An Ontology for Formalising Agreement Patterns in Auction Markets

José J. Durán[1] and Carlos A. Iglesias[2]

[1] Centro para las Tecnologías Inteligentes de la Información y sus Aplicaciones (CETINIA), Universidad Rey Juan Carlos (Spain)
[2] Depto. Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid (Spain)

Abstract. Knowledge and best practices on auction systems are currently disseminated across the research literature, which limits its access, reuse, evaluation and feedback by practitioners. This article presents a systematic approach to collect this knowledge as design patterns, in order to provide assistance to software developers. An ontology has been defined for formalising design patterns in auction systems, with the aim of improving its searchability by software developers. Finally, a case study illustrates how the proposed pattern ontology provides assistance in the development of a dynamic pricing model for an e-commerce service.

1 Introduction

Auctions provide a market system model that enable the exchange of resources on the basis of supply and demand. They have proved to be an effective model for dynamic pricing of resources in different scenarios such as electronic commerce [1], resource allocation [2], service pricing [3] or sponsored search pricing [4].

Nevertheless, software engineers have few available resources that provide them support in the design of auction mechanisms and automatic bidders, since current knowledge and best practices on auctions are disseminated across research publications. It is a good practice to identify the elements of good and reusable designs in auctions, and provide a systematic framework for formalising the experience with these designs. This is precisely the role of *design patterns* [5], which describe general reusable solutions to commonly recurring problems in software design. The notion of design patterns was originated in the object-oriented software engineering community and has been widely accepted by this community, having a strong impact on how object-oriented software is designed, implemented and communicated nowadays.

The purpose of this article is to provide a structured and formalised schema for describing design patterns in the field of agreement technologies and validate it through its application in the domain of auctions.

The rest of the article is organised as follows. Section 2 presents an ontology for describing agreement patterns, in order to provide standard facilities for retrieving patterns based on the user requirements. This agreement patterns ontology has been linked with an auction domain ontology for describing the main concepts of the auction domain, in order to provide a common language for describing the auction patterns. In order to show the applicability of our approach, a case study is developed within section 3. Finally, section 4 summarises the main contributions of this paper and the future research activities.

2 Modeling an Ontology for Auction Patterns

Even though design patterns are usually expressed in natural language, several works have proposed its formalisation in order to provide tool support for consulting a pattern catalogue and guide developers in its application. Ontologies [6] have been considered as a natural formalisation technique that enables an infrastructure for sharing and interconnecting semantically pattern languages in the web. Several works have used ontologies [7] for formalising both the structure of the patterns [8,9] (how to apply the pattern) as well as its intention (when to apply the pattern) [7,10].

This article presents an ontology for formalising the intention of agreement patterns in order to improve its findability. The ontology is organised into three levels as shown in fig. 1 which are detailed below. The first level defines an ontology (*APO*) for agreement patterns (section 2.1), with the aim of facilitating the searchability of the patterns by software developers. A domain ontology (*AUTERMS*) for the auction domain (section 2.2) provides the common terminology for describing auction patterns. Finally, an ontology (*AUPA*) has been defined for describing the auction patterns (section 2.3) . This ontology extends the *APO* ontology and is described using the *AUTERMS* ontology.

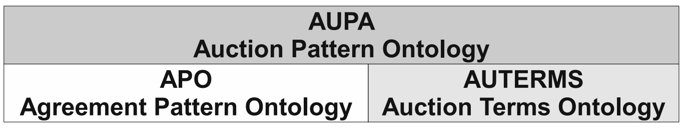


Fig. 1. Layers of the Auction Pattern Ontology

2.1 The Agreement Pattern Ontology (*APO*)

Agreement patterns [11] provide a way to collect best practices for reaching agreements in a structured way. In this section, the Agreement Patterns Ontology (*APO*) is introduced in order to catalogue agreement patterns. This catalogue should support developers in choosing a pattern for a given problem.

In order to determine the scope of the ontology, the ontology should be able to answer the following *competency questions* [12]:

- Which design patterns can solve a given problem?
- Which design patterns can solve a given problem and are applicable in a given context?
- Which design patterns are related to a given domain concept?
- Which design patterns are available for a given task?

The proposed ontology is based on the structure of the DPIO ontology[7]. That ontology describes the relationship between patterns and problem types, which are described by problem constraints. We have extended DPIO by focusing on pattern constraints, instead of problem type constraints.

