AUTOMATIC COMPOSITION OF WEB SERVICES BY PLANNING TECHNIQUES

Jaime Alberto Guzmán Luna, Demetrio Arturo Ovalle Carranza

GIDIA: Grupo de I + D en Inteligencia Artificial
Escuela de Sistemas, Universidad Nacional de Colombia – Sede Medellín
Calle 59A No 63 – 20, Medellín, Colombia
E-mail: {jaguzman, dovalle}@unal.edu.co

Abstract: The composition of Web Services has been a topic of vital interest for supporting the tasks of both electronic commerce and the integration of applications among different firms by means of the web. The aim of this paper is to present an analysis about how the planning techniques in Artificial Intelligence can be used to carry out the composition of web services. This paper describes the main solutions outlined so far under this approach and the current problems for carrying out a true automatic composition, problems which become challenges still to be solved in this line of research.

Keywords: Web Services, Planning in Artificial Intelligence, Semantic Web.

1. INTRODUCTION

The ability to carry out the composition of Web services in an automatic way will change significantly most of the application areas for the technology of web services including both the electronic commerce and the integration systems.

The semantic Web has allowed the semantic marking of the Web services by means of ontologies, which in turn has made it easier for the machines to better understand the way they work, with which through specific reasoning techniques could carry out the task of composition of services.

An area which points in this direction is the planning in Artificial Intelligence (AI), which is an area of research mature enough oriented towards reasoning about the action through different techniques developed along its existence. This paper tries to show that AI planning can be applied to this problem supporting itself in the use of the semantic Web.

This paper is organized in the following way: Section 2 describes the basic aspects of the Web services. Section 3 presents the problem of the composition of Web services as AI planning problem and it presents as well a brief description of the main related works. Section 4 illustrates the different problems that outline the AI planning techniques to face the problem of composition of services. Finally, section 5 presents the conclusions and future work.

2. THE WEB SERVICES

The Web services are autonomous self-described modular applications that can be published, localized and requested through the Web (Martin, 2001). They constitute software modules that “describe a collection of operations that are accessible on the web through standardized messages in XML” (Kreger, 2001).

The research related to the Web services has oriented several of its initiatives toward defining standard forms for innovation (getting messages through), description and discovering services, allowing an easier integration of heterogeneous systems: The SOAP (Simple Object Access Protocol) (W3C Consortium, 2003), permits the exchange of data between application client and application server, by means of a common codification mechanism and format of messages. The WSDL (Web Services Description Language) (W3C Consortium, 2001), allows the description of a Web service by using metadata structured upon the interface that tries to use the application client as well as the written documentation about the Web service including examples of its use. The UDDI (Universal Description Discovery Integration) (UDDI Consortium, 2004), provides a public directory for registration and searching of Web services. It permits the description of the Web services through their physical attributes such as the name, address and functions that they offer. Additionally, the descriptions of UDDI are extended by an additional set of attributes that describe the necessary characteristics to classify the services into categories.

One of the draw backs of the standards based on XML, like those examined before, is their void of explicit semantics so that two identical XML descriptions can mean things totally different, depending on the context they are being used. This diminishes the capacity of finding similar services. The latter is important because the user requesting a Web service does not know which services are available at a certain time and so the semantic knowledge could help identifying the more adequate services for a particular task.

The endeavors for integrating semantic in the Web services allowed the development of the OWL-S ontology (initially known as DAML-S) (Darpa Agency, 2003; W3C Consortium, 2004, Martin, 2006). This consists really in a set of ontologies (Gruber, 1993) for describing both the properties and capacities of the Web services. Its purpose is the supporting of the effective automation of different types of tasks including the discovering, composition, execution and monitoring of the Web services by using intelligent software agents (Hendler, 1999). The superior level of his ontology is composed by: the type Service, which is the root of the ontology, the type ServiceProfile, which describes both the capacities and parameters of the service, the type ServiceModel, which specifies how does the service work (flow of possible operations and paths) and the type ServiceGrounding, which defines the link of the service (protocol, type of messages, etc.), that is, how both the inputs and outputs of the atomic processes of the service would be. That is how, using the ServiceProfile and the ServiceGrounding is possible to carry out the tasks of searching services based on the functions they provide.

