Explanation of Opportunities

Pinar Öztürk[1], Hector Munoz-Avila[2], Agnar Aamodt[1]
[1]Department of Computer and Information Science
Norwegian University of Science and Technology, N-7034 Trondheim, Norway
{pinar, agnar}@idi.ntnu.no
[2]Department of Computer Science & Engineering
Lehigh University, Bethlehem, PA 18015-3084 USA
munoz@cse.lehigh.edu

Abstract. This paper discusses the need to recognize and take advantage of arising opportunities when an agent is autonomously interacting with an environment. We discuss the tasks, methods, model framework, and use it to analyze opportunities across several dimensions: when changing the tasks versus changing the methods, when having a complete knowledge about the methods in the domain versus having partial knowledge, when having a complete model of the domain versus having an incomplete model, when tackling a single task versus multiple tasks and finally we discuss the role of causal explanations in opportunistic case-based decision making.

1. Introduction

One of the holy grails in artificial intelligence research are systems capable of explaining their own behavior (Chien, 1987; Ram & Cox, 1992). From early research in AI systems (Newell&Simon, 1971) to most recent research on online agents explaining their recommendations (McSherry, 2005), the capability of a system to explain its decisions has been a recurrent research topic.

We identify two kinds of explanations: explanations for the system itself and explanations for the user interacting with the system. The latter is typical in systems such as mixed-initiative systems (Ferguson & Allen, 2007) where the user and the system are jointly performing a task and bi-directional communication is needed to coordinate their efforts. Such explanations require not only the system to infer the actual explanations but provide the HCI capabilities to present these explanations in user-understandable form.

Explanations for the system itself are used as part of an introspective reasoning process whereby the agent is acting in an environment and reasoning on the consequences of its actions to determine subsequent actions. Although explanations for a user interacting with the system brings further complexities into consideration (e.g., HCI considerations), it is understood that the capability of a system to explain to itself is a reasonable pre-requisite for systems whose goal is to present explanations to the user (Sørmo et al., 2005). In this work we want to focus on explanations for the system itself for CBR agents (e.g., Veloso et al., 1997).

One type of internal explanations a system produces for itself is triggered by an expectation failure (Hammond, 1986; Cox, 2007; Munoz-Avila et al., 2010). The basic premise is that the system has expectations of the outcomes for actions or queries made in the environment and when these expectations are not met then an expectation failure occurs and the system must explain these discrepancies. For example, in a computer game an agent has planned to follow a trajectory to reach a valuable item. If, after following the trajectory, the item is not found, the robot must explain this discrepancy in order to determine what to do next. For instance, the agent might observe an enemy agent leaving the zone and conjecture that this agent has taken the valuable item. The other type of explanation may be linked to opportunistic planning (Marks et al., 1989) where a current situation evokes a reminding to an alternative goal than the one being pursued, and the agent needs to consider if this sudden opportunity should suppress the current goal. An often used example is that an agent following a plan to go from A to B in a city suddenly observes that a bakery along the route now is offering fresh and cheap bread. The agent may consider deviating from its plan to stop and buy bread. The explanation for this is that eating bread is a frequent activity of the agent's master, and not spending more money than necessary is a persistent constraint. After buying bread the original goal may be pursued again, or not. In the computer game example the agent may be following its route when it suddenly encounters a gap in a nearby forest that would allow it to reach the valuable item faster. In many realistic domains the agents are acting in partially-observable environments. Hence, when the agent planned the original trajectory, it had incomplete information about the geography of the environment. Now it finds the gap in the forest and can consider not continuing with its planned trajectory and instead going through the gap. The explanation for this is that the agent has observed that this will be a shortcut towards the valuable item. It further reasons that it will save time and make it less likely that an opponent agent will reach the item first and hence it is a desirable action to do.

These examples contain the main elements we are interested in in this work: explanations of opportunities for an agent's reasoning where the agent has predefined goals (e.g., reach position A) but also has more general goals that can be formalized as objective functions (e.g., minimize execution time, minimize energy consumption, refill energy source). These functions can be conflicting, for example saving time might require the use of more energy; such as the robot having to go upwards to faster reach a location but consuming more energy in the process.

