Enhancing Information Acquisition in Game Agents

Sebastian Welsh
University of Technology, Sydney
Sebastian.J.Welsh@alumni.uts.edu.au

Yusuf Pisan
FIT, University of Technology, Sydney
PO Box 123, Broadway NSW 2007, Australia
ypisan@it.uts.edu.au

Abstract: *Significant enhancements in the capabilities of software agents can result through improving how they acquire information. Decision making depends on getting the right information, but the issue of what actually constitutes the right information is complex. This paper outlines important characteristics of information acquisition in agents and suggests how to improve the effectiveness of information acquisition in agents in virtual worlds. By taking an affordance oriented approach it is possible to identify information resources that are efficient, reusable and well matched to the capabilities of game agents.*

Keywords: Intelligent agents, Reasoning strategies, AI architecture, Computer Game/Simulation Environments

1. Introduction

It is pretty much axiomatic that good decisions depend on good information. The complete absence of information makes decision making a process of guesswork. Extraneous information makes finding the good information harder and can lead to confusion. Because agents are involved in making decisions the nature and quality of the information they receive is important. Information acquisition (or sensing) must articulate with the decision architecture of the agent, but there are also broader issues of what constitutes good information, how it can be identified, and how agents systems should acquire it.

In most AI research investigation of information acquisition is often subordinate to consideration of decision making architectures. In practice, sensory mechanisms are often unsophisticated compared to their partner decision architectures. Existing investigation of information have emphasised the total value of specific pieces of information (for example [17]) rather than what information should be provided to an agent. Other approaches that have improved the information available to agents have substantially improved agent interactions with the environment. One approach has been annotated environments, which have enhanced particular kinds of behaviour by placing specific information about how to use objects in the world. Using this approach it has been possible to create virtual worlds where actors show a substantial level of behavioural sophistication, while actually being quite simple in operation [2, 3, 4, 10,16]. The increase in the sophistication of agent behaviour that results from annotations suggests that further focus on the information environment of agents can substantially improve their behaviour.

There are many things that agents need to do that are not suited to an annotation approach. Annotations do not provide a mechanism for agents to show general, flexible, innovative actions. Annotations are like schemata, which tell an agent that something should be used in a particular way that follows a behavioural formula[12]. Identifying ways of providing information to agents that improves the way they interact with their environment in a more general and less defined fashion requires something that goes beyond annotation. An approach drawn from affordance theory [5] could be one way of generating adaptable interaction with the environment. Affordance theory emphasizes the relationship between the capabilities of an entity and the information it obtains from the environment. Affordances are about identifying the mutual interactions between an agent in a setting, both what it can do and what can be done to it. Effective sensory mechanisms should help agents interact more accurately and efficiently, and the emphasis of affordance oriented perception is about seeing what can be done in the environment. This focus on affordance based sensing requires identifying what capabilities the actor has, what sort of information is relevant to these capabilities, and how to make these sensory processes efficient.

2. Architectural Influences on Sensing

The sensory processes of agents are closely coupled with their decision making architectures. The sense data acquired by an agent needs to correspond to the material it uses for making decisions, and the sort of actions it takes in the environment. Agents need to sense within the constraints of their environment and the software architecture of virtual worlds creates a hard limit on what an agent can perceive and do. Within these limits, the architecture of the agent determines what information is meaningful to an agent. Narayek[13] describes several architecture based categories of agents, with an emphasis on their decision making mechanisms. Differences between the decision architectures are paralleled by differences in the sort of information they sense and use.

1.1 Reactive agents

Reactive agents operate via a stimulus-response mechanism and retain no internal state[13]. Particular sense inputs activate matched response mechanisms without regard to previous or possible events. For these types of agents, sense data might be boolean, such as the presence or absence of a foe, or variable, such as distance, likability or health. For these agents, all sensory inputs are state-independent, as they must be meaningful, no matter whether the agent has encountered it many times before or not at all. The information required by reactive systems is meaningful on a moment by moment basis. Information can be useful even when it is disconnected from preceding or succeeding events.

