Reducing the impact of AI Planning on end users*

Luis Castillo [1] Juan Fdez-Olivares [1] Óscar García-Pérez [2] Francisco Palao [2] Tomás Garzón [2]

[1]Dpto. Ciencias de la Computación e I.A.
University of Granada, SPAIN
{L.Castillo,Faro}@decsai.ugr.es

[2]IActive Intelligent Solutions
{O.Garcia,F.Palao,T.Garzon}@iactive.es
http://www.iactive.es

Abstract

In this paper we analyze the lessons learned during three projects that are being developed between our research group and our spin-off IActive Intelligent Solutions. These projects are based on the deployment of AI planning technology in three different business environments with a different degree of maturity in ICT. In these three cases we will focus on how to extract domain and problem files and how to integrate action plans with existing information services on the end-user side in order to facilitate the integration of P&S technology with legacy software and end-users work environment.

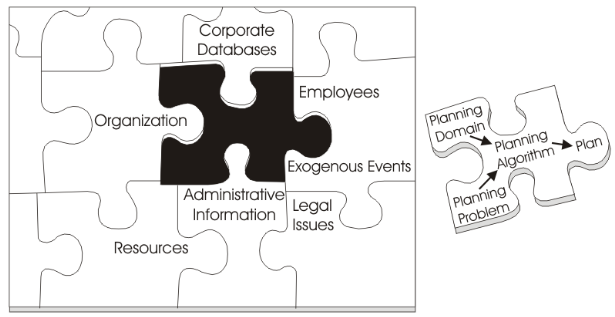

Figure 1: The enterprise-level (business-level) integration puzzle

Introduction

Artificial Intelligence Planning research community has always kept an eye out for applying its technology to real world problems and reach an industry level deployment comparable to that of other disciplines like neural networks, genetic algorithms or fuzzy logic, just to mention some of them. Despite being a relative old discipline, at least older than most of those mentioned before, planning technology has some very good examples of applications but it does not seem to be mature enough to lead an enterprise-wide deployment and to be part of the whole enterprise or business jigsaw puzzle.

There may seem to be many reasons behind this lack of success like the need for efficient planning algorithms, the need for enhanced underlying reasoning processes (uncertainty, time, resources, ...), the need to deal with exogenous events or sensing operations among others. These are some of the lines that are being pursued from the own research community to bridge this well known gap. Although all of these problems focus on the inner part of the planning piece of Figure 1, and this is an important effort to drive this technology forward, there are other important questions that should also be addressed with regards to insert the planning piece into the whole puzzle. Both categories of problems are very important, but papers that fall on the former type seem to dominate in mainstream conferences. Here, we intend to share some of the interfacing issues of the second type that we have learned from our past and current experience that started in our research group and is now being undertaken at our spin-off IActive Intelligent Solutions[1].

The main issue: knowledge integration

One of the most distinguishing features of planning technology is that it is a knowledge-intensive task, that is, planning a solution to a problem in a human-centered environment implies taking into account a lot of knowledge about the problem being solved. This knowledge comes from many different and, mostly, heterogeneous sources and it must be "prepared" somehow to the PDDL (Edelkamp & Hoffmann 2004) input gateway (or similar) to our state of the art planning engines in the form of problem and domain files. In addition to this, the solution plan must also be processed to fit into the structure of the whole puzzle. Therefore we must analyze how these inputs and outputs of our planning algorithms impact into the remaining components of the puzzle and try to reduce this impact as a way to ensure (part of) the success of the planning technology.

In our very modest point of view, the integration of AI planning technology into any existing enterprise or business environment must take into account the following points:

- Technology is an enabling factor of change, not the

*This work has been partially supported under the research contract NET033957/1 with the Andalusian Regional Ministry of Environment and the CICYT Project TIN2005-08945-C06-02

[1]http://www.iactive.es

change itself. That is, the integration of new technology should not produce a change for the environment to adapt to it, instead, technology should adapt to the existing business environment. This is very important because business stakeholders (decision makers, actors and managers) are not intended to know about planning domain languages, nor logical predicates nor any other formalism planning researchers are used to deal with. So there must be a mapping between the language that business stakeholders are used to deal and the language that our planners handle. This mapping is not only a matter of translation but it may also imply a hard work on knowledge extraction and validation.