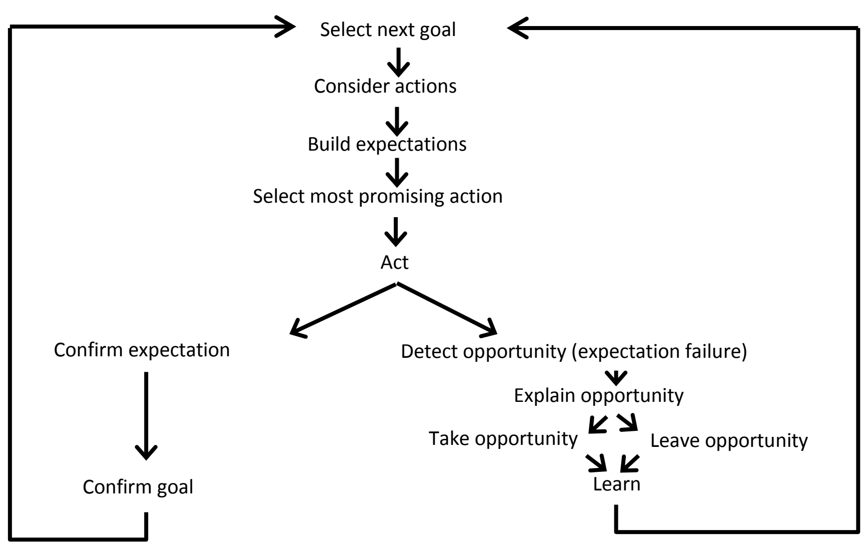

Figure 1: Agent tasks

Figure 1 illustrates the high level task structure of our planning agent. Given a current goal/subgoal the agent plans the next step, which can be simply to select the next action to execute or something more complex such as running a simulator and project what an opposing agent might do from the current state. An action is linked to an expectation about the result of the action. Reaching a goal state is one type of expectation, and there may be others. After having triggered the action the agent will normally complete the action, check the expectations and confirm that the goal has been achieved. If the expectation is not confirmed we have an expectation discrepancy – referred to as an expectation failure. There are several types of expectation failures, but the one we are interested in here – and the one illustrated in the figure – is when an opportunity is detected. This may occur at the point of reaching the expected goal state, or along the way to reaching it. The agent needs to explain whether to take the opportunity or leave it and continue as planned. In any case the agent learns from this incident. Our focus in this study is on the explanation processes that support the decision making; hence the "Explain opportunity" and "Learn" tasks will be given particular attention.

2. Model, Methods, Tasks (Goals)

In order to analyze the types of knowledge involved in agent behavior and related explanations, we adopt the knowledge level (KL) framework as originally defined by Allan Newell (Newell, 1982). This allows us to investigate the internal reasons that trigger opportunities. In the KL framework, a system is described at the knowledge level as an intelligent agent with its own goals and with knowledge of how to achieve these goals. The principle of rationality states that an agent will always use its knowledge in a way that ensures the achievement of its goals - provided the agent has the knowledge needed. The knowledge level enables a system (existing or anticipated) to be described in terms of what it does and why it wants to do it, completely independent of implementation constraints. Hence it is a level applicable to any agent to which it makes sense to ascribe knowledge and rationality.

The problem with using the knowledge level in the original Newell sense for modelling and design of computer systems is that it has no a priori structure. Hence, the knowledge level in this sense cannot be used directly to analyze and structure knowledge. Further, the principle of rationality assumes an ideal rational agent, not bounded by physical or temporal constraints. This has led to modifications of the original knowledge level notion defined by Newell, into a more operational notion of the knowledge level. This may be viewed as moving the knowledge level slightly in the direction of the symbol level, i.e., the level of data structures and programs lying below the knowledge level. Various authors have proposed systems analysis and design methodologies based on a structuring of the knowledge level, as exemplified by the CommonKADS methodology (Breuker & Van de Velde, 1994), the Generic Tasks approach (Chandrasekaran, 1992), the Method-to-Task approach underlying the PROTEGE systems (Musen, 1989), and the Components of Expertise methodology (CoE) of Steels (Steels, 1990). The latter is the more flexible of the methodologies, and this suits our needs best, which is why we have adopted the terminology and modelling philosophy of CoE in our analysis.