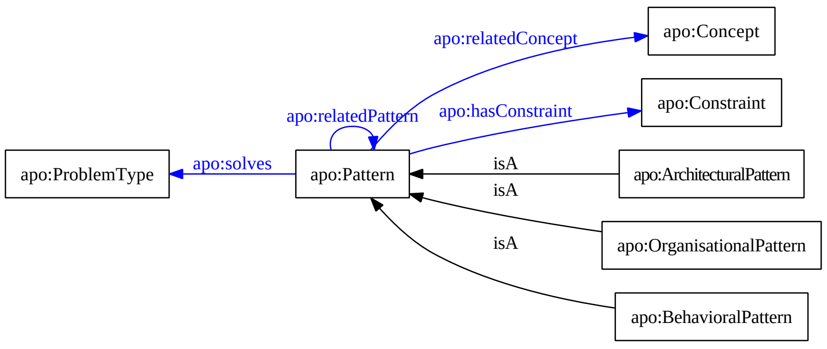


Fig. 2. Ontology for agreement patterns

The structure of the APO ontology is depicted in fig. 2. The main relationships of the core ontology are described below. A *Pattern* can solve one or more *ProblemTypes* and is applicable only if some *Constraints* are fulfilled. A *Pattern* can be related with other *Patterns*. Domain *Concepts* can be related with one or more *ProblemTypes* and one or more *Patterns*. *Patterns* are organised according to the task they solve into:

- *Organisational Patterns.* These patterns collect social structure patterns which define the norms, social and interaction model [13] which form the society as a whole, and which determine, to some varying degree, the actions of the individuals socialised into that structure.
- *Behavioural Patterns.* These patterns collect individual behaviours of the participants in the agreement in order to satisfy a goal.
- *Architectural Patterns.* These patterns describe architectural patterns describing the software architecture of the participants in the agreement.

2.2 The Auction terms Ontology (*AUTERMS*)

The objective of the Auction Terms Ontology (*AUTERMS*) is modelling the auction domain in order to provide a common vocabulary for describing auction patterns. Previous work have also proposed ontologies in the negotiation domain [14,15,16], and have been extended to the auction domain, but we have not found a specific ontology in the auction domain. We have modelled a basic ontology based mainly on the auction patterns proposed by Ré [17], the research survey on auctions by Parsons et al. [1] and the ontology for the agent trading competition developed by Wellman [18]. The core structure of the proposed ontology AUPA is shown in fig. 3.

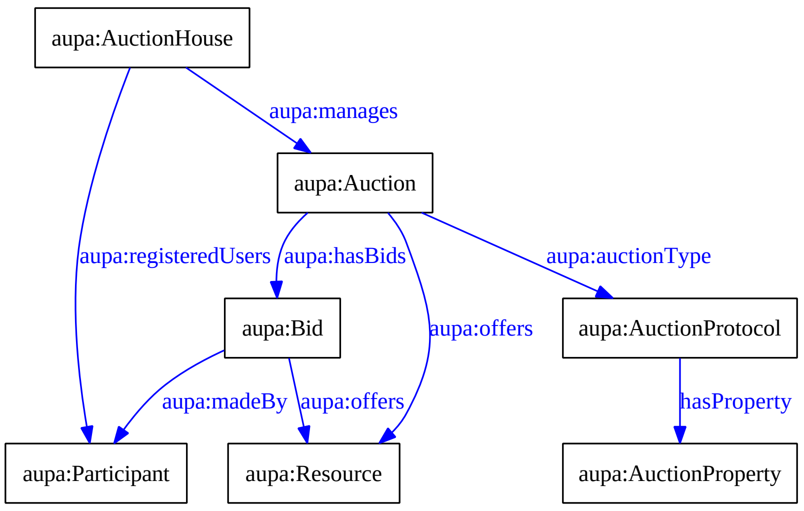


Fig. 3. Auction Ontology

The main concepts of this ontology are AuctionHouse, Auction, Participant, AuctionProtocol and AuctionProperty. *Auctions* are a negotiation process in which different *Participants* exchange information in the form of *Offers* in order to obtain a *Resource*. Those *Auctions* are placed in different *AuctionHouses* which represents the auctioneer of an *Auction*. An *AuctionHouse* has the responsibility of sharing *Auction* information between different *Participants* following the rules of a specific *AuctionProtocol*. An *AuctionProtocol* represents which *AuctionProperties* must be meet in a specific *Auction*. An *AuctionProperty* helps to classify an *AuctionProtocol* in order to find the most suitable based on the problem constraints.