- Technology must be as transparent as possible. Business environments are plenty of software for their daily work and we do not foresee the planning engine to be a standalone application by itself. Instead, we do foresee a planning engine as something as a "plug and play module" or a "helper application" for legacy software so end users may have access to planning technology from and to their everyday applications.

The combination of these two points is, therefore, an important issue to get planning technology integrated into the whole business puzzle. They are not exclusive of AI planning technology, but it is also shared with other mature AI technology. However, planning technology strongly depends on the representation of operational knowledge[2] and this knowledge is not easily available in a explicit form in these environments, so its integration appears to be more difficult. In a increasing number of cases, business stakeholders are starting to pay more attention to business process modeling and management so that they use different languages to represent their operative knowledge and to execute and monitor their business processes. In the best situation, extracting the operative knowledege required for the planner becomes easier and less costly. Easier because all the processes are already modeled and described by the own end user so a planning domain expert only needs to query these models and translate the required details into a planning domain description language like PDDL. And less costly because, if this translation process could be done automatically, any maintenance change on the business rules or processes made by the end users, are immediately translated into the planning domain without the intervention of planning domain experts.

In the forthcoming sections we discuss how the consideration of these points have affected to three different projects. In all of them there was a need to integrate AI planning techniques but none of their end users was supposed to know about these techniques nor artificial intelligence in general, so the integration of the planning piece in their corresponding business puzzle was a subtle issue.

[2]Operational knowledge must be understood as the knowledge about how to get things done, as opposite, or at least different, to descriptive knowledge about how things are.

First case of study: crisis management in Andalusia

In (Fdez-Olivares *et al.* 2006) we were (and in fact we still are) engaged in the application of planning techniques to help fire directors to define a forest fire fighting plan in Andalusia (Spain), it fully covers the stages of preparedness and response and it might be generalized to any other crisis situation. End users are fire directors, in charge of making the main decisions and driving all the operations during the episode, and technical and administrative staff in their chain of command, who are responsible of launching execution orders, monitoring the execution of the plan and updating administrative information.

The problem

The description of the initial state of the problem must contain an exhaustive description of all the fire fighting resources in Andalusia. This implies the representation of large amount of information about facilities (32), brigades (341), pumping vehicles (94), spraying helicopters (27), etc. The goal of the problem must be described in terms of the geographical deployment of operational areas and the target operations. The main issue here is that the planner needs this information and this information is being queried and updated daily by human operators as well, who are not intended to know anything about PDDL nor other planning formalisms. Obviously, there must be a common representation, accessible both by human users and by the planner with the following features:

- It must be rich enough to represent all the knowledge required for the planner.
- It must be easily accessible by human operators to query and update daily operations, with no training effort on any additional language.
- It must be easily accessible by the planner to extract the required knowledge into a PDDL problem file.

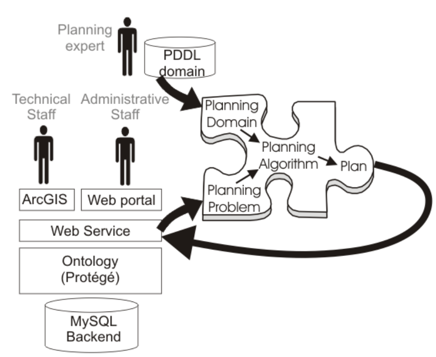

Figure 2: Integration of the first case on crisis situations

The representation chosen was an underlying ontology designed in Protégé[3] with a MySQL back-end to support an efficient and concurrent access to the information. On top of

[3]http://protege.stanford.edu

this ontology we built a web service with a clearly defined API. This web service may be used either from a web portal (so that administrative staff easily query and update the knowledge in the ontology through any web browser, see Figure 3) or from an existing GIS application[4] (so that technical staff may easily introduce the geographical layout and target areas of the goal, see Figure 4)

