Robot Transparency: Improving Understanding of Intelligent Behaviour for Designers and Users

Robert H. Wortham, Andreas Theodorou, and Joanna J. Bryson

University of Bath, Bath BA2 7AY, UK,
{r.h.wortham, a.theodorou, j.j.bryson}@bath.ac.uk,
WWW home pages: http://www.robwortham.com,
http://www.recklesscoding.com, http://www.cs.bath.ac.uk/~jjb/

Abstract. Autonomous robots can be difficult to design and understand. Designers have difficulty decoding the behaviour of their own robots simply by observing them. Naive users of robots similarly have difficulty deciphering robot behaviour simply through observation. In this paper we review relevant robot systems architecture, design, and transparency literature, and report on a programme of research to investigate practical approaches to improve robot transparency. We report on the investigation of real-time graphical and vocalised outputs as a means for both designers and end users to gain a better mental model of the internal state and decision making processes taking place within a robot. This approach, combined with a graphical approach to behaviour design, offers improved transparency for robot designers. We also report on studies of users' understanding, where significant improvement has been achieved using both graphical and vocalisation transparency approaches.

Keywords: Robot Transparency, Reactive Planning, Behaviour Oriented Design, Instinct Planner, ABOD3, Arduino, POSH

1 Introduction

Autonomous robots can be difficult to design. Robot designers often report that they have difficulty decoding the behaviour of their own robots simply by observing them. This may be because the robot behaviour provides too little information to enable the designer to envisage the internal processing within the robot giving rise to its behaviour, or it may be because the designer cannot store or recall all the program details necessary to create an adequate mental model against which the behaviour can be evaluated. As robot complexity increases, measured in terms of the range of possible behaviours, robot designers find it increasingly hard to debug their robots. This may lead to long periods of forensic offline debugging, reducing designer productivity and possibly leading to project abandonment or downsizing.

Those who encounter and interact with a robot without knowledge of the design and operation of its internal processing face an even greater challenge to create a good model of the robot simply by observing its behaviour. *Transparency* is the term used to describe the extent to which the robot's ability, intent, and situational constraints are understood by users [18][3]. Humans have a natural if limited ability to understand others. However this ability has evolved and developed in the environment of human and other animal agency, which may make assumptions to which artificial intelligence does not conform. Therefore it is the responsibility of the designers of intelligent systems to make their products transparent to us [24][27].

We believe that transparency is a key consideration for the ethical design and use of Artificial Intelligence. We are working on a research programme to provide the knowledge and software tools to create a layer of transparency. This helps with debugging and with public understanding of intelligent agents. In this paper we review findings from user studies, using purpose-made tools, to back our original hypothesis regarding the usefulness of transparency.

2 Autonomous Agent Architectures and Transparency

Early work to build software architectures for real world robots soon recognised the problem that robots operate in dynamic and uncertain environments, and need to react quickly as their environment changes [13]. Brooks reinforced the point that a 'Sense-Model-Plan-Act' (SMPA) architecture is inadequate for practical robot applications [7]. Brooks' subsumption architecture is a design pattern for intelligent embodied systems that have no internal representation of their environment, and minimal internal state.

Modularity, hierarchically organised action selection, and parallel environment monitoring are recognised as important elements of autonomous agent architectures [8]. Modularity is important to simplify design. Hierarchical action selection focusses attention and provides prioritisation in the event that modules conflict. Parallel environment monitoring is essential to produce a system that is responsive to environmental stimuli and able to allow the focus of attention to shift. These ideas of modularity and hierarchy are essentially similar to some writers' modular and hierarchical models of human minds [15][19]. Bryson argues that both modularity and hierarchical structures are necessary for intelligent behaviour [9]. Subsequent work established the idea of reactive planning operating together with deliberative control, for example the Honda ASIMO robot [21] and the IDEA architecture used by NASA for the Gromit exploration rover [14].

2.1 Planning, Methodology, and Architecture

Bryson [10] extended Brooks' ideas, embracing agile and object oriented design [2], to create the Behaviour Oriented Design (BOD) development methodology. BOD requires some form of hierarchical dynamic planning system, and Bryson introduced the Parallel Ordered Slip-stack Hierarchical (POSH) planner for this purpose. The BOD architecture is used widely for both research and game AI [16][17]. POSH is straightforward to understand for beginners, allowing them to program their first agents quickly and easily [6]. Wortham has also recently re-implemented, extended, and optimised POSH for embedded micro-controllers [25].