In order to support the composition of services several languages have been developed, among which stand out: the BPELAWS (Business Process Execution Language for Web service) (Curbera et al., 2002), the WSCI (Web Services Choreography Interface) (Intalio et al., 2002) and the OWL-S (by means of ServiceModel). However, carrying out the automation of the composition of Web services, a language needs well define semantics along with its syntactic structures. The semantics help in the definition of the reasoners for interpreting the descriptions of the web services.

The OWL-S with its base firmly rooted in description logics has got well established formal semantics. The ServiceProfile and the ProcessModel of OWL-S have been structured to allow intelligent agents (Hendler, 1999) interpreting the market and reason about the composition making easy its automation, in contraposition to BPELAWS and WSCI which do not exhibit any semantic form.

Next is detailed how different techniques of AI planning can be applied to carry out the automatic composition of Web services by using ontologies like OWL-S.

3. THE COMPOSITION OF SERVICES AND THE PLANNING TECHNIQUES

The composition of Web services is defined as the process through which new personalized services are created from already existent ones by means of a process of discovering, integration and dynamic execution of these services in a deliberative order for satisfying the requirements of the users (Chakraborty et al., 2002).

The Artificial Intelligence has, for a long time, centered on reasoning about actions and plans for facing real problems through one of its more important disciplines: the AI planning (Allen, 1990; Ghallab et al., 2004; McDermott and Hendler, 1995; Nilsson, 1980). Planning is the reasoning about the action. It is a process of explicit and abstract deliberation that chooses and organizes actions anticipating to the expected results. This deliberation tries to reach by the best possible way some pre-established objectives. The automatic planning is an area of Artificial Intelligence that studies this process of deliberation computationally.

In general a planning problem can be described like a tuple (S,So,G,A,R), where S is the set of all the possible states of the world, $So \subset S$ enotes the initial state of the world, $G \subset S$ denotes the objective state of the world in which the planning system will try to search, A is the set of actions that the planner can carry out from one state to another of the world, and the relation of translation $R \subseteq S$ x A x S defines the precondition and effects for the execution of each action. In terms of the Web services, So and G are the initial state and the objective state specified in the requirement of Web service requesters. R denotes the state changing function of each service.

Next are presented a set of works current in the literature that apply some AI planning technique for carrying out the composition task of Web services (ICAPS, 2003; ICAPS, 2004; AAAI, 2005). Due to the expressivity power of OWL-S based on description logics (Grosof et al., 2003), most of the related works use it as an outer language either to describe Web services or for describing the composed service (resulting plan).

3.1. Situation Calculus

Mcllraith et al. (McIlraith and Son, 2002; McIlraith et al., 2001; McIlraith and Narayanan, S., 2002) are based on a model of software agents that can reason about the Web services to carry out in an automatic way the discovering, execution, composition and inter-operation of Web services. In this work the authors adapt and extend the language Golog (Levesque et al., 1997), a language of logic programming built upon situation calculus, a classic tool for planning, which is used as a natural formalism by the agents for representing and reasoning about this problem. In this model, the user's requirement and the restrictions of the service are represented through the language of first order of situation calculus.

The authors defined each Web service like a primitive action or a complex action. The primitive actions are conceived as actions that alter the world because they change the state of the world or actions that collect information because they change the agents' state knowledge. The complex actions are compositions of individual actions. The basic agent of knowledge provides a logic codification of both the preconditions and effects of the actions of the Web services in the situation calculus language. The agents use structures of a procedural programming language composed with defined concepts for the Web services and their restrictions using deductive mechanisms. So a composed service is a set of atomic services linked by the language structures of procedural programming language (if-then-else, while, and others).

3.2. Hierarchical Task Network (HTN) Planning

The HTN planning (Erol et al, 1994) is a planning method by decomposition of tasks. Contrary to other planning concepts, the central HTN' concepts are not the states, but the tasks.