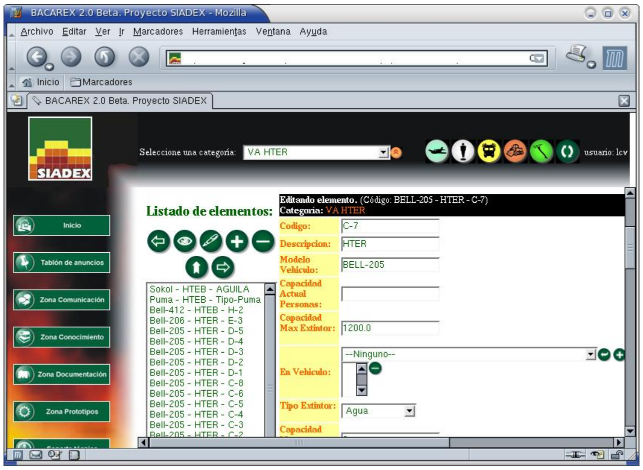

Figure 3: Access to the knowledge base through a web browser for administrative purposes

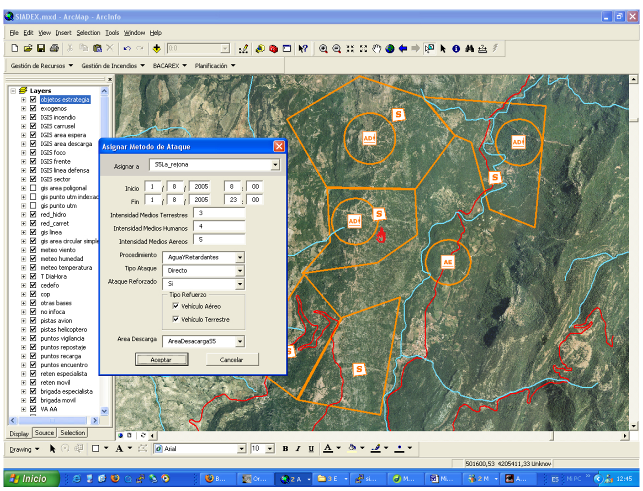

Figure 4: Access to the knowledge base through ArcMap for technical purposes

The domain

The current degree of maturity on ICT of the forest fire fighting service in Andalusia does not allow them to have formalized their fire fighting protocols in any well known process language like OWL-S or XPDL. Instead, they only have training and working documents that explain these protocols. Therefore, we decided to encode the domain by hand for our HTN planner, as an HTN extension of PDDL 2.2 (Castillo *et al.* 2006), and keep it away from administrative and technical staff so that only a planning expert may update and modify it. We were not happy with this decision because neither administrative or technical staff may have access to the fire fighting protocols encoding and thus, their independence (and therefore the transparency of planning technology) was severely reduced. This was mandatory since, currently, administrative and technical staff are not expected to be introduced in process description languages. However, Andalusian administration is introducing these languages gradually at all the levels of the administration and we foresee that, in a near future, forest fighting technical staff will be skilled enough to formalize and represent their fighting protocols in some of these languages and then, the adoption of our planning technology will be much more transparent, as it is shown in the next cases of study.

The plan

Finally, the planner has been integrated as an additional toolbar of the ArcGIS suite so that it is called just by clicking a button on their everyday desktop. The plan obtained by our planner is also introduced in the whole business puzzle of the technical and administrative staff. Our planner is able to obtain the plan in an enhanced XML format that includes, all the temporal constrains (direct and inferred constrains), binding of variables and annotations gathered during the search process. This allows us to translate this XML plan into other formalisms like a chronogram for the GIS platform, a MS Excel file, a Gantt chart or a proprietary format of the technical staff (Figure 5).

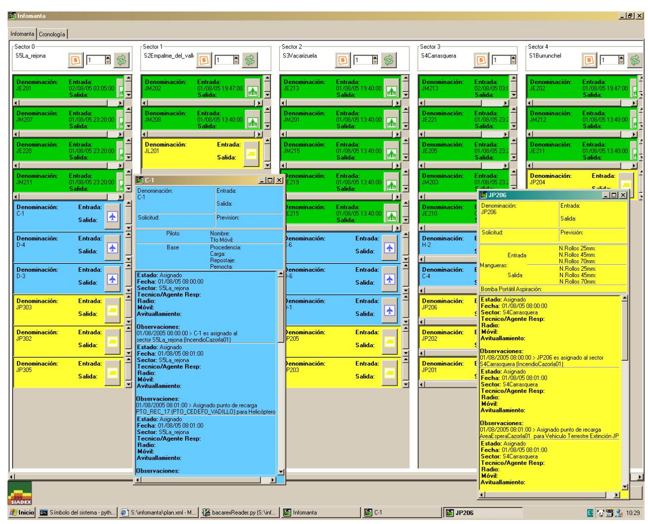

Figure 5: A proprietary format for plan visualization

As may be seen, introducing the plan into the legacy software infrastructure is the easiest task given our enhanced XML representation of the plan.