Many agents operating in First Person Shooter (FPS) environments have used highly reactive architectures, with the result that sensory data is immediate and atomically meaningful. For example Ledgewalker[7] is a Counterstrike bot that mainly uses reactive elements. Its sensory systems are matched to associated actuator systems to form behaviours, which are primed to trigger whenever appropriate sense data is received. For example, the shooting actuator uses positional sensors for enemy location to determine where to shoot. Inputs to the shoot actuator is based on spatial inputs, and any variability in performance is due to errors introduced into sensing, rather than in the actuator process. This operational independence means sense data for each behaviour actuator must be discrete. The sort of information acquired by the sensors reflects this. Dedicated sensors acquire information about entities, such as the nearest enemy, and events, such as nearby sounds. This approach can deliver competent behaviour, especially in settings where an agent can operate reflexively. Discrete information is the material of reflex and reaction, well suited to purely reactive mechanisms.

1.2 Trigger agents

Trigger agents differ from reactive agents as they have internal states which adjust to new information over time[13]. Like reactive agents, they have a fixed repertoire of hard wired responses to sensory stimuli, but these responses are altered by changes in agent state. Trigger agents can operate reflexively like reactive agents, but they also have variable internal characteristics such as beliefs that alter the way they interact with the environment. Because of this their information environment is a lot richer, as they can use this information in different ways. More significantly they can also use information that would be meaningless without temporal or causal context.

One type of information unavailable to purely reactive agents is temporal information. Chase behaviours are substantially improved by keeping positional information for a time, as it helps an agent track moving targets more effectively[18]. These predictive capabilities require an agent to relate positional and time information which serves as a predictive mechanism for chase or flight interactions with other moving objects.
Internal states also permit agents a more sophisticated behavioural repertoire, as they influence the context and relevance of information to an agent. They can be used to alter an agents receptivity to a stimulus, or to contribute to more complex predictive mechanisms. The sensory mechanisms used in the Thief series used stateful characteristics to represent differing levels of sensory awareness[9]. Differing levels of alertness altered agent sensory acuity and thereby influenced the likelihood of responding to objects in the sensory field. While the information available to the agents remains constant, the salience of information varies depending on agent state. The SOAR architecture[8] used similar sensory mechanisms to human players to build a model of likely opponent behaviour, permitting the agent to preempt the actions of other entities in the environment. The information sensed by stateful agents is similar to the stimuli used by reactive agents, but they can use it to self-generate information. Without the faculties to generate contextual stimuli, agents are limited to operating on information that is conceptually discrete. Stateful information requires the presence mechanisms such as temporal sensitivity or internal context.

1.3 Anytime agents

Anytime agents demonstrate adaptable behaviours that move beyond the fixed responses of reactive and trigger agents[13]. These agents use a basic set of capabilities that can be variably combined to create more complex behaviours, and can be re-applied to the same event in different ways. The information used by these systems needs to be similarly modular so that the information applies to a range of candidate actions and plans.

One example of this approach is Goal Oriented Action Planning [15, 16], which dynamically generates plans from a repertoire of behaviours in order to achieve a goal. GOAP matches this information by mapping goals and actions to available resources using a plan searching mechanism. Each goal specifies the conditions in the world that need to be fulfilled using a relatively simple data structure. The agent looks for sequences of actions that will fulfill the goal condition by searching through sequences of actions that generate the desired world state. To determine appropriate actions, agents use sensors matched to appropriate action subsystem to query the environment for appropriate objects, which are tagged according to their purpose.

Agents using a GOAP architecture demonstrate sophisticated interactions with their environment. For example, Orkin [15] describes the following behaviour generated by GOAP:

> *While testing a new feature that allows the player to steal an NPC's weapon, a developer was surprised when the NPC responded by running to grab a pipe off the wall and returning to flog the player with it!*

The sophistication of this behaviour is partially due to the improved decision making architecture of the agents, and partially due to the type of information the agents acquire. To acquire information about opportunities in the environment the agent used sensory mechanisms that similar to most other contemporary approaches, as it used dedicated sensors paired with action subsystems[14]. In this example, the pipe is tagged as a weapon, permitting the sensor to recognise it as an appropriate resource. Despite the flexibility of the planning mechanism, objects are still defined according to specific roles. The information available to agents in this approach is contained in a repository of categories and properties of objects termed Smart Objects[16]. This information is a distillation of the developer's knowledge and intelligence into information that is useful to an AI. While the planning mechanisms available to the agent are flexible and integrable, the information available to the agent is static.

3. Enhancing Information Resources

The type of information that an agent needs depends on how it is supposed to use it, but there are other issues associated with sensing. While the architecture of an agent determines what sort of information is meaningful, another issue is what features, characteristics and elements of the environment reach the agent. Information must be salient as well as structurally comprehensible. Given the need to do some particular thing, what information is the most effective? Similarly, what sort of information is the most useful for a range of actions an agent might perform? Selecting what an agent should sense requires balancing specificity (and behavioural brittleness) with generality (but possibly suboptimal behaviour).

