_5 : The insurance service pays the repair company, determines the liability and process the claim accordingly.

On analysing the above scenario, it can be seen that negotiation arises at various points between the services. Before any specific accident and claim is handled, there are negotiations to draw up long-term contracts between the insurance company and DamageSecure and, between DamageSecure and repair services. During the claims process itself, negotiation takes place between DamageSecure and the repair services.

3. WEB SERVICES NEGOTIATION

Owning and using grid resources may be expensive and resource owners would prefer to charge for resources instead of the free sharing and provision of resources. There are various forms of automated negotiation possible between web services for task and resource allocation and sharing. One such negotiation for task allocation can be found in the InsuranceGrid for finding the best repair service. A consumer, such as DamageSecure, has requirements that need to be fulfilled and is trading for a task to be executed, or for resources so as to perform a task. A service provider, a repair service here, has advertised its service and is ready to meet requests that comply to the service interface. However, the service provider retains control of how much service information is exposed to the client.

Reconciling the consumer's and provider's preferences and constraints can be carried out through negotiating SLAs. Agreements on resource provisioning may not only include the provider's commitment to execute a task or provide the resources but also include terms about performance levels and penalties. Creating agreements of this type is called provisioning. Some specific examples of where negotiation may be used in drawing up SLAs for resources, in addition to our insurance business case, include resource management about provision capability to perform some task for consumer, advanced reservation, on-demand access, scheduling and coordinating on-demand agreements, and policy based dynamic negotiations for Grid services authorisation.

3.1 CNP between Web Services

The first protocol we implemented between web services is the Contract Net protocol (CNP). The protocol occurs between a manager (DamageSecure) and a number of contractors (Repair services). The manager has a task to be achieved, and aims to outsource this task to contractors. First an agreement between the manager and contractors is arrived upon through the CNP. Then the contractor informs the manager of the result of executing the task or executing the proposal. Our CNP service is generic and can be applied to deploy resource and task allocation negotiations between web services, not only in the insurance business case but also in other web services applications. The contractor and

manager are services, and each exposes their interface. For example, the contractor service publishes a `cfp` (call for proposal) method as part of its interface. A manager service invokes the contractor's `cfp` method. The contractor's implementation of the `cfp` method will enable it to evaluate and choose the appropriate response to an invocation of its method.

Thus, the contractor and manager services expose and implement the methods: `cfp`, `accept`, `reject`, `propose`, `refuse`, `result`. The API is exposed by a CNP service as a WSDL file. Figure 2 below lists the porttype of the CNP service.

```
<wsdl:portType name="CNP">
  <wsdl:operation name="do_Negotiation"
    parameterOrder="Context_Job contractor_list">
    <wsdl:input message="impl:do_NegotiationRequest"
      name="do_NegotiationRequest"/>
    <wsdl:output message="impl:do_NegotiationResponse"
      name="do_NegotiationResponse"/>
  </wsdl:operation>
  <wsdl:operation name="get_result" parameterOrder="ContextJob">
    <wsdl:input message="impl:get_resultIn" name="get_resultIn"/>
    <wsdl:output message="impl:get_resultOut" name="get_resultOut"/>
  </wsdl:operation>

  <wsdl:operation name="cfp" parameterOrder="M cfp_1">
    <wsdl:input message="impl:cfpIn" name="cfpIn"/>
    <wsdl:output message="impl:cfpOut" name="cfpOut"/>
  </wsdl:operation>

  <wsdl:operation name="propose" parameterOrder="C proposal">
    <wsdl:input message="impl:proposeIn" name="proposeIn"/>
    <wsdl:output message="impl:proposeOut" name="proposeOut"/>
  </wsdl:operation>

  <wsdl:operation name="acknowledge" parameterOrder="C proposal">
    <wsdl:input message="impl:ackIn" name="ackIn"/>
    <wsdl:output message="impl:ackOut" name="ackOut"/>
  </wsdl:operation>

  <wsdl:operation name="reject" parameterOrder="M proposal">
    <wsdl:input message="impl:rejectIn" name="rejectIn"/>
    <wsdl:output message="impl:rejectOut" name="rejectOut"/>
  </wsdl:operation>

  <wsdl:operation name="refuse" parameterOrder="C cfp_1">
    <wsdl:input message="impl:refuseIn" name="refuseIn"/>
    <wsdl:output message="impl:refuseOut" name="refuseOut"/>
  </wsdl:operation>

  <wsdl:operation name="accept" parameterOrder="M proposal">
    <wsdl:input message="impl:acceptIn" name="acceptIn"/>
    <wsdl:output message="impl:acceptOut" name="acceptOut"/>
  </wsdl:operation>
</wsdl:portType>
```