A planning system based on HTN decomposes the chosen task into a set of subtasks, and so on, until the resulting set of tasks consists of just atomic tasks (primitive), which can be executed directly when invoking some atomic operation. During each round of decomposition of tasks, certain given conditions have to be assessed if they are violated (for instance, exceeding a certain amount of financial resources) or not. The problem of planning is totally solved if the chosen complex task is decomposed into a set of primitive tasks without violating any of the given conditions.

An approach for using HTN planning on the Web services was proposed in (Wu et al., 2003), in which the SHOP2 planner is applied for the composition of Web services that are described through the DAML-S. SHOP2 is an HTN planner. The authors believe that the decomposition concept of tasks in the HTN planning is quite similar to the decomposition concept of composed processes outlined in the ontology of the DAML-S processes. The authors also maintain that the HTN planner is more efficient than other planning languages, as it is the case of Golog. In this work the process of translating DAML-S into SHOP2 is very well detailed.

3.3 Rule-based Planning

A technique for generating composed services from a declarative description at a high level is presented in the work of Medjahed (Medjahed, 2002). The method uses composability rules for establishing whether two services are compatible for composing a service. His composition approach is based on four phases: First the phase of specification allows a high level description of the desired composition by using a language named Composite Service Specification Language (CSSL). The matchmaking phase uses composability rules for generating composition plans that conform to service requester's specifications. If there does exist more than one generated plan, in the selection phase, a plan based on the established quality parameters of the composition (OoC) is chosen.

The final phase is the generation phase. A detailed description of the composed service is generated automatically and presented to the user requiring the service.

In this work, the emphasis on the composability rules is important since they are the ones that influence more on the plan generation. The composability rules consider both the syntactic and semantic properties of the Web services. The syntactic rules include rules for ways of operating and the rules for protocols of interaction links among the services. The semantic rules include the following aspects: (1) the messages composability, with which is defined whether two services are compatible for carrying out a composition analyzing if the output message of a service is compatible with the input message of another service; (2) the operation semantic composability, defines the compatibility among the domains, categories and purposes of the two services; (3) the quality of the composition, defining the preferences of the users of services taking into account the quality operations for the composed service; (4) composition soundness, considering if a service composition is reasonable.

The composability rules are the main contribution of this method, since they define the possible attributes of the Web services that can be used in the service composition. These rules can be used as a guide to other methods based on planning. Another tool for constructing composed services using the generation of a plan based on rules is SWORD (Shankar and Fox, 2002). SWORD does not employ the standards used in the description of services such as WSDL and DAML-S, instead it uses an entity-relation (ER) model for specifying Web services. In SWORD, a service is modeled through both its preconditions and posconditions. They are specified in a general model that consists of entities and relation among the entities. A Web service is represented in the form of a Horn rule that denotes the preconditions which are reached if the preconditions are true. In order to create a composed service, the user requiring the service only needs to specify the initial and final states to compose the service, then the generation of the plan can be reached using an expert system based on rules.

Along with the general composition methods, an interesting work done by SWORD is the analysis done about the uncertain results that sometimes can generate the linking based on rules if a precondition can not establish just one poscondition. The authors argue that the uncertain results can be avoided only when the preconditions are functionally dependent with the posconditions within a service. This fact can happen in most of the works related in this paper, but not all the authors declare it as explicitly as this work does it.

4. PROBLEMS OF THE PLANNING TECHNIQUES IN THE COMPOSITION OF WEB SERVICES

In spite of the works done in the area of AI planning for carrying out the composition of Web services, so far does not exist a true automatic solution for this task. This is due to a series of problems that present the planning techniques for facing this domain, which by and large have not been deeply treated in the solutions outlined in the former section (Sirivastava, 2003).