Given this refined notion of the knowledge level, a consensus was established within the knowledge acquisition and modelling community that knowledge, in its most general sense, could meaningfully be structured into three main categories - or viewed from three different perspectives: Task knowledge, Method knowledge, and Model knowledge (see Figure 2). Task knowledge describes what to do. Tasks are tightly connected to goals, and sometimes used interchangeably. A task is defined by the goals that a system tries to achieve. Method knowledge describes how to do it, i.e. a method is a means to accomplish a task (e.g. to solve a problem). A reasoning method is a typical example. Model knowledge is knowledge about the world that a method needs to accomplish its task. Examples of model knowledge are facts (i.e., what in planning is referred to as the state of the world), heuristics, causal relationships, multi-relational models, and – of course – specific cases. The term 'model' may be confusing, given that task and method knowledge can be viewed as models as well. As mentioned, we borrow this terminology from the CoE framework. Other terms used for this knowledge perspective include ‘domain knowledge’, 'object knowledge', and 'application knowledge'.

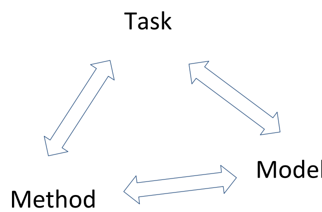

Figure 2: Knowledge perspectives

In CoE, the three knowledge perspectives are not, at the outset, assumed to be applied in any particular order when designing systems. Any sequential combinations of them may in principle be used. Starting from a task perspective, for example, the building of a task hierarchy can be followed by designing the appropriate methods for each task, followed by the knowledge models needed by each method. Starting from a model perspective would analyze what type of knowledge characterizes the domain in question (e.g. degree of uncertainty, complexity of objects), which would point to suitable methods for processing the knowledge, and in turn suitable tasks to be handled by those methods.

In our context, and at the highest level, a task may be viewed as a knowledge unit that contains one or more goals that need to be achieved in order to accomplish the task, and where a goal may be achieved through the execution of one or more plans. A plan hence serves as a method for achieving a task’s goal, for which it relies on one or more domain models.

3. Opportunities and partial versus complete knowledge

The ability to utilize opportunities is envisaged to be one of the important characteristics of intelligent agents. Opportunities can be defined as conditions where the current goal can be suspended or paused in order to pursue another rewarding one, or the planned course of action can be changed to increase the utility of the agent. Opportunistic behavior has attracted researchers in artificial intelligence over the years. For example, in Hayes-Roth & Hayes-Roth (1979), a cognitive model of opportunistic behavior is proposed. In Birnbaum (1986), three important issues related to opportunistic behavior are defined: recognition of the opportunities, ability to suspend/modify the current goals, and decision of whether to pursue the opportunity or not under current circumstances. In Birnbaum & Collins (1984) and Simina & Kolodner (1995) the focus is on opportunity recognition, while in Ram & Hunter (1992) both opportunity recognition and the suspension of the dominant goal to pursue a new one in the context of story understanding are dealt with.

We can distinguish between two main sources of opportunities: (1) those caused by the agent’s incomplete knowledge and (2) those caused by factors exogenous to the action.

The internal sources are due to the imperfectness of the agent’s believes/knowledge about the external world or the uncertainty about its own reasoning knowledge. The agent might know the actions it can take but it might not know (1) the consequences of some of these actions, (2) when to prefer an action over another one, and (3) what are the best combinations of actions to achieve its goals. The external sources of opportunity on the other hand relates to the observability or dynamicity of the environment. In the next sections we elaborate on these two dimensions of opportunities.

As an example of an opportunity resulting from the agent's incomplete knowledge, the agent might execute an action to harvest a goldmine. Unknown to the agent, harvesting a goldmine results in the agent encountering gems hidden in the goldmine. The agent was not planning to seek gems but it knows they are very valuable and acquiring resources is one of its over-arching objective functions. After harvesting goldmines multiple times, the agent will learn to find gems by harvesting gold. When this happens, this is no longer an opportunity as it will become part of its domain knowledge.