2.3 Auction pattern Ontology (*AUPA*)

Previous works have also proposed the usage of design patterns in agreement technologies and auctions. Iglesias et al. [11] propose *Agreement Patterns* for formalising recurring solutions to agreement problems. Ré et. al. [17] propose auction patterns from an object oriented perspective. In the field of multiagent systms, Oluyomi [19] proposes a classification scheme for agent oriented patterns which is applied to a relevant number of agent patterns. Jureta et al. [20] describe three agent oriented patterns in the auction domain.

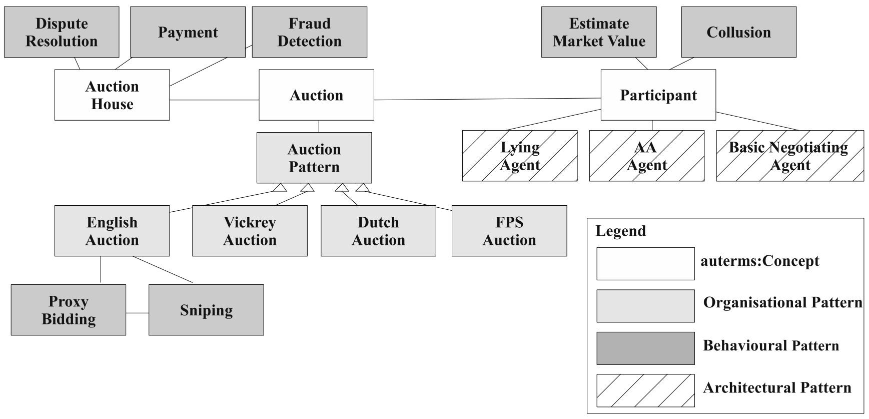


Fig. 4. Auction Pattern Catalogue

This article extends the work developed by Iglesias et al. [11] in order to organise, identify and describe semantically auction patterns as a specific family of agreement patterns. In order to validate our approach, several patterns have been formalised within this ontology. Fig. 4 shows these patterns grouped by the related Concept in the domain on *AUTERMS* ontology. In addition, the auction patterns are classified in Architectural, Organisational and Behavioural according to the *APO* Ontology.

The pattern catalogue includes the following patterns:

- *Organisational:* AuctionPattern, English Auction, Dutch Auction, Vickrey Auction, First Price Sealed Auction
- *Architectural:* Lying Agent [21], AA Agent [22], Basic Negotiation Agent [19]
- *Behavioural:* Proxy Bidding [20], Sniping, Dispute Resolution [20], Payment [20], Fraud Detection [20], Colussion, Estimate Market Value.

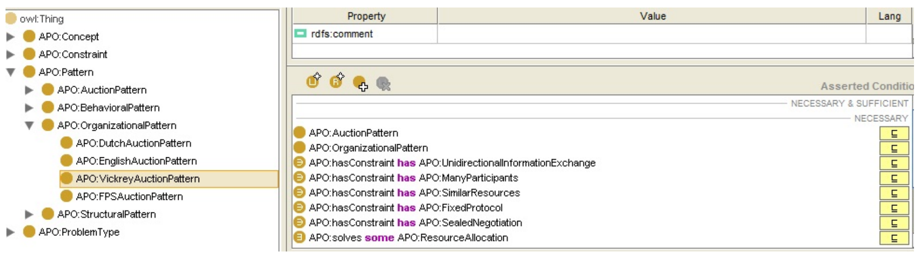
In order to illustrate how auction patterns are described, the pattern *Vickrey Auction* is described in natural language in table 1, while its semantic description is depicted in fig. 5 and fig. 6. The ontology has been defined using the ontology editor Protégé [23]. Thanks to the semantic description, the catalogue can be consulted and filtered according to the user constraints, as described in section 3.