Figure 2: CNP Service PortType

3.2 English Auction between Web Services

Figure 3 illustrates the English auction as the second negotiation protocol that we have implemented between web services. An auctioneer web service informs bidder web services about the start of the auction and the issues on which to bid – for example, price of repair, speed of repair and location of repair company. A bidder service replies with a bid on the indicated issues. The auctioneer evaluates the bids, chooses the best bid, `highest_bid`, and in-

votes the `request_bid` method on the bidder services with `highest_bid` as parameter. By doing so, the auctioneer is invoking the bidder services to submit bids again respective to `highest_bid`. The bidder services evaluate whether they can submit higher bids and if so send their revised bids. The auctioneer again evaluates received bids and requests higher bids from the remaining bidders with respect to the new highest bid. This process of requesting new bids and submitting higher bids continues until the auction deadline or there is only one bidder left. At these terminating conditions, the auctioneer invokes the `result` method on the bidder services indicating the end of the auction and informing them of the winning bid. Figure 4 gives the porttype of this English auction.

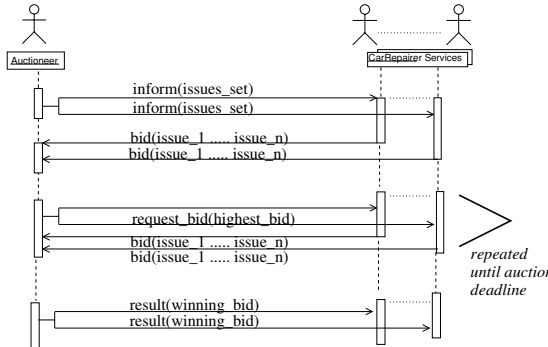


Figure 3: English Auction in the InsuranceGrid

```

<wsdl:portType name="Bidder">
  <wsdl:operation name="chooseStrategy"
    parameterOrder="strategy deadline">
    <wsdl:input message="chooseStrategyIn"
      name="chooseStrategyIn"/>
    <wsdl:output message="chooseStrategyOut"
      name="chooseStrategyOut"/>
  </wsdl:operation>

  <wsdl:operation name="SubmitBid" parameterOrder="bid">
    <wsdl:input message="SubmitBidIn" name="SubmitBidIn"/>
    <wsdl:output message="SubmitBidOut" name="SubmitBidOut"/>
  </wsdl:operation>

  <wsdl:operation name="Result" parameterOrder="bid">
    <wsdl:input message="ResultIn" name="ResultIn"/>
    <wsdl:output message="ResultOut" name="ResultOut"/>
  </wsdl:operation>

  <wsdl:operation name="getBidderID">
    <wsdl:input message="getBidderIDIn" name="getBidderIDIn"/>
    <wsdl:output message="getBidderIDOut" name="getBidderIDOut"/>
  </wsdl:operation>

  <wsdl:operation name="getMember">
    <wsdl:input message="getMemberIn" name="getMemberIn"/>
    <wsdl:output message="getMemberOut" name="getMemberOut"/>
  </wsdl:operation>
</wsdl:portType>
  