As an example of an opportunity resulting from factors exogenous to the agent's actions, the agent encounters a gem while harvesting, but this time, the game has different rules. Gems are not usually hidden between gold. This opportunity arose because early on, and unknown to the player, a dwarf harvest gold and drop the gem. Unlike the previous case, this is an exogenous event, and not the result of incomplete knowledge.

Case-based learning is a natural method to identify and learn new knowledge when opportunities arise as a result of incomplete or imperfect knowledge. Each opportunity could be stored as a (problem,solution) case. Typically, the problem is annotated with the conditions in the state of the world and the actions performed (e.g., harvesting the goldmine) and the solution is the opportunity that arose (e.g., find a gem and picked it up). As more cases are captured with same or similar solutions an abduction process can be performed in the problem part of the case to find the common general conditions (if any) that makes the solution true. A case-based approach to explanation that is relevant to our research, and with the same assumptions about partial observations and imperfect knowledge, is the model of explanation patterns, e.g. as implemented in the AQUA system (Ram, 1994).

4. Changing Goals versus Changing the Method

As described in the previous section, each goal in a task structure may be achieved through one of a set of alternative methods, typically through the execution of a plan. From the knowledge representation perspective, a task can be represented as a frame where the goal and the method are represented as slots. If the goal is a high-level one, then the method of the task may not be directly executable, but rather the task is decomposed into a number of subtasks each having its own subgoal of the higher level goal. When there is more than one alternative method to achieve the same goal, then the method attribute of the task will have all the alternative methods as its value.

Opportunism, therefore, may be either about agents adopting another goal than the currently pursued one, or adopting a new method or trajectory towards the current goal. An example of the former type is that the agent may pass by a goldmine on its way to the objective when it remembers the general goal of harvesting gold in order to make castle improvements. An example of the latter is in the forest gap scenario we discussed earlier, in which the agent notices a forest gap which gives higher payoff in reaching the goal state (e.g., will reach it in less time thereby making it less likely opponent will reach valuable item). We may consider both as instances of revising the plans and re-planning. In both cases the agent is not blindly committed to his current set of goals and methods.

It is natural that there may be alternative methods for achieving a goal, but not all of them would be possible under all situations. In particular, at the immediately executable level, there will be constraints, particularly when actions are performed in the physical world. Suppose a certain action involving the use of a particular tool. If this tool is not available, then a rational agent will not choose a method that relies on this action, which is fine. However, in dynamic environments some constraints that were met at a moment may not be met later on. This enforces the agent to change its current method to another one that fits the new situation. Such expectation failures are rather typical in fast changing environments, and the system needs to explain the reasons for the deviation from the anticipated consequences of an action. This explanation will justify the reason for the method change, in terms of "dynamic constraint violation". If the failure is compensated by noticing a new or unexpected tool that can be substituted with the planned one, the agents should go for it, justified by an explanation for the change of method.

A similar, albeit different reason for change of action course occurs in partially observable domains. It may not be possible to check in advance whether all constraints of an action/method are satisfied in environments that are only partially observable. The agent may select and start a plausible action and observe on the way if the committed course of action is possible. If it is impelled to make a change in its behavior because of recently perceived conditions violating the constraint of the current method, the agent should explain this violation by attributing to the partial observability of this constraint.

5. Partial Model of the State

In many domains the agent interacts in a situation where the agent can only see part of the environment. This includes situations such as a robot gathering information from its mounted sensors, which have limited reach. In digital games, frequently, games include rules limiting the view of the environment. For example, in first-person shooters a player only views what is within visual range of its avatar. In a real-time strategy game, players can only view areas that are within reach of units that the player controls; the player has no information of the other areas. This is referred to as the fog of war and it intends to simulate what commanders experience when controlling armies in real-life conflicts.

Partial observability is an orthogonal issue to whether the agent has a complete model of the environment. Even if an agent has a complete model of the domain, partial observability might introduce uncertainty resulting in opportunities; an agent might know all conditions and effects of an action but because of the partial observability it might not be able to determine if there is a discrepancy between the intended effects of the action and the actual outcome. For example, an action to bombard some objective has the expectation of the objective to be destroyed. But because of partial observability, the agent doesn't realize that the objective was moved of location and, hence, remains intact after the bombardment.