In summary, planning technology has been introduced silently, integrated with the regular working environment of the administrative and technical staff (web browsing, GIS

[4]ArcMap by ESRI http://www.esri.com

software, spreadsheets and Gantt editors). The unique drawback is that technical staff cannot modify the domain by themselves but through our intervention. This is a matter of the maturity of the forest fire fighting field and it is expected to change in the near future with the adoption of standard business process modeling languages.

Second case of study: learner centered design

This second case focus on the distance learning field, particularly in what is known as learner centered design. In this case, the introduction of planning technologies allows us to define customized learning paths for a given course. That is, the goal is the arrangement of the resources associated to the course taking into account the goals selected by the instructor and the own needs, features and constraints of every student, so that every student in the same course will have its own learning path to the goals.

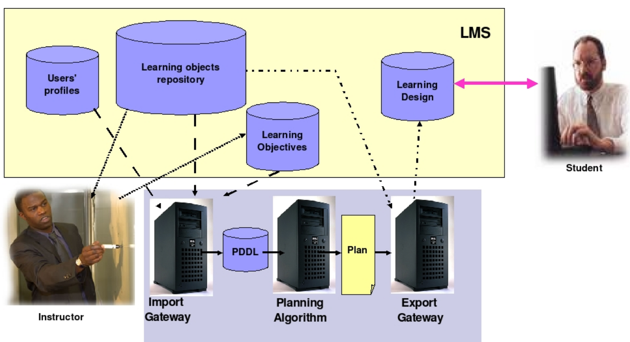

Figure 6: General view of a Learning Management System

A Learning Management System (LMS) is composed of several related databases (Figure 6):

- The learning objects repository contains all the educational resources (documents, videos, photographs, schemes, etc) that could be linked to make up a course. Every learning object is labeled by means of an extensive set of standard metadata (IMS-GLC 2007) so that the instructor may describe the main features of the learning object and its adequacy to different student profiles.
- The user profiles database contains extensive information about each student: personal data, preferences, learning style, academic history, his/her hardware/software platform and others. It follows the IMS-LIP standard (IMS-GLC 2007).
- The learning objectives are specified by the instructor for each course, so all the students of the same course are intended to reach the same goals.
- The learning design database contains a timed sequence of learning objects that each student must follow to reach the course' goals adapted to his/her own features. It follows the IMS-LD specification (IMS-GLC 2007).

The goal of a LMS is to serve as a framework for the definition of a course and for the student to follow that course in a distance learning setting.

The introduction of planning techniques in this environment may be described by the following steps (Figure 7:

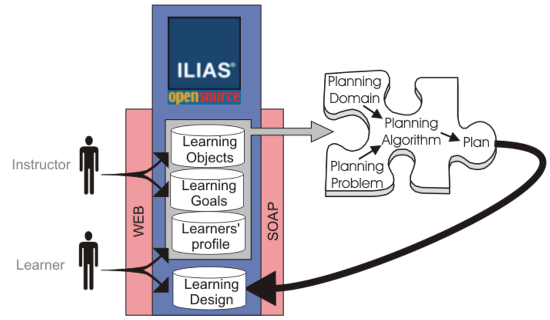

Figure 7: Integration of AI planning into the ILIAS Learning Management System

1. The learning objects repository is labeled using a extensive set of standard metadata, mainly a specific subset of metadata that encode the structure and dependence of the learning objects (for more details see (Castillo *et al.* 2007)).
2. (Dotted lines) The instructor explores the repository and define the learning objectives of a given course.
3. (Dashed lines) Our system explore the different databases of users profiles, learning objects and learning objectives and generate the necessary PDDL 2.2 (Edelkamp & Hoffmann 2004) files for our HTN planner to run. The planner is executed and a customized learning plan is obtained for every student registered at the same course.
4. (Dotted/dashed lines) The learning plan is translated into a form playable or understandable by the LMS, usually under the IMS-LD specification.
5. The plan is executed (or played) by the student to follow the course adapted to its own features and needs.