```

Figure 4: English Auction Service PortType

In the InsuranceGrid, the English Auction can be used

instead of the CNP for negotiation between DamageSecure and RepairServices. This occurs during interaction i_3 in figure 1. As with the CNP, DamageSecure selects the repair companies which are skilled in repairing the damages detailed in the damage report. Here DamageSecure and repair services have the English Auction deployed so as to negotiate according to this protocol. DamageSecure acts as the auctioneer and informs the repair services about the start of the English auction by invoking the `inform` method on the selected repair services naming the issues on which the auction will be conducted. Each repair service generates a value for each issue according to their preferences and their strategy and responds to DamageSecure with a bid. DamageSecure evaluates the bids according to its own preferences, chooses the best bid, then invokes the `request_bid` method on the repair services with the highest bid as parameter. The repair services decide whether to continue bidding, and if so generate their bid and respond to DamageSecure with their new bid. DamageSecure has deadlines for both the overall auction and for each round of bidding.

3.3 Strategies for CNP and English Auction

In the CNP negotiation, there are several stages where either the repair services or DamageSecure have to make a decision according to the negotiation subject and their own preferences. 1) DamageSecure has to generate a cfp, 2) a repair service evaluates the cfp, 3) a repair service generates a proposal, 4) DamageSecure evaluates the proposals, and 4) DamageSecure generates an acceptance. Similarly in the English auction, DamageSecure as the auctioneer has to evaluate the bids and choose the highest bid with respect to the issues of the repair job. On the other hand, on receiving a request to bid from DamageSecure, repair services have evaluated whether they can continue to participate in the auction given the current highest bid and if so, then they have to generate a bid by assigning a value to each issue of the repair job. DamageSecure and the repair services in both negotiation protocols have personal preferences for each issue in the negotiation subject. These preferences are the preferred value, the reserve value, the weight of the issue and the utility of the issue. For example, for the issue price, DamageSecure would have a preferred value of £50, a maximum value of £100, and 0.7 as the normalised weight for this issue to define its importance relative to other issues.

We implemented three strategies for evaluating and generating bids and proposals: truth-telling strategy, constant mark-up, and time dependent.

3.3.1 Truth Telling Strategy

In the CNP, when generating a cfp, DamageSecure (the manager) constructs the cfp with its preferred value for each issue. On receiving a cfp, a repair service checks whether the cfp is within its reserve values for each issue, and if so, the repair service as a contractor replies with a proposal where each issue is given the repair service's own preferences as value. If the cfp lies outside the reserve values for negotiable issues, then the repair service's proposal is instantiated with the repair service's reserve values. If the cfp lies outside the reserve values for non-negotiable issues, then the repair service replies with a refusal. DamageSecure evaluates the proposals and accepts the best proposal ensuring that all the values in the accepted proposal lie within its preferences.

The English auction truth telling strategy resembles the CNP truth telling strategy. DamageSecure informs the start of the auction but does not ground the issues with values as with a cfp. The bidders assign values for each issue in the bids first with their preferred value and finally with the lower end of their reserve value, depending if they are still participating in the auction, that is if the highest bid is within their reserve value.

3.3.2 Constant Markup/down Strategy

When its `cfp` or `request_bid` method is invoked, a repair service evaluates the cfp or the highest bid to respectively decide whether it can submit a proposal or whether it can continue to participate in the auction. The evaluation function will return `true` if it is calculated that the cfp or the highest bid lies within a margin of the repair service's preferences. Thus, even if the cfp or the highest bid is outside the repair service's preferences but within a defined margin, subsequent negotiation can bring the proposal or the bid within the preferences. Otherwise, if the evaluation function returns `false`, meaning that the cfp or the highest bid is outside the repair service's margin of preferences, then the repair service will invoke the `refuse` method on DamageSecure, or will stop participating in the auction.

When generating a proposal or a bid, a repair service considers its preferences and the cfp or the current highest bid if past the first round of the auction. The generated value of an issue in a proposal or a bid is a function of the value in the cfp or the highest bid and of the preferred value in a repair service's preferences. For any issue, the generated value may happen to lie outside the repair service's reserve value because the value in the cfp or the highest bid lies significantly outside the reserve value. In this case, the repair service assigns its reserve value for that issue. For example, a repair service wants to be paid a higher price than what is in the cfp, which is denoted by `cfp_price`. The price to be proposed by the repair service is then calculated from the repair service's preferred price and `cfp_price`. However if this calculated price is lower than the repair service's minimum `reserve_value` for price, the proposal price is assigned the contractor's reserve value.