Today, development frameworks for autonomous robots vary but are typically based on a behaviour based model, with reactive planning controlling the immediate behaviour of the robot, and a higher deliberative level serving to interact with the reactive layer to achieve longer term goals. Other design patterns also exist for specific application areas, for example in social robotics [20]. These patterns can be layered on top of the underlying reactive behaviour based model, as they specify behaviour at a high level of abstraction. These development frameworks, design patterns and architectures are valuable because they reduce the number of degrees of freedom that the robot designer has to develop their system. This may seem counter-intuitive, but as Boden shows in her seminal work on computational creativity [4], useful creativity is only achievable within some pre-existing framework to limit the search space within which individuals can search for novel solutions.

The Japanese Poka-yoke approach is used effectively within manufacturing, to guide employees in their work and reduce human error [22]. The purpose of Poka-yoke is to eliminate product defects by preventing, correcting, or drawing attention to human errors as they occur, and this is generally achieved by using templates, jigs or other devices to reduce the number of degrees of freedom available to operators within a manufacturing environment. A parallel can be drawn here with robot development frameworks, which similarly guide robot designers, helping them to achieve their desired robot functionality in a well structured, productive, and transparent environment. In this context transparency is taken both to mean the extent to which the framework is well known to the designer and to other designers, and the extent to which the framework itself provides timely and useful feedback to the designer during development.

The essential problem faced by a robot designer when observing a robot can be summed up as "why is the robot doing {X BEHAVIOUR}?" which for the developer really means "what code within the robot is executing to drive this behaviour?" Observers and users of a robot may ask the same question, but in this case what is meant is "What is the robot trying to achieve by doing {X BEHAVIOUR}? What is the purpose of this behaviour?" Fortunately, the action selection mechanisms within robots are typically arranged such that both the developer's question, and the observer or users' question may be answered in the same way, by identifying the names of the particular action modules being employed. More than one module may be active at any given time, however robot actions are typically hierarchically structured, and designers and users may be interested in receiving information from different levels within this hierarchy. In addition, during the robot behaviour design process, the designer needs rapid access to the structure of the robot's hierarchical control mechanism. This requirement favours a graphical approach to the problem of behaviour design, and has been addressed by various systems AI tools, such as the graphical Advanced BOD Environment (ABODE) [5][11].

2.2 Designing Systems for Traceability and Transparency

Consider the situation where a self-driving car does not detect a pedestrian and runs over them. Who is responsible? The robot for being unreliable? The human passenger, who placed their trust in the robot? The robot designer or manufacturer? Given that the damage done is irreversible, accountability needs to be about more than the apportionment of blame — you cannot punish a robot. These questions are matter of ethics, and beyond the scope of this paper [27]. However, what is clear is that when errors occur in autonomous systems they must be addressed, in order that they do not re-occur. Traceability of autonomous agent behaviour is essential to determine the causes of these errors. Transparency in an architecture can facilitate traceability, as data used by the decision making mechanism becomes accessible and thus recordable. This allows developers to recreate incidents in controlled environments and fix issues.

Transparency can help us trace incidents of misbehaviour even as they occur, as we can have a clear, real-time understanding of the goals and actions of the agent. However, errors are prone to be made, and when they do they must be addressed. In some cases errors in coding must be redressed, and in all cases these reports should be used to reduce the probability of future mishaps. Implementing transparency requires capture of both sensor data and the internal state of the robot. If these are retrievable, the cause of misbehaviour becomes more likely traceable. Adding traceability to the action selection mechanism allows us to record and understand the sequence of events the lead to an incident, similar to the purpose of an aeroplane's black box flight recorder. This can allow not only the apportionment of accountability for the incident, but also help robot designers to make appropriate adjustments in future versions of the robot.

In order to explore these problems of transparency further, and to develop simple software tools that can assist designers in the creation of robots, the authors have embarked upon a programme of research in robot transparency. As part of this work, we have developed a small maker robot, new graphical design tools and a graphical real-time debugger. We have also explored vocalisation of transparency information as an audible alternative to visual communication. Further details of the robot are provided in the next section. We have also conducted transparency experiments with subjects having no prior knowledge of the robot or its purpose. Section 4 reviews this work to date. We demonstrate that abstracted and unexplained real-time visualisation of a robot's priorities can substantially improve human understanding of machine intelligence, even for naive users. We further demonstrate that based on the same data feed, a vocalised output from the robot can also be used to improve understanding. Section 4.5 reports the efficacy of our visual design and debugging tools for robot developers. Section 5 describes our conclusions so far from this work, and also planned future activities.