The problem and the domain

This case of study provides a more formal framework for inserting planning technology since the standards used for labeling metadata and user profiles provide a great amount of knowledge (Figure 8) that can be exploited to extract descriptive and operational knowledge for the planner. In particular in (Castillo *et al.* 2007) we show that this knowledge is rich enough so as to automatically extract the problem and the domain files for a HTN planner from a SOAP interface (W3C 2007) provided by the web services of the ILIAS LMS (ILIAS 2007), a well known platform for distance learning.

These metadata are introduced directly from the LMS (Figure 9) and they all belong to the standards commonly used in distance learning, so there is no additional impact on end users (instructors).

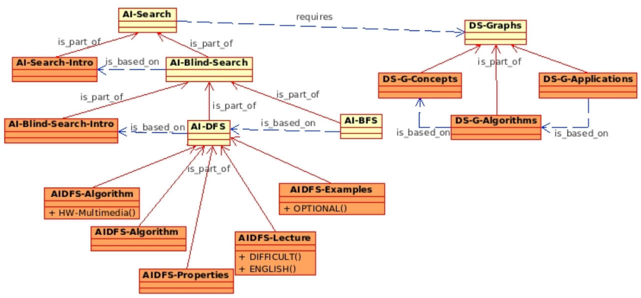

Figure 8: An exhaustive labeling of learning objects (compound objects in light shadow and simple objects in darker shadow) showing ordering, dependence and composition relations. It also shows that every simple object may be labeled with other features like its language, its hardware and software requirements, its degree of difficulty and its optionality amongst other

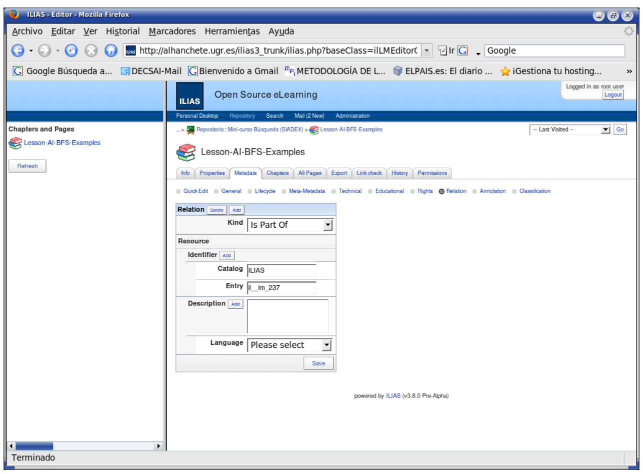

Figure 9: All the metadata and profiles are introduced through the standard web interface of the LMS

The plan

The HTN domain and problem files, which are automatically extracted from the LMS, are fed into the planner and a plan is obtained for every student registered in the same course. Although each plan might be different, all of them will allow students to reach the same goals, but taking into account the special features of each student. Finally, this plan is inserted back into the LMS to be played in the form of a IMS-LD compliant file.

In summary, the cost of introducing planning techniques in this business environment is dramatically reduced and the technology is fully transparent to end users (instructors). The effort made by the instructor to encode the metadata of the learning objects, something that can be considered usual in any LMS, is enough for obtaining the most subtle part of the planning piece: the problem and the domain files. Later on, the plan is easily inserted in the LMS platform with no additional cost. This means that, if end users are skilled on some high level formalism for their daily work and this formalism is able to encode some descriptive and/or operative knowledge useful for the planner, then we can extract problem and domain knowledge directly from these formalisms without having to depend on others (planning experts) nor having to learn a different formalism. The following case follows this argumentation and introduces a third case of study in which end users are skilled in a process description language.

Third case: semantic web services composition

Semantic web services techniques support the way in which already existing syntactic web services can be extended with a semantic layer in order to be automatically discovered, composed and invoked. The main goal of this third case of study is to provide a logical framework for an HTN planner to be capable of both interpreting SWS descriptions and, given a high level service request, reasoning about them in order to automatically compose and execute a sequence of web services that provides the service requested (see (Fdez-Olivares *et al.* 2007) for more details). There are several standard proposals for SWS but OWL-S (Martin *et al.* 2003) is a very good choice to this purpose for the following reasons. On the one hand, OWL-S process model allows to represent web services as processes with typed input/output parameters, preconditions and effects and a compositional hierarchy of atomic and compound processes. And, on the other hand, it is based on a data model built on top of an OWL ontology consisting of classes, properties and instances. Therefore, our goal, in this case, is translating the OWL-S process and data models into an HTN extension of PDDL 2.2 domain and problem files, call the planner and obtain a plan as a timed sequence of actions that could be used as an executable sequence of web services to give a response to the high level service request.