Name	Vickrey auction
Alias	Second-price sealed-bid auction.
Keywords	Auction, service pricing.
Problem Type	Resource assignation.
Problem	Auction of multiple similar resources, or items, in which participants should be encouraged to share their true valuation using incentives.
Context	There are different consumers willing to get the resources, and the real value of the resource is not known.
Solution	This auction is defined by the next properties, as seen on fig. 5 : – **One round:** Bids are received in a unique round, which last until all participants have made their offer, or until a specific time. – **Sealed:** Offers are not visible, in any mean, to participants, until the auction has finished. – **Multiple items per bid:** Offers should be for more than a resource, and associated with a price. Only an unique bid per participant. – **Second pricing:** the highest n bids are awarded the resource and pay a price equal to the n+1 highest amount bid. – **Incentive compatibility:** Each participant maximizes its expected utility, by revealing their true valuation, as final price is not dependant on the offer made, but in the last highest offer. Also, in scenarios with different auctioned resources, a preference assignation of resources, will be an incentive to bid as high as possible [24]. The most important advantage of this auction, is that sellers don't require to have knowledge about buyers willingness to pay. Because of that, this auction is highly suitable in scenarios without that information.
Examples	Mobile spectrum assignment[1], Internet advertising[25].
Related patterns	Sealed auction, Fake bidding.

Table 1: Vickrey Organisational Pattern in Natural Language

3 Case study

In order to illustrate the practical application of agreement patterns, this section describes a case study consisting of the development of an opera ticket selling service.


owl:Thing
- APO:Concept
- APO:Constraint
- APO:Pattern
  - APO:AuctionPattern
  - APO:BehavioralPattern
  - APO:OrganizationalPattern
    - APO:DutchAuctionPattern
    - APO:EnglishAuctionPattern
    - APO:VickreyAuctionPattern
    - APO:FPSAuctionPattern
  - APO:StructuralPattern
- APO:ProblemType

Property	Value	Lang
rdfs:comment		

Asserted Conditio

NECESSARY & SUFFICIENT

NECESSARY

APO:AuctionPattern ⊑
APO:OrganizationalPattern ⊑
APO:hasConstraint **has** APO:UnidirectionalInformationExchange ⊑
APO:hasConstraint **has** APO:ManyParticipants ⊑
APO:hasConstraint **has** APO:SimilarResources ⊑
APO:hasConstraint **has** APO:FixedProtocol ⊑
APO:hasConstraint **has** APO:SealedNegotiation ⊑
APO:solves **some** APO:ResourceAllocation ⊑

Fig. 5. Screenshot of Protégé of the semantic description of the Vickrey Pattern

```
<owl:Class rdf:ID="VickreyAuctionPattern">
  <rdfs:subClassOf rdf:resource="#AuctionPattern"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#OrganisationalPattern"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasConstraint"/>
      </owl:onProperty>
      <owl:hasValue rdf:resource="#ManyParticipants"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:hasValue rdf:resource="#DivisibleResources"/>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasConstraint"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>

   // similar restrictions for #FixedProtocol, #SealedNegotiation,
   // and #UnidirectinalInformationExchange
```

Fig. 6. OWL class of Vickrey Auction Pattern

An opera thether desires to maximize its benefit from sold tickets, which are sold by different online enterntainment ticket selling services. Several strategies for offering this service are available, such as defining a fixed pricing policy based on position of the seat or dynamic pricing based on an auction protocol. The opera theather desires to explore this second alternative.

In order to evaluate the ontology, we are going to review the competency questions enumerated in section 2.1, translate these consults into formal questions in SPARQL, and evaluate the result set of the queries.

Which design patterns can solve a given problem?. In our case, we are interested in consulting all the organisational patterns that solve the problem *Resource Allocation*. Fig. 7 shows the corresponding SPARQL query. The result set contains all the Organisational Auction Patterns of the ontology. In a full version of the ontology we could obtain other patterns not related with auctions, such as bargaining.

```
SELECT DISTINCT ?pattern
WHERE {
  ?pattern rdfs:subClassOf apo:OrganisationalPattern.
  ?pattern rdfs:subClassOf apo:Pattern.
  ?pattern rdfs:subClassOf [
    rdf:type owl:Restriction;
    owl:someValuesFrom apo:ResourceAllocation;
    owl:onProperty apo:solves
  ].
}
```

Fig. 7. SPARQL query for organisational patterns for resource allocation problem

Which design patterns can solve a given problem and are applicable in a given context?. The results of the previous query can be filtered specifying constraints. In our domain, opera seats are auctioned. We can decide whether opera seats are divisible or indivisible. In our case, we can sell all the seats for each opera performances, but each seat can be sold at a different price, so we add the constraint *DivisibleResource*. Fig. 8 shows the corresponding SPARQL query, which would return the *Vickrey Auction Pattern*.