DamageSecure evaluates received proposals or bids to decide which proposal to accept or which bid to declare as the highest bid. For the CNP, a proposal has to fall within DamageSecure's preferences because there is not another round for counter-proposals. In both the English auction and the CNP, DamageSecure computes the utility of all the received proposals and chooses to accept or to declare as the highest bid the proposal with the highest utility.

3.3.3 Time Dependent Strategy

Both the CNP and the English auction are dependent on the time remaining until the end of negotiation. The CNP deadline relates to DamageSecure's deadline for receiving proposals. In the English auction, there is a deadline for receiving bids for each round and an overall deadline for the auction for the auctioneer. A bidder only has its personal deadline for the overall auction. As the auctioneer, DamageSecure assigns a value to each bid as a function of the valuation for each issue and their relative importance of the issues forming the bid. Evaluation of a bid involves summing the valuation (score) of each issue in the bid. Let V_j^a be the evaluation of agent a about an issue j . Let ω_j^a de-

note the weight agent a associates to issue j . Then, a bid b , consisting of issues j , is simply evaluated by agent a as:

$$V^a(b) = \sum_{1 \leq j \leq n} \omega_j^a V_j^a(b[j]) \quad (1)$$

A repair service evaluates the call for bids using the above function and computes its own bids according to its deadline. The closer the current time is to a bidder's deadline, the greater the amount that the bidder concedes in generating its bid. The bid is computed as a function of the proportion of the remaining time over the total time allocated to the auction by that bidder. The bid $bid_b^t[j]$ of bidder service b at time t with deadline t_{max} is calculated in equation 2 below for each issue j with max_j and min_j as preferred and reserve values for that issue.

$$bid_b^t[j] = min_j + \frac{min(t, t_{max})}{t_{max}}(max_j - min_j) \quad (2)$$

This is for a decreasing issue where the bidder prefers a high value for that issue and has a minimum value as reserve. For an increasing issue where the bidder prefers a low value for that issue, the above equation 2 is changed to $min_j + (1 - x)$ where x is the second part of the addition in equation 2.

4. DEPLOYING NEGOTIATION IN THE INSURANCEGRID

We implement two actual deployments of the English auction and the CNP service in Tomcat [2] and in Globus Toolkit 4 (GT4) [8] respectively. We mention both deployments since both platforms are popular for service deployment.

4.1 Storing Contracts and Preferences

The issues that we consider for negotiation are:

- *Price*. This is a function of the labour costs, repair costs and profit margin calculated on the repair service side. On DamageSecure's side, price is what it is willing to pay for the repair job.
- *Location*. This is distance of the car repairer from a particular location.
- *Type of material1*. The type can be original, copy or second-hand, where copy is new, but not from the original car builder. There is the option of replacing car parts with original or second-hand parts.
- *Speed of repair*. This is how soon can the repair company perform the repairs.
- *Appointment date*. This is the date to leave or collect the car.
- *Method of repair (repair or replace)*. After analysing the damages, a repair service may propose to either repair or replace certain car parts.
- *Type of paint (metallic or not)*. For some cars, metallic paint may be needed and this would cost more and in addition to the normal car painting.

- *Quality of repair.* This is a qualitative attribute and can be defined by comparing with previous repairs or using expert services on car repairing.
- *Colour.* This is the colour of the car and is normally non-negotiable.

Various parties in the insurance business case have private data-bases in which damage reports, negotiated contracts and their private preferences are stored [7]. The databases in the insurance scenario are queried at various points before and during the negotiations to appropriately evaluate and generate cfps, proposals, bids, acceptances and rejections.