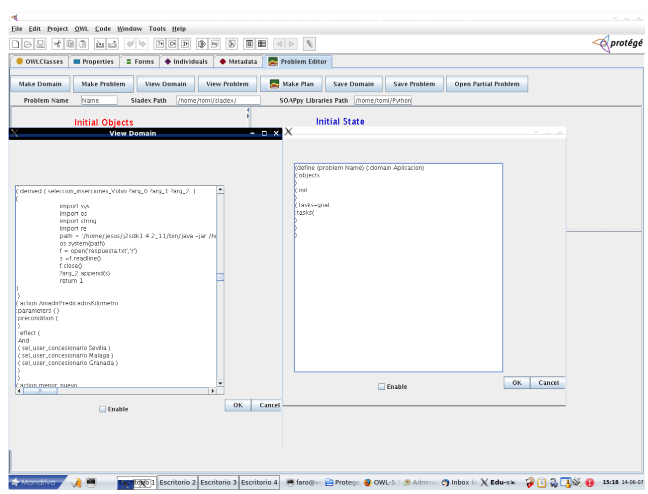

Figure 10: Our system has been embedded into an OWL and OWL-S editor environment as Protégé

The problem and the domain

The translation process first maps the OWL data model into the PDDL data model by translating OWL classes, properties and instances into PDDL types, predicates and objects, respectively.Then it maps the OWLS process model into an HTN domain that represents the operation of both atomic and composite processes as primitive tasks and task decomposition schemes, respectively. Atomic processes are mapped as PDDL durative actions and the workflow pattern of every composite process is mapped into a method-based task decomposition scheme that expresses the operational semantics of the control structures found in that composite process.

The plan

The planner makes use of the knowledge encoded in the domain (representing the OWL-S process model) as a guide to find a sequence of temporally annotated primitive actions that represents a suitable composition (with possibly parallel branches) of atomic processes. This sequence is sent to a Monitor module that is in charge of both scheduling the execution of atomic processes according to their temporal information and sending execution orders to an Executive module. This module is in charge of executing web services invocations and sending back the information.

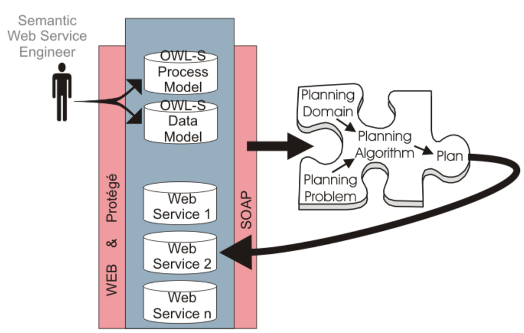

Figure 11: The integration of our HTN planner into a web service composition based business environment

In summary, this last example has also shown how a planning engine may be seamless integrated into environments with a strong underlying formalization of processes. In this case, it is a web service composition based environment and domain experts are supposed to have skills on process design languages like OWL-S (in the case of other languages, the procedure would be similar). The point is that their business models written in OWL-S are rich enough so as to extract valid PDDL domain and problem files, so the introduction of planning technology is fully transparent for these end users and it may be fully embedded into their existing working environments.

The way forward

After these three cases, we have learned two important lessons that might be considered complementary. On the one hand, there is a common issue about the **integration** of AI planning technology into existing business environments, what imply the need to share the information between end users and the planner. AI planning needs much knowledge and most part of it is dynamic, it depends on end-users databases and not only the user but also the planner may modify this knowledge, so there must be a common representation of the knowledge or, at least, a gateway to get the knowledge from and to the available sources. No one will accept to replicate their data or re-type it by hand as the input to the planner. The planner must adapt to the existing structure of the data and get what it needs wherever it is. In most cases, the sources of knowledge are very heterogeneous since end users may have its information distributed on different platforms, operating systems or database systems. This also implies that the target environment must be either based on, or ready to adopt, a service oriented architecture (**SOA**) based on the extensive use of web services, in order to enable this **interoperability** of different platforms and to grant access to the whole set of data available in the enterprise. In the case that the target environment is not adapted to a SOA, the planning module might be a catalyst for the introduction of such technologies since it is the main interested part in having a common access to the whole enterprise data (Fdez-Olivares *et al.* 2006).