Partial observability might also result in unforeseen opportunities for the agent to seek new goals. An agent might pursue to gather gold by harvesting from a gold mine that is heavily defended by an opponent. While approaching the mine with the purpose of taking it, it might encounter a lightly defended opponent's truck that is transporting gold from the mine. At this point the agent can generate a new goal to attack the transport to capture the gold. This opportunity only arises as a result of the opponent's truck getting into the view range of the agent's units. If the world is fully observable, such opportunities wouldn't arise. The agent could simply plan to reach the objectives and only add new ones as a result of execution failures.

6. Single versus Multiple Goals

As human beings, when pursuing a specific goal, we are usually aware of other goals we may be pursuing, or should be pursuing. These other goals may be vague goals residing in the background, or more acute ones. While psychologists for many years were concerned about behavior driven by single goals, recent research has focused more strongly on the processes of dealing with the number of unfinished goals, often conflicting, that are hanging over us. Researchers have discovered gaps in the understanding of multiple-goal behavior, and particularly in how we integrate behavior directed at a singular goal with the decision-making involved in selecting between constantly pending goals. Proposals for integrating goal-directed and decision-making processes include several theories of motivation, including those related to expectations, needs, and value of the outcome, where a particular challenge is related to dynamically changing motivations, expectations, and needs (Vancouver et al., 2010).

To recap our setting (Figure 1), we have an agent executing a plan that will accomplish a goal. The plan consists of several plan steps in sequence, and in each plan step the agent is executing an action that is intended to accomplish a goal. So, we may imagine that there is a backbone of goals on the agent's stack that are linked to plan steps in the original plan. Let us also imagine a goal pool of sleeping goals in the background, from which one or more goals at a particular point in time can be awakened up by the occurrence of an unexpected opportunity. Given that we are interested in whether to pursue the original goal or to jump onto the just awakened goal, we will not be dealing with multiple original goals or multiple opportunity goals. Our multiple-goal problem then boils down to the process of being reminded of an opportunity goal, and selecting between the original and the opportunity goal in a case-based reasoning setting.

An early model that deals with this problem from a CBR perspective is the opportunistic memory model of the TRUCKER system (Hammon et al., 1993). TRUCKER simulates the planning of pick-up and delivery of parcels from a set of trucks within an area. A set of goals are put on an agenda, and if an opportunity is detected during plan execution that will satisfy a goal later in the agenda, it will try to merge that goal with the current goal instead of treating the two goals independently. The new plan is stored in the case base indexed by both goals, which then constitute a goal pair. When encountering any of these goals later, the system will search the agenda for the other goal, and check for the opportunity to merge them again. Experiments showed that learning to handle multiple goals in this way improved system performance. In PARETO (Pryor & Collins, 1994), another early system in the trucking domain, the planning task is for a single truck to pick up and deliver tools and other equipment between construction sites and stores. PARETO deals with the multiple-goals problem by anticipating that faulty assumptions – i.e. expectation failures – will occur. It is able to react quickly and flexibly to unexpected situations through a general mechanism that constantly look for unforeseen opportunities, check for possible dependences and conflicts with the existing plan, and apply a cost-benefit function to evaluate the alternatives, which also includes an estimate of how easily is to achieve an alternative goal.

More recently, Vancouver et al. (2010) discussed a cognitive model of how humans deal with multiple goals. Goal selection is related to motivation and self-regulation, and specific agent-oriented models are described to capture the various aspects of self-regulation, expectancy, time considerations, and decision making for choosing the next task.

Our approach to multiple goals handling takes inspiration from these and other earlier results. Initially we suggest to distinguish between the main task at a certain point in time, with its associated goal, and the plan to achieve the goal, on the one hand, and a set of tasks and related goals – in a kind of task/goal pool – on the other. The main task may be changed during plan execution. Tasks and goals form a network structure of explanation relations, such as causality and dependency. This is ongoing work.