1.4 Generalisation and Behavioural Fit

Most of the time it is possible for agents to act on relatively non-specific information. By dealing with generalities rather than specifics whenever possible it is more likely that the cost of acquiring and processing information will be reduced. In many environments, there are highly portable characteristics that translate across a range of scenarios. Consider a high-ceilinged warehouse, with an elevated office supported by poles at the rear of the building. There are stairs leading to one door at the front of the office, while another door at the side connects to a walkway that runs across the back wall and side opposite the stairs, with a ladder down to the floor. Underneath the walkway is a large truck. There are several routes to the office from the front of the building, including the stairs, the ladder and clambering onto the truck and then the walkway. Converting the scenario to a jungle setting does not change the problem. If a tree house replaces the office, branches replace the walkway, vines replace the ladder and an imperturbable elephant replaces the truck the nature of the problem and the salient features of the environment remains the same. In either setting, only limited features of the objects are relevant to the problems the agent faces and there is a lot of information that can be used in a general way.

This argument for generality has been made effectively by Harnard [6]. His argument is that categorisation is a way of managing variability in sense data. An ability to categorise and identify invariant features is an essential part of figuring out what is important in the environment. This categorisation process is mutable- we might look at a garden of flowers and differentiate between the primroses and other types of flowers, or differentiate between the flowers and other plants [6]. Without a means of weighting input, any two entities are as similar as any other two[19] so categories result from grouping by general criteria. Categorisation provides a heuristic for handling sensory information in manageable chunks. Identifying how to condense the information available to an agent is an important issue. It affects the cognitive capabilities of the agent by determining how the agent identifies generalities, and it affects the efficiency of the agent by altering how wide reaching its sensory and cognitive mechanisms must be.

Generalising information resonates with the way many objects are used in virtual worlds. An object that features in many games is the trusty crate. Their common use illustrates potential benefits of categorisation that minimise extraneous information. In games, crates are often used for cover, de-facto ramps and occasionally storage. A crate is rarely actually used as a crate! A crate used as a ramp or cover could be replaced with a chest, box or bale with no change in utility. WordNet[11] uses a semantic network of words and concepts to provide information about their relationship with other words and concepts. A feature of WordNet is the ability to detail the component parts of a crate. The edited meronym query for crate in Figure 1 shows how little agents actually need to sense about crates. In many environments, all agents need to know about crates are that they can be climbed on, provide cover, can be moved, and are heavy enough to cause damage if they fall on someone.

crate -- (a rugged box (usually made of wood); used for shipping)
=> box -- (a (usually rectangular) container; may have a lid)
HAS PART: base -- (a flat bottom on which something is intended to sit)
HAS PART: lid -- (a movable top or cover (hinged or separate) for closing the opening of a container)
HAS PART: lock -- (a fastener fitted to a door or drawer to keep it firmly closed)
HAS PART: bolt, deadbolt -- (the part of a lock that is engaged or withdrawn with a key)
HAS PART: keyhole -- (the hole where a key is inserted)
HAS PART: tumbler -- (a movable obstruction in a lock that must be adjusted to a given position (as by a key) before the bolt can be thrown)

Figure 1: *Component features of a crate.*

Generalising information offers additional advantages. Generalised information is potentially more durable and less situationally fragile than scripted uses of objects. An agent attempting to push a crate onto an intruder is really just looking for is something solid and heavy. This could be achieved by scripting the action into the crates. However, if there are other solid heavy objects nearby, they would need to have scripts attached as well. While embedding scripts into the environment is a powerful tool it also limits the extent to which an agent can innovate in the environment. An alternative approach is to make information about the attributes that make an object worth pushing available to agents. Some of this information exists in the object, such as its size and weight. Other information exists in its relationships, such as whether it can be pushed to an appropriate destination (in this case off an edge). Of course other knowledge is also important, such as when it is appropriate to push things onto people.

4. Matching Capabilities to Characteristics

From an affordance perspective, the actions available to an agent in the environment are its capabilities. Because several actions may be possible in a particular setting, the information that reaches the agent should indicate what capabilities can be used, and what they can be used on. This information needs to be fitted to the capabilities an agent has, as they might only marginally differ in what makes them feasible. Providing information in a general way permits an agent to ask several questions about what they can do with the environment. One question is "what can I do with the things around me?", and another is "where is something I can use in a particular way?".