These databases are stored and queried using an ontology service, called WS-DAIOnt service [3]. This is where the semantic aspect of our work lies. The WS-DAIOnt service defines a framework for creating ontology access services in a Grid environment. WS-DAIOnt uses the standard grid data access vocabulary, and extends the data access mechanisms with the patterns, properties and behaviours needed for providing ontology access. The WS-DAIOnt specification and the accompanying realizations define the data access services that are needed for dealing with ontologies in Grid environments [3]. Insurance companies have private backoffice databases for containing details of insured customers and damage reports that need to be processed. DamageSecure has a private database (also backoffice) for storing previously drawn up contracts between insurance companies and registered repair companies. A car repairer itself holds a database that contains the contracts between itself and insurance companies.

An insurance service selects a damage report from its internal databases and delegates the repair to DamageSecure. Given a damage report, based on the type of damage and contract between repair and insurance companies, DamageSecure will select a number of repair services from the “Repair Company” Database. Selected repair services already have a contract with the insurance company and will be sent a *cfp* or *inform* to participate in the CNP or auction. In the middle of the negotiation, when evaluating the cfp or bid and calculating what proposal or bid to send, a repair service will analyze the damages in detail and make a calculation of the repairs or replacements costs by using their “parts” internal databases. There is also a public database that stores repair costs of various car parts and this database is used by DamageSecure to evaluate and generate the values in *cfp* and proposals. There is another public database about registered cars which is accessed by repair services and DamageSecure to get details about cars. There is a Motor Vehicle Database which is a public database that contains information about known cars characterised by brand, model, type and newprice.

4.2 Deploying CNP service in GT4

Figure 5 shows the deployment of the CNP in globus toolkit 4. Web services are usually stateless, that is there is no state that persists from one invocation to another. To add statefulness to a web service, a resource class is created to store all state information. Each resource has a unique key to enable stateful interactions with that web service and to use that particular resource. GT4 is an open source toolkit for building Grid-based applications.

The resource implementation **Negotiation Resource** contains the resource properties and the resource key as private

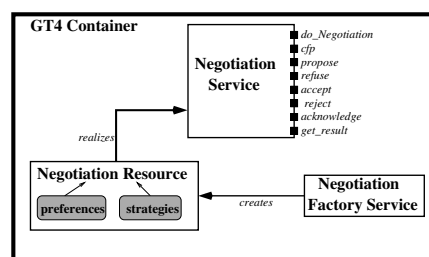


Figure 5: GT4 Deployment of the CNP Service

attributes and accessor methods for each resource property. The resource property for the CNP are *JobName* as an identifier of a particular negotiation, and the negotiation protocol class which can access the preferences and strategies of the CNP service. The resource class implements a constructor method for initialising the resource properties and adding them to a common resource property set.

We implement and deploy a second web service, called the CNP_Factory service, to deal with multiple resources. The factory service provides clients with an interface to request the creation of new WS-Resources and returns an endpoint reference (EPR) to a new WS-Resource. Thus, creation of a new resource requires retrieving the resource home, using it to create a new resource and creating the EPR which is composed of the new resource key and the CNP service URI.

4.3 Deploying English Auction in Tomcat

Figure 6 shows our deployment of the English auction in Apache Axis and Tomcat. Apache Axis is an implementation of the SOAP (“Simple Object Access Protocol”) submission to W3C. Apache Tomcat is a JAVA servlet container that can support web services Requirements. The Web Services Deployment Description (WSD) file is used to deploy the English auction service. This is an XML configuration file specifying details of the English auction service, such as the namespaces, the service name, the qualified name for the binding and for the operations, to be made available to the Axis engine. We use Apache Axis’s *WSDL2Java* utility to create client stubs and server-side skeleton code from the English auction service WSDL specification. While the English auction is deployed in one Tomcat container for each bidder, the private data is itself distributed over several databases as shown in figure 6.

Figure 7 shows the front end of the InsuranceGrid. Figure 8 shows the deployed InsuranceGrid and a trace of the execution of the CNP as inspected by the business process monitoring tool [10].