7. Decision making with Opportunity-driven Explanations

The question of what an explanation is, and particularly what constitutes a good explanation, is much debated but still open question. Within the philosophy of science as well as AI there have been numerous discussions over various views to explanation and explanation theories (see for example the overview paper in the Internet Encyclopedia of Philosophy (Mayes, 2001)). Explanation is often associated with causation, in the physical sense of following natural laws and that things being explained exist in the real world. However, there are also theories of explanation in which entities that are part of an explanation do not necessarily exist in a literal sense, but are entities useful for organizing human experiences, to facilitate the construction of mental models, or to communicate understanding between individuals. In AI in general the type of explanations associated with abductive reasoning has a strong position, and in influential work in CBR as well (Leake, 1995). Viewing the abductive inference step as an "inference to the best explanation" (Harman, 1965) put some pragmatic constraints to the role of an explanation as well as to the explanation itself, which should be useful for the type of explanations of interest in our context.

Explanations of opportunity-based decisions revolve around the conditions and reasons that lead to appearance of opportunities, how they relate to an agent's pool of goals and general motivations, and why pursuing these were rewarding (with a reference to the parameters of the objective function) or alternatively why the agent decided not to depart from the current goal. An account of explanation we are studying is the combination of inference to the best explanation with uncertainty, as given by partial observation of the environment or partial knowledge of the domain. This is in line with what Leake refers to as plausible explanations, also guided by their usefulness in a particular situation. A theory of explanation that takes this into account introduces probabilistic conditioning, in the Bayesian sense, into the explanation framework, and hence extends the notion of inference to the best explanation to also account for unknown phenomena (Okasha, 2000).

We are studying ways to store explanations or generalized patterns as part of an agent's opportunistic planning case base. The explanations may inform the agent's decision making in various ways. One is about the goal commitment, which has thoroughly been studied in the context of BDI agents (Kinny and Georgeff, 1991). Although that work investigates whether to abandon the current goal while our study also include a pause in the execution of the current goal, both study the selection of goals or tasks for immediate pursuit. Another difference is that the BDI agents deal with general algorithms for goal commitment, while we intend to learn the decision explanations situated in concrete opportunities. The cases will capture knowledge about the situations that gave rise to goal opportunities in the past. It may be just one-time event or there may be some causal dependency between the situation and the appearance of an opportunity. In the latter case there will be several cases with similar explanations. Such knowledge is useful when the agent selects its next goal. It is recognized that, if possible, an agent makes a plan that serves more than one goal at the same time. Goal-opportunities, although not certain, can be taken into consideration when making plan decisions. Similarly, when selecting the method/action to achieve the decided goal, knowledge about the precondition-related opportunities, such as different tools needed for a method, can be used when evaluating the alternative methods. If one of the methods can be executed using either of two alternative tools, and the tool opportunities are possible, then the agent will tend to select this method.

Our focus is on causal explanations, in the wide sense of causation, to justify an agent's choice of goal – whether it is to maintain the existing goal or jump on a goal associated with a sudden opportunity. As for the particular structure of explanations as well as the case structure itself, having the explanations in the solution parts of the cases is our current line of investigation.

8. Final Remarks

In this paper we have presented initial steps towards a framework for explanation of opportunities particularly tailored towards CBR agents. We have explored issues with taking advantages of opportunities when interacting with the environment. Using the tasks, methods, model framework as an umbrella for our discussion, we analyzed how reasoning with opportunities arises when changing the tasks versus changing the methods, when having a complete knowledge about the methods in the domain versus having partial knowledge, when having a complete model of the domain versus having an incomplete model, and when tackling a single task versus multiple tasks. We also analyzed how causal explanations relate to opportunities.

Acknowledgements. This work is supported in part by NSF grant 1217888.

References

Birnbaum, L. & Collins, G. (1984). Opportunistic Planning and Freudian Slips. Proceedings of the Sixth Conference of the Cognitive Science Society. Lawrence Erlbaum Associates.

Breuker, J.A., Van de Velde, W.: CommonKADS Library for expertise modelling. IOS Press (1994)

Chien, Steve A. Extending Explanation-Based Learning: Failure-Driven Schema Refinement. No. UILU-ENG-87-2203. University of Illinois at Urbana-Champaign, 1987.