On the other hand, and very related to this, it is the question about the level of automation of the workflow management at the target business environment. In the case that this business environment has automated (or is automating) its operational processes within what is known as **Business Process Management** (BPM) (Fischer 2007) that takes into account resources, employees, applications, documents and the own organization, the application of AI planning technology seems to be less costly. It would also be less independent of third party planning experts, since all of the knowledge needed to encode domain and problems for the planner may be extracted from their BPM suites (like in cases 2 and 3 before). This is becoming particularly good since most relevant enterprises are currently engaged in a process of automation of their operational business processes by using standard languages like OWL-S but mostly XPDL and BPEL (Fischer 2007) and there is a clear mapping between these languages and our planning domain description languages like PDDL either for plain or HTN domains (Castillo *et al.* 2007; Fdez-Olivares *et al.* 2007; Sirin *et al.* 2004). Since BPM suites integrate business analysts, technical developers and business managers, they can modify their business rules by themselves and the planning domain will be automatically updated, without the intervention of external planning experts and thus, increasing the transparency of this technology. Even more, these BPM suites are strongly based on the use of an underlying SOA, what provides the best context for a seamless and deep integration of AI planning technology. In the case that the target environment is not aware of this BPM technologies, the implantation of AI planning would be more difficult since business experts will still depend on planning experts to modify planning domains as soon as their business rules change (like the first case in this paper).

In any case it is worth to recall that these issues of transparency and integration of AI planning technology are a key factor to implement and deploy our technology, that is, we must seamlessly integrate with end users data, but also with their business rules, so that we do not induce a change on the enterprise before the adoption of our technology but reduce the impact that this technology might have after adopting our technology.

References

Castillo, L.; Fdez-Olivares, J.; García-Pérez, O.; and Palao, F. 2006. Efficiently handling temporal knowledge in an HTN planner. In *Sixteenth International Conference on Automated Planning and Scheduling, ICAPS*.

Castillo, L.; Reynaga, L.; González-Ferrer, A.; Fdez-Olivares, J.; and García-Pérez, O. 2007. Knowledge engineering and planning for the automated synthesis of customized learning designs. In *Conference of the Spanish Association for Artificial Intelligence*.

Edelkamp, S., and Hoffmann, J. 2004. The language for the 2004 international planning competition. http://ls5-www.cs.uni-dortmund.de/ edelkamp/ipc-4/pddl.html.

Fdez-Olivares, J.; Castillo, L.; García-Pérez, O.; and Palao, F. 2006. Bringing users and planning technology together. experiences in SIADEX. In *Sixteenth International Conference on Automated Planning and Scheduling, ICAPS*.

Fdez-Olivares, J.; Garzón, T.; Castillo, L.; García-Pérez, O.; and Palao, F. 2007. A middleware for the automated composition and invocation of semantic web services based on htn planning techniques. In *Conference of the Spanish Association for Artificial Intelligence*.

Fischer, L. 2007. *2007 BPM & Workflow Handbook*. Workflow Management Coalition.

ILIAS. 2007. ILIAS Learning Management System website. `http://www.ilias.de/ios/index-e.html`.

IMS-GLC. 2007. IMS Global Learning Consortium. `http://www.imsglobal.org/`.

Martin, D.; Burstein, M.; Lassila, O.; Paolucci, M.; Payne, T.; and McIlraith, S. 2003. Describing web services using owl-s and wsdl. http://www.daml.org/services/owl-s/1.0/owl-s-wsdl.html. DAML-S Coalition working document.

Sirin, E.; Parsia, B.; Wu, D.; Hendler, J.; and Nau, D. 2004. Htn planning for web service composition using shop2. *Journal of Web Semantics* 1(4).

W3C. 2007. Soap version 1.2. `http://www.w3.org/TR/soap12-part1`.