3 The R5 Robot

As first presented by Wortham et al [25], R5 is a low cost maker robot[1], based on the ARDUINO micro-controller [1], see Figure 1. The R5 robot has active infra-red distance sensors at each corner and proprioceptive sensors for odometry and drive motor current. It has a head with two degrees of freedom, designed for scanning the environment. Mounted on the head is a passive infra-red (PIR) sensor to assist in the detection of humans, and an ultrasonic range finder with a range of five metres. It also has a multicoloured LED "headlight" that may be used for signalling to humans around it. The robot is equipped with a speech synthesis module and loudspeaker, enabling it to vocalise textual sentences generated as the robot operates. In noisy environments, a blue-tooth audio module allows wireless headphones or other remote audio devices to receive the vocalisation output. The audio output is also directed to a block of four red LEDs, that pulse synchronously with the sound output. It also has a real-time clock

[1] Design details and software for the R5 Robot: http://www.robwortham.com/r5-robot/

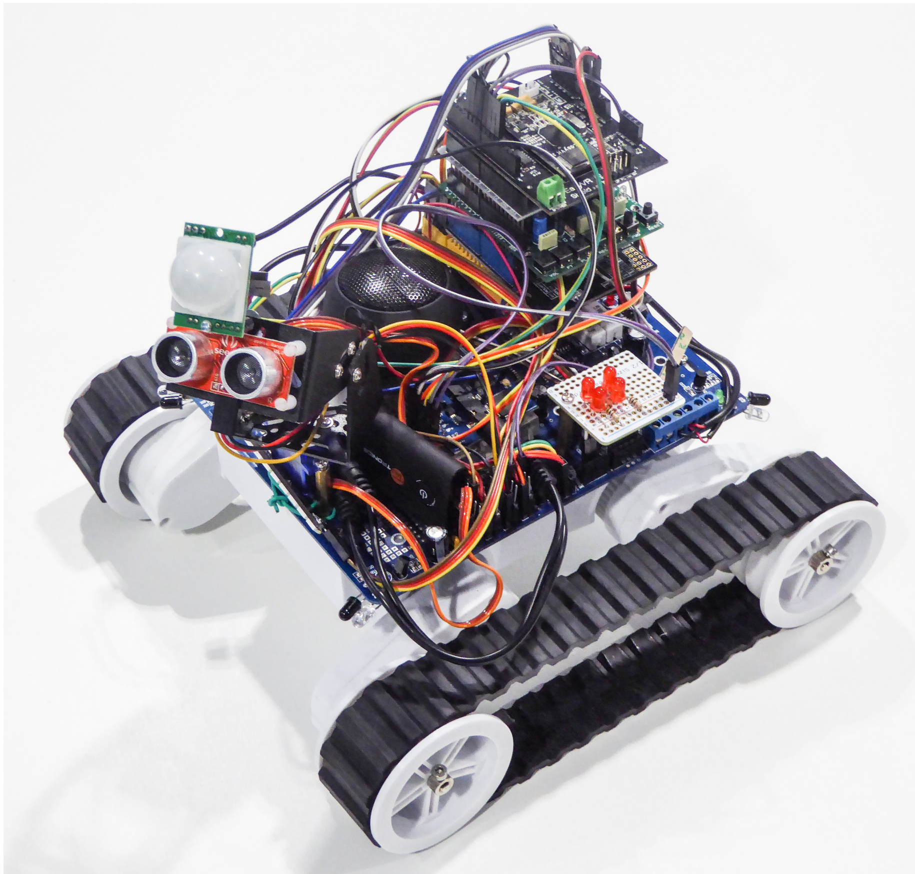

Fig. 1. The R5 Robot. This photograph shows the head assembly with PIR and ultrasonic range-finder attached. The loudspeaker and blue-tooth audio adapter are also visible. The Four red LEDs are powered from the audio output, and serve as a visual indication of the vocalisation [25][28].

(RTC) allowing the robot to maintain accurate date and time, a wifi module for communication and an electronically erasable programmable read only memory (EEPROM) to store the robot's configuration parameters. The robot software is written as a set of C++ libraries. The following section outlines the methodology and tools used to write its program.