Which design patterns are related to a given domain concept?. In our domain, we could be interested in which design patterns are related with the *Auction House* concept as shown in Figre 9.

Which design patterns are available for a given task?. In case we desire to define the architecture of a bidder, we can query the *Architectural Patterns* related with the concept *Participant* (Fig. 10).

```
SELECT DISTINCT ?pattern
WHERE {
  ?pattern rdfs:subClassOf apo:Pattern.
  ?pattern rdfs:subClassOf apo:OrganisationalPattern.
  ?pattern rdfs:subClassOf [
    rdf:type owl:Restriction;
    owl:someValuesFrom apo:ResourceAllocation;
    owl:onProperty apo:solves
  ].
  ?pattern rdfs:subClassOf [
    rdf:type owl:Restriction;
    owl:hasValue apo:DivisibleResources;
    owl:onProperty apo:hasConstraint
  ].
}
```

Fig. 8. SPARQL query for organisational patterns of resource allocation of divisible resources

```
SELECT DISTINCT ?pattern
WHERE {
  ?pattern rdfs:subClassOf apo:Pattern.
  ?pattern rdfs:subClassOf [
    rdf:type owl:Restriction;
    owl:someValuesFrom auterms:auctionHouse;
    owl:onProperty apo:relatedDomainConcept
  ].
}
```

Fig. 9. SPARQL query for design patterns related with the Auction House Concept

```
SELECT DISTINCT ?pattern
WHERE {
  ?pattern rdfs:subClassOf apo:Pattern.
  ?pattern rdfs:subClassOf apo:ArchitecturalPattern.
  ?pattern rdfs:subClassOf [
  rdf:type owl:Restriction;
    owl:someValuesFrom auterms:participant;
    owl:onProperty apo:relatedDomainConcept
  ].
}
```

Fig. 10. SPARQL query for architectural patterns related with the concept Participant

4 Conclusions and future work

Design patterns can contribute to promote knowledge reuse and advance in the field of agreement technologies, since the expertise of practices is formalised and can be easily shared among practitioners, allowing them to share their experiences and understand better the advantages. limitations and applicability of these patterns.

In this paper, several auction patterns have been identified in the research literature and described and classified according to a pattern form. In addition, a domain ontology for auctions has been defined in order to provide automated reasoning on the application of patterns.

Current work is focused on several directions. First, our aim is progressing on the formalisation of the patterns themselves in order to provide and ontology-based design pattern repository as [26,27,7]. Second, since the targeted users of this research are software developers, our goal is providing at hand support during their development tasks, through the integration of this tool in a standard IDE such as Eclipse or Netbeans. In addition, this paper has presented an initial set of identified patterns, and our aim is enlarging this set, analysing available systems and documented practices as well as with the cooperation of other researchers and users.

Acknowledgements

This research has been partly funded by the Spanish Ministry of Science and Innovation through the projects Ingenio Consolider2010 AT *Agreement Technologies* (CSD2007-0022)), OVAMAH (TIN2009-13839-C03-02) and T2C2 (TIN2008-06739-C04-03/TSI) as well as the Spanish Ministry of Industry, Tourism and Trade through the project RESULTA (TSI-020301-2009-31).

References

1. Parsons, S., Rodríguez-Aguilar, J.A., Klein, M.: Auctions and bidding: A guide for computer scientists. **24**(2) (August 2004) 271–287
2. Lin, W.Y., Lin, G.Y., Wei, H.Y.: Dynamic Auction Mechanism for Cloud Resource Allocation. IEEE (May 2010)
3. Szymanski, B.: A Novel Auction Mechanism for Selling Time-Sensitive E-Services. IEEE
4. Asdemir, K.: Bidding patterns in search engine auctions (2006)
5. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Resusable Object-Oriented Software. Addison-Wesley Professional (1995)
6. Henninger, S., Corrêa, V.: Software pattern communities Current practices and challenges. In: Proceedings of the 14th Conference on Pattern Languages of Programs (PLoP). (September 2007) 2007
7. Kampffmeyer, H., Zschaler, S.: Finding the pattern you need: The design pattern intent ontology. Lecture Notes in Computer Science **4735** (2007) 211