4.1. Problems related to the modeling of the world

The main problem of most of the planning models based on states derives from the fact that the Web is a vast and dynamic environment of active resources. On the other hand, the planning domains are usually characterized by states of the domain that are totally known, where the planner is the only agent that can change the state of the domain. The research endeavors should try to capture the main characteristics of a state of the Web, among which stand out:

Management of Incompleteness. The state of the Web is vast and can not be completely known. The Web domain is inherently incomplete and is not expected to model within the planner a complete description of all the Web, then mechanisms and strategies are required in order to carry out the knowledge acquisition and circumscribe the description of the domain of the planning problems to be handled (Kuter, 2005).

Management of events. The state of the Web changes independently of the planner. In the planning models this is usually handled through the concept of events, that is, the changes of state that are out of the planner's control (Ghallab, 1998). In the domain of the Web time seems to be a crucial factor during the composition, since during its planning changes can happen dynamically.

Management of inconsistencies. The state of the Web can contain contradictory information coming from different information sources. Concepts such as certainty and believes should be modeled and kept in mind during the composition process for dealing with this aspect (Ambite, 1998).

4.2. Problems related to actions and operators of composition mechanism

It is important to highlight that an adequate composition mechanism not only has to keep in mind the fore mentioned elements about the state of the Web, but also this model has additionally to consider other specific characteristics of the Web related to the modeling of its actions/operators (McIlraith and Fadel, 2002).

The operators can change, appear and disappear. On the Web, new services are available and change as time passes, while old services disappear. It is no possible to assume that these characteristics would be under the planner and/or the executor's control, but it is to be expected though that the composition mechanism would be capable of discovering, monitoring and keep updated such changes in its model of the domain.

The time of the actions execution. Many planning models assume an instant execution of the actions, and do not take into account the required time for carrying out its execution. Moreover, it is frequent that planners delegate to the execution phase, the monitoring of the failures of its actions.

In contrast, the actions on the Web take time (depending on factors like the broad of the band and the overcharged in the servers) and probably fail. That is why it is necessary looking for a model that includes both time restrictions and that takes into account the handling of failures right from the plan generation phase.

4.3. Problems related to the plans representation

"A plan is a set of actions, that if executed at the initial state they will be transformed into the object state". This definition of a solution plan is common to most planning models (Fikes, 1971) and it is easy to see that this definition can be also applied to the Web's domain. At present, most of the Web tasks and of the objectives oriented toward activities that are carried out on the Web can be described basically as sequence of actions, that is on the Web, the plans are sequences required of interrelated services done about Web entities.

Although a plan as a total sequence or partially ordinates of activities are a formalism that is sufficient for planning on the Web at a basic level. However, it is meaningful researching for new planning models that can reflect in the structure of their plans, some typical element of the Web's domain, like for instance the management of contingencies, the no-determinism, and dynamic aspects of this domain. In order to achieve this, it is necessary to work on planning models that have the following characteristics.

Expressive and flexible models of plans. There are a few models that have a satisfactory handling of actions with duration, cycles, conditionals, etc. (Lin, 1995).

Models that combine the knowledge of the plan with tasks oriented languages. The planning models based on hierarchical approaches (Erol, 1994b), as well as models for workflow management (PLANET Network, 2003) offer examples of formalisms oriented to tasks that can be included in generative planning models, which look quite similar to how the sequences of activities on the Web are carried out.

4.4. Problems related to plan monitoring, execution, censing and replanning

In the traditional planning models, the solution plan is monitored during the execution phase. It is how the actions to be executed must be selected, the proper performers are shot and the sensors are activated in order to monitor in the real world if the effects expected from the actions are achieved. If the information coming from the sensors indicates that the actions fail, then frequently, a planning mechanism is appealed for.

Again the Web's domain (where the virtual and the real are things that overlapped) can be characterized in a "specific Web" through concepts related to planning such as the execution and the censing.

The execution. Since the Web is huge, the execution may imply complex decisions about the service selection to be invoked among a set of available and equivalent services. Moreover, the execution takes time as much for the processing as for the data transferring, then a series of criteria are necessary to combine both the problems of time execution with the band's broad problems.