3.1 BOD and Reactive Planning

Development of the software for the robot follows Bryson's BOD methodology [10]. We use the *Instinct* reactive planner [25] as the core action selection mechanism for the R5 robot. The Instinct Planner is based on a POSH Planner [10]. The Instinct reactive plan is produced using the Instinct Visual Design Language (iVDL) graphical authoring tool [25]; an example plan for the R5 robot is shown in Figure 3. The development process follows this simple algorithm:

1. Develop low level behaviours and senses based on robot physical capabilities.
2. Develop more complex behaviours and sensor fusions (compound sensor models) as necessary, based on an analysis of the likely scenarios that the robot will face.
3. Produce a reactive plan using iVDL, again based on the functional requirements for the robot, and an analysis of the likely scenarios that the robot will face.
4. Test and iterate both the behaviour design, sensor model and reactive plan design until the robot behaviour is as required in the anticipated range of operational environments. During this iterative process, use the Instinct transparency feed both at runtime and offline to understand the interaction of subsystems and explain the resultant behaviour of the robot.

The Instinct Planner[2] includes significant capabilities to facilitate plan design and runtime debugging. It reports the execution and status of every plan element in real time, allowing us to implicitly capture the reasoning process within the robot that gives rise to its behaviour. The planner has the ability to report its activity as it runs, by means of callback functions to a monitor class. There are six separate callbacks monitoring the Execution, Success, Failure, Error and In-Progress status events, and the Sense activity of each plan element. In the R5 robot, the callbacks write textual data to a TCP/IP stream over a wireless (WiFi) link. A Java based Instinct Server receives this information and logs the data to disk. This communication channel also allows for commands to be sent to the robot while it is running. Figure 2 shows the overall architecture of the planner within the R5 robot, communicating via WiFi to either the logging server, or the ABOD3 tool, described in Section 4.1. The robot typically operates with a plan cycle rate of 8Hz, yielding a transparency data rate of approximately 100 report lines per second (depending on the depth of the plan hierarchy), or 3,800 bytes of data per second.

4 Evaluation of Methods and Results

For all our experiments to date, the robot has operated with the same plan, see Figure 3. The robot's overall function is to search a space looking for humans. Typical real world applications would be search and rescue after a building collapse, or monitoring of commercial cold stores or similar premises. As first presented by Wortham et al [28], the robot reactive plan has six Drives. These are (in order of highest priority first):

- Sleep : this Drive has a ramping priority. Initially the priority is very low but it increases linearly over time until the Drive is released and completes successfully. The Drive is only released when the robot is close to an obstacle and is inhibited whilst the robot confirms the presence of a human. The sleep behaviour simply shuts down the robot for a fixed interval to conserve battery power.
- Protect Motors : released when the drive motor current reaches a threshold.

[2] The Instinct Planner and iVDL are available on an open source basis, see: http://www.robwortham.com/instinct-planner/