8. Henninger, S., Ashokkumar, P.: An Ontology-Based Metamodel for Software Patterns. at 18th Int. Conf. on Software Engineering . . . (2006)
9. Montero, S., Díaz, P., Aedo, I.: Formalization of web design patterns using ontologies. Lecture Notes In Computer Science (2003) 179–188
10. Harb, D., Bouhours, C., Leblanc, H.: Using an Ontology to Suggest Software Design Patterns Integration. Lecture Notes In Computer Science (2009) 318
11. Iglesias, C.A., Garijo, Fernández-Villamor, J.I., Durán, J.J.: Agreement patterns. In: Proceedings of the CAEPIA09 Workshop on Agreement Technologies(WAT 2009), Sevilla, WAT09-CAEPIA09 (November)
12. Noy, N.F., McGuinness, D.: Ontology Development 101: A Guide to Creating your First Ontology. Technical Report SMI-2001-0880, Stanford Medical Informatics, Stanford (2001)
13. Dignum, V., Vázquez-Salceda, J., Dignum, F. In: Omni: Introducing social structure, norms and ontologies into agent organizations. Springer-Verlag, New York (2005) 181–198
14. Tamma, V., Wooldridge, M., Dickinson, I.: An ontology based approach to automated negotiation. In Padget, J.A., Shehory, O., C. Parkes, D., Sadeh, N.M., Walsh, W.E., eds.: Agent Mediated Electronic IV. Designing Mechanisms and Systems. AAMAS 2002 Workshop on agent mediated electronic commerce (AMEC IV), held at the AAMAS 02 conference, Bologna, Springer Verlag (2002) 219–237
15. Tamma, V., Phelps, S., Dickinson, I., Wooldridge, M.: Ontologies for supporting negotiation in e-commerce. Engineering Applications of Artificial Intelligence **18**(2) (March 2005) 223–236
16. Paschke, A., Kiss, C., Al-Hunaty, S.: NPL: Negotiation Pattern Language. economics and management (2005)
17. Ré, R., Braga, R., Masiero, P.: A pattern language for online auctions management. Proceedings of PLoP (2001)
18. Wellman, M., Wurman, P., O'Malley, K., Bangera, R., Lin, S.d., Reeves, D., Walsh, W.: Designing the market game for a trading agent competition. IEEE Internet Computing **5**(2) (2001) 43–51
19. Oluyomi, A.O.: Patterns and Protocols for Agent-Oriented Software Development. PhD thesis, Faculty of Engineering. University of Melbourne, Australia. (November 01 2006)
20. Jureta, I., Kolp, M., Faulkner, S., Do, T.T.: Patterns for Agent Oriented e-Bidding Practices. In: Proceedigns of 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2005), Part IIKES (2), Melbourne,Australia (2005) 814–820
21. Koolmanojwong, S., Jiamthapthaksin, R., Daengdej, J.: An agent architecture for competitive application environment. Aerospace Conference, 2004. Proceedings. 2004 IEEE 3079–3089
22. Vytelingum, P., Cliff, D., Jennings, N.R.: Strategic bidding in continuous double auctions. Artificial Intelligence **172**(14) (2008) 1700–1729
23. Stanford Center for Biomedical Informatics Research: Protégé ontology editor and knowledge acquisition system. http://protege.stanford.edu/ (2006)
24. Nishino, N., Fukuya, K., Ueda, K.: Service Design in Movie Theaters Using Auction Mechanism with Seat Reservations
25. Edelman, B., Ostrovsky, M.: Strategic bidder behavior in sponsored search auctions. Decision Support Systems **43**(1) (2007) 192–198
26. Harb, D., Bouhours, C., Leblanc, H.: Using an ontology to suggest software design patterns integration. In Chaudron, M., ed.: Models in Software Engineering.

Volume 5421 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2009) 318–331

27. Pavlic, L., Hericko, M., Podgorelec, V.: Improving design pattern adoption with Ontology-Based Design Pattern Repository. IEEE (June 2008)