B. Chandrasekaran: Generic Tasks in Knowledge-Based Reasoning; High-Level Building Blocks for Expert System Design. *IEEE Expert* 1(3) (1986) 23-30.

Ferguson, G., & Allen, J. (2007). Mixed-initiative systems for collaborative problem solving. AI magazine, 28(2), 23. Chicago

Hayes-Roth, B. & Hayes-Roth, F. (1979). A cognitive model of planning. Cognitive Science 3 (pp. 275–310).

Hammond, K., Converse, T., Marks, M.: Opportunism and learning. *Machine Learning* 10 (1993) 279-309.

Hammond, K. J. (1986). Learning to anticipate and avoid planning problems through the explanation of failures. Intelligence, 5(7), 8.

Harman, G.H.: The inference to the best explanation, *Philosophical Review* 74 (1) (1965) 88-95.

Kinny, D. and Georgeff, M. (1991) Commitment and effectiveness of situated agents. Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, Sydney, Australia, 82-88

Leake, D.B: Abduction, Experience, and Goals: A Model of Everyday Abductive Explanation. *The Journal of Experimental and Theoretical Artificial Intelligence*. 7 (1995) 407-428.

Marks, M., Hammond, K., & Converse, T. (1989). Planning in an Open World: A Pluralistic Approach. Proceedings of the Eleventh Cognitive Science Society, Ann Arbor, MI.

McSherry, D. (2005). Explanation in recommender systems. Artificial Intelligence Review, 24(2), 179-197.

Mayes, G.R.: Theories of explanation. *Internet Encyclopedia of Philosophy* (2001) http://www.iep.utm.edu/explanat/.

Munoz-Avila, H., Aha, D.W., Jaidee, U., Klenk, M., & Molineaux, M. (2010). Applying goal directed autonomy to a team shooter game. Proceedings of the Twenty-Third Florida Artificial Intelligence Research Society Conference. Daytona Beach, FL: AAAI Press.

Musen, M.A.: Conceptual models of interactive knowledge acquisition tools. *Artificial Intelligence* 1(1) (1989) 73-88.

Newell A.: The knowledge level, Artificial Intelligence, 18 (1982) 87-127.

Newell, A., & Simon, H. A. Computer simulation of human thinking. Rand Corporation. (1961)

Okasha, S.: Van Fraasen's critique of Inference to the best explanation. *Stud. Hist. Phil. Sci.* 31 (4) (2000) 691-710.

Patalano, A.L, Seifert, C.M.: Opportunistic planning; being reminded of pending goals. *Cognitive Psychology* 34 (1997) 1-36.

Pryor, L., Colling, G.: Opportunities; a unifying framework for planning and execution. *AIPS 1994 Proceedings.* AAAI (1994) 329-334.

Ram, A.: AQUA; questions that drive the explanation process. In: Schank, R.C.: Kass, A., Riesbeck. C.K.: Inside case-based explanations. Ch. 7 (1994), Psychology Press, 207-261.

Ram, A., & Cox, M. T. (1992). Introspective reasoning using meta-explanations for multistrategy learning.

Simina, M. D and Kolodner, J. L. (1995). Opportunistic reasoning: A design perspective. Seventh Annual Cognitive Science Conference, Pittsburg, PA, July 1995, 78-83

Steels, L.: Components of expertise. AI Magazine, 11(2) (1990) 29-49.

Sørmo, F., Cassens, J., Aamodt, A.: Explanation in Case-Based Reasoning; Perspectives and Goals . *Artificial Intelligence Review*. 24 (2) (2005) 109-143.

Vancouver, J.B, Weinhardt, J.M, Schmidt, A.M.: A formal, computational theory of multiple-goal pursuit; integrating goal-choice and goal-striving processes. *Journal of Applied Psychology* 95 (6) (2010) 985-1008.

Veloso, M. M., Mulvehill, A. M., & Cox, M. T. (1997, July). Rationale-supported mixed-initiative case-based planning. In AAAI/IAAI (pp. 1072-1077).