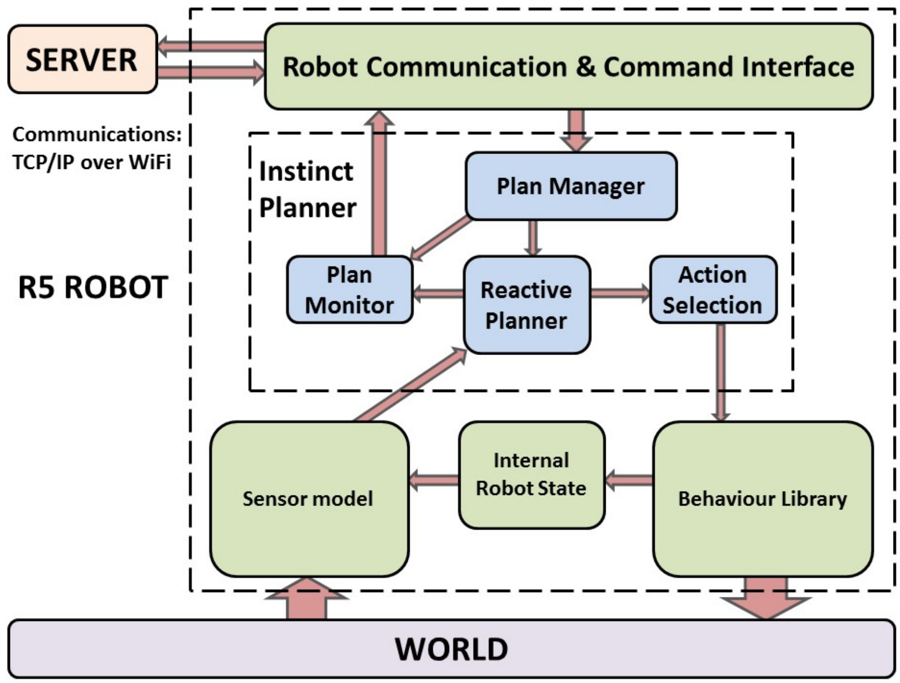

Fig. 2. R5 Robot Software Architecture showing the main architecture components and their structure. Note that the robot establishes a remote connection to the Server, from which it can optionally download an Instinct plan, download plan element names, receive user initiated commands and publish its transparency data feed [25].

– Moving So Look : enforces that if the robot is moving it scans for obstacles.
– Detect Human : released when the robot has moved a certain distance from its last confirmed detection of a human, is within a certain distance of an obstacle ahead, and its PIR sensor detects heat that could be from a human. This Drive initiates a fairly complex behaviour of movement and coloured lights, designed to encourage a human to move around in front of the robot. This continues to activate the PIR sensor, confirming the presence of a human (or animal). It is, of course, not a particularly accurate method of human detection.
– Emergency Avoid : released when the robot's corner sensors detect a nearby obstacle. This invokes a behaviour that reverses the robot a short distance and turns left or right by 90 degrees.
– Roam : released whenever the robot is not sleeping. It uses the scanning ultrasonic detector to determine when there may be obstacles ahead and turns appropriately to avoid them.

We investigate two quite distinct methods for communicating the real-time transparency feed to both the robot designers and users. The first method uses a graphical approach, the second uses an audible approach. These are described in more detail below.

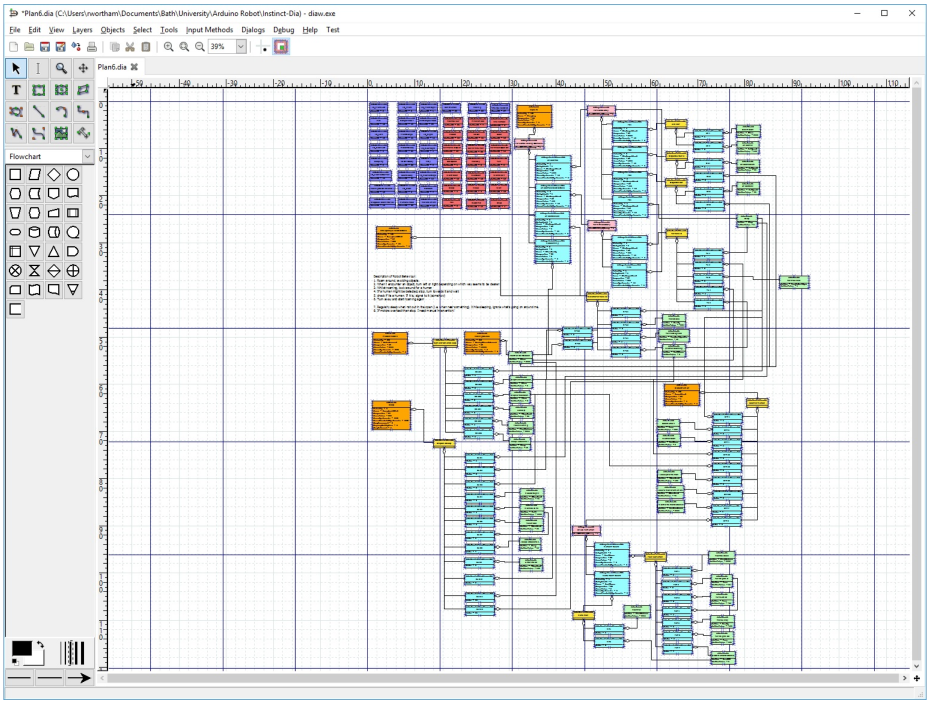

Fig. 3. An Instinct Reactive Plan for the R5 Robot, produced using iVDL within the Dia open source drawing tool. The plan labels are not visible at this resolution, but this screen shot gives an indication of the complexity of the reactive plan, and also shows how the plan was produced graphically [25].