The properties of the crate examined previously illustrate how this information can be presented in a way that fits the capabilities of an agent. Consider an agent with three capabilities that deal with moving things around: *carry*, *throw* and *push*. Each of these capabilities has common requirements, as objects must be movable and solid, but they also have differing requirements such as the size and weight of the object. Thus, the information relevant to these capabilities could be classified as *movable*, *size*, and *weight*. *Movable* describes the physical constraints on the object. If the object is bolted to the door (or is the door) it would be a complex process to relocate it, so generally the object is immovable. *Size* and *weight* are measurements, so they need to provide an estimate of value. One simple categorisation might break them into 4 categories. *Size* could be generalised as small (carry in one hand), medium (2 hands), large (2 people) or massive (> 2 people). Weight could be generalised as light (an apple), medium (a person), heavy (a piano) or really heavy (a truck). The articulation between the properties of the crate and an agents capabilities are shown in Table 1. The prerequisites for a particular capability compared to the attributes of a particular object indicate whether a particular capability can be used on an object. By using these simple generalisations, it is possible to provide an agent with a lot of information about their environment in a condensed and reusable form.

Table 1: *Matching capabilities to object characteristics*

Capability	**Requirements**	**Properties**	**Feasible**
Carry	Movable Small – Medium Size Light – Medium Weight	Movable Medium Heavy	N
Throw	Movable Small Size Light Weight	Movable Medium Heavy	N
Push	Movable Medium – Massive Size Light – Heavy Weight	Movable Medium Heavy	Y

It may seem more efficient to simply flag what can be done to an object instead of indicating the properties an object has. Simply tagging an object with possible actions makes sense, subject to certain conditions. If all the potential actors on the object have identical capabilities, the properties of the environment are identical for each participant. However, if the innate capabilities of agents vary, this approach will result in mismatches between what the agent can do, and what the agent is told it can do. For example, a robotic spider might have substantially different capabilities to a human. It may be weaker (and therefore less capable of pushing things) but it may be able to climb vertical surfaces. Moreover, the physical properties of each object are relatively constant in a virtual environment. A crate has the same dimensions and weight, no matter whether the actor is weak or strong, short or tall. What changes is which of those properties relate to an agent, and to the particular capabilities the agent possesses. This approach of viewing object properties as constant also accommodates environments that have characters with variable skills. To revisit Table 1, the requirements for the throw capability can be extended to include heavy objects if the strength of an agent improves. This approach means that the environment does not need to change when the agent does.

Making information simpler by generalising it helps agents obtain information in a manageable way. It also reduces the details agents need to deal with. This approach helps the agent, but it doesn't necessarily make for quality AI. The argument against this approach parallels that of Chalmers et al.[1]. They argue that much of what constitutes intelligence comes from being able to extract the important information from the environment, and that having an experimenter or designer do this on behalf of the agent detracts from the AI itself. This limitation holds for the approach in this paper. If the information available to an AI is filtered by the designer, much of the intelligence is siphoned away from the AI. While this criticism is important and valid, it needs to be treated pragmatically. Generalising information for the consumption of an agent may not be ideal, but it provides a means of testing how agents that use this type of information can perform. The results of this approach peels back another layer of the problem and provides impetus for future work looking to improve this part of information acquisition.

5. Conclusion

Paying closer attention to the processes of information acquisition can enhance the way that agents in virtual worlds operate. The need to get good information is paramount but the issue has seen very little direct analysis, at least within the domain of AI. In part, this is because there are plenty of other worthwhile issues to be tackled. However, whenever a piece of software is used to provide analysis and decision making based on some information, the questions of what is important to know and how the information can be sensed should be asked. Addressing information acquisition implicitly foregos the opportunity to examine the nature of the information environment and to apply information-oriented approaches to AI. These approaches include wider examination of the aspects of information acquisition, including material drawn from the cognitive sciences.

This preliminary analysis suggests that agents will benefit from acquiring information that closely articulates with what they can do in their environment, and is efficient and reusable. Agents need to acquire information about a range of issues that extend beyond object properties. These include identifying the relationships between objects and information about other actors. Actors also need to be able to adjust their perceptual focus based on their motivation and circumstances. There are many approaches that can suggest ways of improving information acquisition as sensing has been an issue for psychologists and philosophers for far longer than computers have existed. However, what computing based sensing does is bring flexible tools and perspectives into the field. The ability to easily specify and manipulate information in a virtual worlds provides an interesting and flexible way of examining the issue.