5. EVALUATING NEGOTIATION IN THE INSURANCEGRID

Figure 9 shows the test cases, that is the damage reports in the InsuranceGrid. The negotiation between DamageSecure and repair services is about reaching an agreement for repairing the damages detailed in the damage reports. We evaluate the bids and agreements in the InsuranceGrid, and the relative value of the negotiation protocols and of the strategies. Here we run 10 repair services as bidders competing to obtain the repair job with DamageSecure. Figure

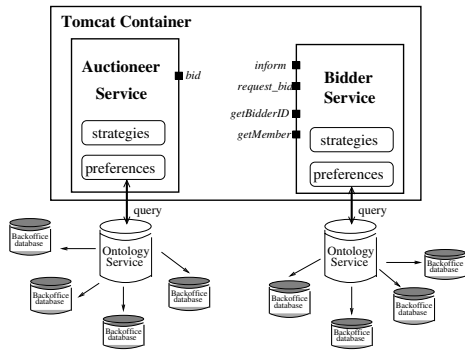


Figure 6: Tomcat Deployment of the English Auction Service

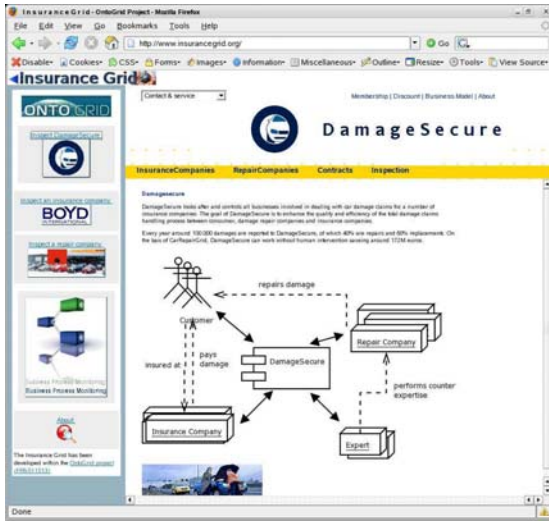


Figure 7: Front End of the InsuranceGrid

10 shows the quality of proposals for the CNP or bids for the English Auction with respect to time for the truth-telling and Decrement strategies. We compare these protocols with the case of no negotiation at all in the Insurance Grid. It can be seen that any negotiation provides a better quality of bids for DamageSecure than no negotiation. The CNP protocol having only one round of negotiation provides proposals of less value than the English Auction.

Figure 11 shows that the time constrained English auction performs better and yields better agreement for DamageSecure than the CNP. Figure 12 shows how the quality of the winning bids changes with an increasing auction deadline. It shows that as the auction run longer, there is an increase in the quality of the winning bids.

Overall we can deduce that it is more worthwhile for DamageSecure to participate in a full-blown negotiation protocol such as the English auction. Also, an auction with a longer deadline is more beneficial to DamageSecure than an auction with a shorter deadline, given that the cost of an auction running in minutes is negligible.