The censing. In the Web's environment, the censing mainly means looking actively for information, that is, collecting much information (Knoblock, 1995; Golden, 1998; Naveen and Knoblock, 1997) as results from the Web sources (for instance Web pages, searching in data bases, videos transference, results from services that are invoked). The main problems open in this area are related to the information from different no homogeneous sources and the decisions to be taken with what has been censed.

5. CONCLUSIONS

The problem of the composition of Web services was explored in this work outlining how to apply the AI planning techniques for carrying out this task. Also several existent works were related in which different planning techniques are applied in Artificial Intelligence to carry out the task of composing services automatically. Lastly, several of the main problems found when applying AI techniques in this domain were identified, discussing its characteristics.

In general, the solutions for carrying out the automatic composition of services that are outlinted in this paper, use planning techniques that can be classified as classics. They are based on a series of simplifications of the real world that originate the problems outlined in the former section (Pollack and Horty, 1999). However, several of these problems could be tackled by different techniques coming from a line of AI planning research known as practical planning (Weld, 1999; Wilkins, 1988] or neoclassic planning (Ghallab *et al*., 2004).

As a future work, the proposal is to study the application of practical planning techniques to look for, in the context of automatic composition of services, tackling the specific problems related to the managing of incomplete information about the state of the world during the planning of the composed service as well as the managing of the uncertain behavior of the services during their execution, failings presented in general by most solutions outlined in this paper.

For this, in the context of the composition of services the utilization of contingent planning techniques (Warren, 1976; Etzioni *et al*., 1992; Blythe, 1998), reactive planning (Drummond *et al*., 1993; Belker *et al*., 2004), and the integration of both planning and execution (Sapena and Onaindia, 2004, Lemai and Ingrand, 2004) will be studied trying to apply and adapt the best technique or a hybrid solution capable of solving these problems. In the same way, different techniques for the perception of the Web surroundings and its integration with these planning techniques (McCarthy and Pollack, 2000; Drapper et al., 1994), will be studied.

ACKOWLEDGEMENTS

This work is partially funded by the DINAIN Project 20101006096, "Multi-Agent Model of Planning Under Incomplete Knowledge for the Automatic Web Services Composition" and the Doctoral Thesis "Multi-Agent Model of Concurrent Planning and Exècution for the Semantic Web Services Composition in Partially Observable environments", sponsored by Colciencias, Universidad Nacional de Colombia, Medellin campus, and the World Bank, framed within the supporting program to the national scientific community in Doctoral programs, concourse 2004.

REFERENCES

AAAI. (2005). Twentieth National Conference on Artificial Intelligence, AAAI 2005, Pittsburgh, Pennsylvania.http://www-rcf.usc.edu/~skoenig/ icaps/icaps04/, last visit, April, 15 of 2006.

Allen, JF., Hendler, J. Tate, A. (1990). Reading in Planning, Morgan Kaufmann.

Ambite, JL., Knocblock, CA. (1998). Flexible and scalable query planning in distributed and heterogeneus enviroments. In: *Proceedings of the 4th. International Conference on*

Artificial Planing Systems (AIPS-98), Ed Simmons R., Veloso M. and Smith S. Editors, AAAI Pres, Menlo Park, California, USA. pp 3-10.

Belker, Th., Hammel, M., amd Hertzberg, J. Plan projection under the APPEAL robot control architecture. In: *Proceedings of the 8th conference on Intelligent Autonomous Systems*.

Blythe, J. (1998). Planning under Uncertainty in Dinamic Domains. PhD Thesis, Computer Science Department, Carnegie Mellon University, Pittsburg, PA.

Chakraborty, D., Finin, T., and Perich, F. (2002) A Reactive Service Composition Architecture for Pervasive Computing Enviroments, Singapore, PWC.

Curbera F., Goland, Y., Klein, J., Leymann, F., Roller, D, Thatte, S. and Weerawarana, S. (2002). Business Process Execution Language for Web Services, http://www-106.ibm.com/ developerworks/library/ws-bpel/, Version 1.0.