6. References

[1] Chalmers, D., French, R. & Hofstadter, D. (1991), `High-Level Perception, Representation and Analogy: A Critique of Artificial Intelligence Methodology', *Journal of Experimental and Theoretical Artificial Intelligence*.

[2] Doyle, P. (1999), Virtual intelligence from artificial reality: Building stupid agents in smart environments, *in* `AAAI '99 Spring Symposium on Artificial Intelligence and Computer Games'.

[3] Doyle, P. & Hayes-Roth, B. (1998), Agents in annotated worlds, *in* K. P. Sycara & M. Wooldridge, eds, `Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)', ACM Press, New York, pp. 173-180.

[4] Forbus, K. D. (2004), Under the hood of The Sims, Technical report, Northwestern University. http://www.cs.northwestern.edu/forbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/v3_document.htm

[5] Gibson, J. J. (1979), *The Ecological Approach to Visual Perception*, Mifflin, Boston.

[6] Harnard, S. (2003), There is no Concrete (or: Living Within One's Means), *in* `Access to the Abstract: An Interdisciplinary Conference on the Epistemology of Abstract Objects', University of Southern Denmark at Odense.

[7] Khoo, A., Dunham, G., Trienens, N. & Sood, S. (2002), Efficient, Realistic NPC Control Systems using behaviour-Based Techniques, *in* K. Forbus & M. S. El-Nasr, eds, `Papers from 2002 AAAI Spring Symposium', AAAI.

[8] Laird, J. & Duchi, J. (2001), Creating Human-Like Synthetic Characters with Multiple Skill Levels: A Case Study Using the Soar Quakebot, *in* `Papers from the 2001 AAAI Spring Symposium on Artificial Intelligence and Computer Games', pp. 54-58.

[9] Leonard, T. (2003), Building an AI Sensory System: Examining the Design of Thief: The Dark Project, *in* `Game Developer's Conference Proceedings'. http://www.gamasutra.com/gdc2003/features/20030307/leonard_01.htm

[10] Macedonia, M. (2000), `Using technology and innovation to simulate daily life', *Computer* **33**(4), 110.112.

[11] Miller, G.A. Fellbaum, C. Tengi, R. Wolff, S. Wakefield, P. Langone, H. & Haskell, B. (2004), WordNet 2.0 Online Search, http://wordnet.princeton.edu/cgi-bin/webwn

[12] Minsky, M. (1975), A Framework for Representing Knowledge *in* Winston, ed, `The Psychology of Computer Vision', McGraw-Hill

[13] Narayek, A. (2002), *Intelligent Agents for Computer Games*, Springer, pp. 414-422.

[14] Orkin, J. (2004*a*). personal correspondence.

[15] Orkin, J., Baillie-de Byl, P., Borrajo, D., Funge, J., Garagnani, M., Goetz, P., Kelly III, J. J., Millington, I., Schwab, B. & Young, R. M. (2004), Working Group on Goal oriented Action Planning, Technical report, International Game Developers Association. http://www.igda.org/ai/report-2004/goap.html.

[16] Orkin, J. (2004*b*), Symbolic Representation of Game World State: Toward Real-Time Planning in Games, *in* D. Fu, ed., `AAAI "Challenges in Game AI" workshop'.

[17] Poole, D., Mackworth, A. & Gobel, R. (1998), *Computational Intelligence: A Logical Approach*, Oxford University Press, New York.

[18] Sanchez-Crespo Dalmau, D. (2003), *Core Techniques and Algorithms in Game Programming*, New Riders, Indianopolis.

[19] Watanabe, S. (1985), *Pattern Recognition: human and mechanical*, Wiley, New York.

PROCEEDINGS OF THE 2005 INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE

ICAI'05

Volume II

Editors

Hamid R. Arabnia
Rose Joshua

Associate Editors

Oscar Castillo, Sven F. Crone, Muddassar Farooq
David de la Fuente, Peter M. LaMonica
Raymond A. Liuzzi, Eric Matson, Patricia Melin
Ronaldo Menezes, Youngsong Mun, Ying-Lie O
Jose A. Olivas, Robert Stahlbock, Alexander Toet

Las Vegas, Nevada, USA
June 27-30, 2005

ISBN: 1-932415-66-1, 1-932415-67-X (1-932415-68-8)
Printed in the United States of America