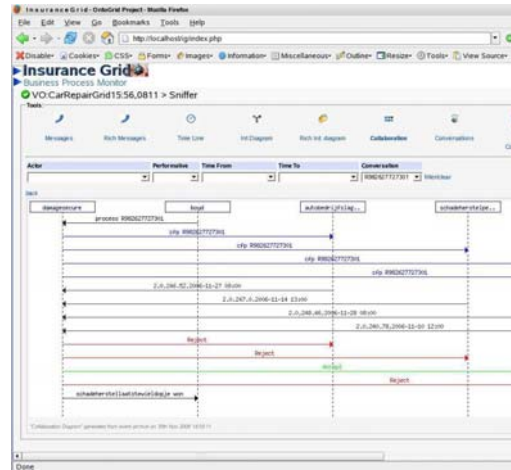


Figure 8: Business Process Monitoring in the InsuranceGrid

Test Case ID	Date	Status	Damage	Location	Item	Priority
Test_Case001	2006-05-07	new	rear middle	Scheveningen	P527334070	low Black
Test_Case006	2006-05-09	new	front right	Scheval	P527334070	U. V. B. F. Timmer
Test_Case003	2006-05-15	new	rear left	Loon op Zand	P527334070	J. J. Timmer
Test_Case002	2006-05-19	new	rear left	Wijksterik	P527334070	N. F. C. Z. Cohen
Test_Case001	2006-05-21	new	front left	Wijksterik	P527334070	N. R. O. D. Smulders
R0000220462	2006-06-08	new	right	Wijksterik	0112701658	D. K. W. de Vries
R0000790215	2006-06-07	new	left	Wijksterik	R000060710	N. R. O. D. Smulders
R0000061100	2006-06-07	new	front left	Den Haag	P4912001483	G. van Orange
R0000483226	2006-05-11	new	rear right	Tilburg	P3964019623	G. J. van Orange
R0000874985	2006-05-18	new	front middle	Tilburg	P4871783118	C. M. E. Wouda
R0000170602	2006-05-12	new	right	Wijksterik	0113086221	D. Smulders
R0000444610	2006-06-01	new	rear right	Scheval	R000052513	S. J. J. Wouda
R00047024526	2006-05-10	new	front middle	Wijksterik	P1900070233	R. J. Zalm
R0073270490	2006-05-15	new	rear middle	Den Haag	P4910840980	M. L. Linsen
R0072081932	2006-05-12	new	rear middle	Den Haag	P3242003060	Z. J. J. Janssen
R00699021910	2006-05-16	new	right	Wijksterik	R400000096	N. V. V. Linsen
R0061310181	2006-05-26	new	left	Loon op Zand	P4910840980	N. V. V. Linsen
R0062727263	2006-05-15	new	right	Loon op Zand	P443000812	J. J. Smulders
R00714060706	2006-05-15	new	left	Loon op Zand	P4914240708	N. R. R. Linsen
R00730832618	2006-05-10	new	right	Tilburg	P5681123369	D. Smulders
R007200713209	2006-05-19	new	front left	Breda	P5333571160	H. T. De Bruin
R00710817493	2006-05-16	new	front right	Breda	0248510159	H. T. De Bruin
R00621147863	2006-05-18	new	right	Wijksterik	P3243014708	H. T. De Bruin
R00402390227	2006-05-23	new	right	Scheval	0270220114	D. J. J. Bollenboom
R00462888074	2006-05-26	new	rear right	Scheveningen	P4910840980	G. van Orange
R00081724212	2006-05-18	new	front middle	Tilburg	P000247378	D. B. E. Wageningen
R00403307607	2006-06-07	new	rear left	Tilburg	P000247378	D. B. E. Wageningen
R0040750388	2006-06-06	new	front middle	Wijksterik	P5327330212	D. J. J. de Vries

Figure 9: Business Test Cases in the InsuranceGrid

6. CONCLUSIONS

The aim of our work is to develop semantic web services that can cooperate, reach agreements and self-organise in order to effectively and correctly solve overall problems. There is a need for such techniques in the insurance sector to save time and costs to expedite repair claims. As such, automated negotiation is already used successfully in number of financial domains, such as the stock market. To gain similar benefits, we believe that the use of automated negotiation in the insurance market provides a more healthy market for claim settlement than previously, leading to an increase in quality of repairs and to a decrease in prices of repairs. Fur-