Darpa Agency. (2003). DAML-S and OWL-S: http://www.daml.org/services/daml-s/0.9/, last visit, March, 29 of 2006.

Draper, D., Hanks, S. and Weld, D. (1994). Probabilistic planning with information gathering and contingent execution. Proceedings of the 2nd Conference on AI Planning Systems (AIPS), pp 31-36.

Drummond, M., Swanson, K., Bresina, J. and Levinson, R. (1993). Reaction-first search. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1408-1414.

Erol, K., Hendler, J., and Nau, D. (1994). Semantics for Hierarchical task network planning. Technical Report CS-TR-3239, Computer Science Department, University of Maryland.

Erol, K., Nau, D. and Hendler, J. (1994b). HTN: Planning: Complexity and Expressivity. In: *Proceedings of AAAI-94*, Seatle.

Etzioni, O., Hanks, S., Weld, D, Draper, D., Lesh, N., and Williamson, M. (1992). An approach to planning with incomplete information. In: proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning, pp. 115-125.

Fikes, R. and Nilson, N. (1971). STRIPS: A new approach to the Application of Theorem Providing to Problem Solving. *Artificial Intelligence*, 2, 189-208.

Ghallab, M, Nau, D. and Traverso, P. (2004). Automated Planning. Theory and Practice. Morgan Kaufman.

Ghallab, M. (1998). Chronicles as a practical representation for dealing with time, events and actions. In: *Proceedings of the 6th Italian Conference on Artificial Intelligence (AIIA'98)*. Padova (Italy) Lecture Noteson AI, pp.6-10.

Golden, K. (1998). Leap before you look: information gathering in the PUCCI Planner, In: *Proceedings of AIPS*.

Grosof, B., Horrocks, I., Volz, R. and Decker, S. (2003). Description logic programs: Combining logic programs with description logic. In: *The 12th International Conference on the World Wide Web (WWW-2003*), Budapest, Hungary.

Gruber, T.R. (1993). "Toward principles for the design of ontologies used for knowledge sharing". In: Formal Ontology in Conceptual Analysis and Knowledge Representation. The Netherlands: Kluwer Academic Publishers.

Hendler, J. (1999). Is there an intelligent agent in your future. Nature, Web Matters.

ICAPS. (2003). The 13th International Conference on Automated Planning & Scheduling, Trento Italy. http://icaps03.itc.it/, last visit, April, 10 of 2006.

ICAPS. (2004). The 14th International Conference on Automated Planning & Scheduling, Whistler, British Columbia, Canada. http://www-rcf.usc.edu/~skoenig/icaps/icaps04/, last visit, April, 10 of 2006.

Intalio, Sun Microsystems, BEA Systems, SAP. (2002). Web Service Choreography Interface (WSCI).

Knoblock, GA. (1995), Planning Executing, Sensing and Replanning for Information Gathering, In: *Proceedings of the International Joint Conference on AI, IJCAI*.

Kreger, H. (2001). Web Services Conceptual Architecture (WSCA 1.0). http://www-4.ibm.com/software/solutions/webservices/, last visit, April, 15 of 2006.

Kuter, U., Sirin, E., Parsia, B., Nau, D., and Hendler, J. Information gathering during planning for web service composition. Journal of Web Semantics, 3(2), 2005.

Lemai, S. And Ingrand, F. (2004). Interleaving temporal planning and execution in robotics domains. In: *proceedings of 15th International Joint Conference on Artificial Intelligence (AAAI).*

Levesque, J., Reiter, R., Lesperance, Y., Lin, F. and Scherl, R. (1997). GOLOG: A logic Programming Language for Dynamic Domains. Journal of Logic Programming. 31 (1-30), 59- 83.

Lin, SH. Dean T. (1995). Generating optimal policies for high-level plans with conditional branches and loops. *In: Proceedings of the Third European Workshop on Planning*. pp 205-218.

Martin, D., Burstein, M., McDermott, D., McIlraith, S., Paolucci, M., Sycara, K., McGuinness, D., Sirin, E., and Srinivasan, N. (2006). Bringing semantics to web services with OWL-S. World Wide Web Journal Special Issue on Web Services: Theory and Practice. To Appear.

Martin, J. (2001). Web Services: The Next Big Thing. XML Journal. http://www.sys-con.com/xml/ last visit, March, 29 of 2006.

McCarthy, CE. And Pollack, ME. (2000). Towards focused plan monitoring: A technique and an application to mobile robots. Autonomous Robots, 9(1):71-81.

McDermott, D. and Hendler, J. (1995). "Planning: What is, What it could be, an introduction to the special issue on planning and scheduling". Artificial Intelligence, Vol 76, pp. 1-16.

McIlraith, S. and Fadel, R. (2002). Planning with Complex Actions. In: *Proceedings of AIPS 2002 Workshop on Exploring Real World Planning*, Tolouse, France.

McIlraith, S. and Narayanan, S. (2002). Simulation Verification and automated composition of Web Service. In: *the Proceedings of the Eleventh International World Wide Web Conference*, Honolulu, Hawaii, HI, USA.

McIlraith, S. and Son, TC. (2002). Adapting Golog for composition of semantic web services. In: *Proceedings of the Eighth International Conference on knowledge Representation and Reasoning (KR2002)*, Toulouse, France.

McIlraith, S., Son, TC. and Zeng, H. (2001). Semantic Web Services. IEEE Intelligent Systems, 16 (2), 46-53.

Medjahed, B., Athman, B., and Ahmed K. E. (2002). Composing Web Services on Semantic Web. The VLDBJournal, 12(4).

Naveen, A. Knoblock, GA. (1997). Levy Information Gathering Plans with Sensing Actions. In: *Recent Advances Planning in AI Planning: Proceedings of European Conference on Planning ECP97*. Springer-Verlag, New York.

Nilsson, NJ. (1980). Principles of Artificial Intelligence. Tioga Publishing.

PLANET Network. (2003). Workflow-Roapmap, Borrajo D. editor, Planet Workflow Management R&D Roadmap, PLANET-II Technical Coordination Unit on workflow Management, http://www.planet-noe.org/ TCUs/index.html, last visit, April, 5 of 2006.

Pollack, ME. Amd Horty, JF. (1999). There's more to life than making plans: Plan management in dynamic, multi-agent environments. AI Magazine, 20(4):71-84.

Sapena, O. and Onaindia, E. (2004). Domain independient anytime planning for dynamic environments. In: *Proceedings of CCIA.*

Shankar, P. and Fox, A. (2002). Sword: A developer toolkit for web service composition. In: *The Eleventh World Wide Web Conference*, Honolulu, HI, USA.

Sirivastava, B., Koelher, J. (2003). Web Service Composition, Current solutions and Open problems, In: *ICAPS 2003*, Trento, Italy.

UDDI Consortium. (2004). UDDI, Universal Discovery Description and Integration Protocol: http://uddi.org/pubs/uddi_v3.htm, last visit, April, 5 of 2006.

W3C Consortium. (2001). WSDL, Web Services Description Language: http://www.w3.org/TR/ wsdl, last visit, April, 10 of 2006.

W3C Consortium. (2003). SOAP, Simple Object Access Protocol http://www.w3.org/TR/soap12-part0/, last visit, April, 10 of 2006.

W3C Consortium. (2004). OWL http://www.w3.org/TR/owl-ref, last visit, April, 10 of 2006.

Warren, D. (1976). Generating conditional plans and programs. Proceedings of summer conference on Artificial Intelligence and Simulation on Behavior, pp 344-354.

Weld, DS. (1999). Recent advances in AI planning. In: AI Magazine, 20(2):93-123.

Wilkins, D. (1988). Practical Planning: Extending the Classical AI Planning Paradigm. Morgan Kaufmann.

Wu, D., Sirin, E., Hendler, J., Nau, D. and Parsia, B. (2003). Automatic Web services composition using SHOP2. In: *Workshop on planning for web Services*, Trento, Italia.