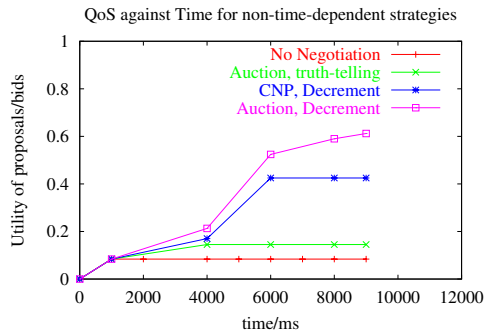


Figure 10: Comparison of Non-time dependent strategies

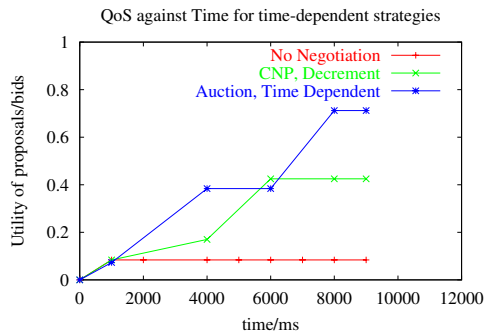


Figure 11: Comparison of time dependent strategies

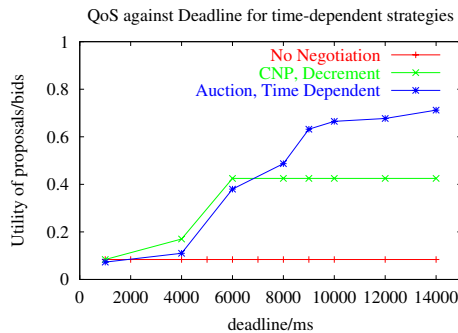


Figure 12: Quality of bids with changing deadlines

thermore, the speed of claim settlement (in Europe and the rest of the world) will rapidly increase through automated negotiation. Also the claim settlement process will be standardized within using the InsuranceGrid. The above are both demands of the 5th Motor Insurance Directive set by the European commission, (http://ec.europa.eu/internal_market/insurance/motor_en.htm).

Thus, we have identified the opportunity to re-use agent interaction and negotiation techniques for web services negotiation in the InsuranceGrid. In this paper, we described

our deployment and implementation of the CNP and of the English auction and their associated strategies for negotiation in the InsuranceGrid. The semantic aspect is achieved through querying of an ontology service to obtain the profile and preferences of DamageSecure, insurance and repair services. The findings of our deployment indicate the benefit of using automated negotiation techniques between web and grid services and in the InsuranceGrid.

Future work includes deploying the bilateral negotiation protocol and a deeper empirical evaluation of the deployed protocols. We are also investigating the integration of our negotiation work with the WS-Agreement specification. Another avenue of further work lies in investigating the sharing of ontologies and protocols in heterogeneous virtual organizations.

The work presented in this paper is funded by the FP6 EU project Ontogrid (FP6-511513). <http://ontogrid.eu/ontogrid/home.jsp>.

7. REFERENCES

- [1] BBC, Science and Nature. *BBC Climate Change Experiment*. BBC, <http://bbc.cpdn.org/>, 2006.
- [2] J. Brittain and I. Darwin. *Tomcat: The Definitive Guide*. O'Reilly, 2003.
- [3] M. Esteban Gutierrez, A. Gomez-Perez, and O. Munoz Garcia. Ontology access in grids with WS-DAI Ont and the RDF(S) realization. In *3rd European Semantic Web Conf. (ESWC 2006)*, 2006.
- [4] I. Foster, N. Jennings, and C. Kesselman. Brain meets brawn: Why grid and agents need each other. In *AAMAS 2004*, New York, USA, 2004. ACM.
- [5] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2003.
- [6] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1981.
- [7] J. Smulders, C. Van Aart, P. Van Hapert, V. Fintelman, and P. Storms. *Ontogrid, Deliverable 9.1, Business Cases and User Requirement Analysis*. EU FP6 project, <http://ontogrid.eu/>, 2005.
- [8] B. Sotomayor and L. Childer. *Globus Toolkit: Programming Java Services: Pt. 4*. Morgan Kaufmann, 2006.
- [9] Space Sciences Laboratory UC Berkeley. *Search for Extraterrestrial Intelligence*. University of California, <http://setiathome.ssl.berkeley.edu/>, 2004.
- [10] C. Van Aart. *Ontogrid, Deliverable 6.1, Business Cases Monitoring: Specification and Design of Intelligent Debugging Tool*. EU FP6 project, <http://ontogrid.eu/>, 2005.
- [11] C. van Aart, K. Van Marcke, R. Pels, and J. Smulders. International Insurance Traffic with Software Agents. In F. van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence*. IOS Press, Amsterdam, 2002.