4.1 Graphical Presentation of Transparency Data

ABODE [5] is an editor and visualisation tool for BOD agents, featuring a visual design approach to the underlying lisp-like plan language, POSH. This platform-agnostic plan editor provides flexibility by allowing the development of POSH plans for usage in a selection of planners, such as JyPOSH [12] and POSHsharp [17].

Currently, we are working towards the development of a new integrated agent development editor, ABOD3 [24] [23]. Rather than an incremental update to the existing ABODE, the new editor is a complete rebuild, with special consideration being given to producing expandable and maintainable code. It is developed in Java and uses the JavaFX GUI-framework[3] to ensure cross-platform compatibility. A public Application Programming Interface (API) allows adding support for additional BOD derivatives, other than those already supported.

ABOD3 allows the graphical visualisation of BOD-based plans, including its two major derivatives: POSH and Instinct. The new editor is designed to allow not only the development of reactive plans, but also to debug such plans in real-time, to reduce the

[3] See: http://docs.oracle.com/javafx/

time required to develop an agent. This allows the development and testing of plans from the same application, facilitating rapid prototyping. Finally, the tool is domain-agnostic, as plans can be used in a variety of different planners, within differing execution environments including robots, agent based models and game AI. ABOD3 provides an API that allows the editor to connect with planners, presenting debugging information in real time. For example, it can connect to the R5 and the Instinct planner by using a built-in TCP/IP server, supporting the simple Instinct textual transparency feed, see Figure 5. Plan elements flash as they are called by the planner and glow based on the number of recent invocations of that element, see Figure 4. Plan elements without any recent invocations start dimming down, over a user-defined interval, until they return to their initial state. This offers abstracted backtracking of the calls. Sense information and progress towards a goal are displayed. Finally, ABOD3 provides integration with videos of the agent in action, synchronised by the time signature within the recorded transparency feed. The simple UI and customisation provided by ABOD3 allow it to be

Fig. 4. The ABOD3 Graphical Transparency Tool displaying the Instinct plan [28, Figure 3] in debugging mode. The highlighted elements are the ones recently called by the planner. The intensity of the glow indicates the number of recent calls [23][24].

employed not only as a tool for developers, but also to present transparency information to the end user. Establishing the most appropriate level of complexity with which users may interact with the transparency-related information is crucial. Hiding and rearrang-

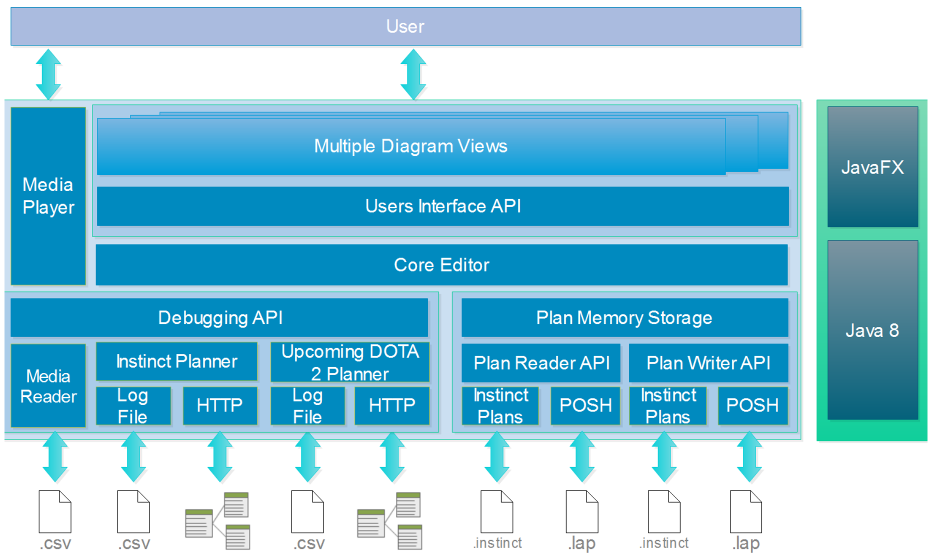

Fig. 5. System Architecture Diagram of ABOD3, showing its modular design. All of ABOD3 was written in Java for cross-platform compatibility. APIs allow the expansion of the software to support additional BOD planners for real-time debugging, BOD based plans, and User Interfaces. The editor can be tailored for roboticists, games AI developers, and even end users [23].

ing subtrees allows developers using ABOD3 to tune what and how much information they will expose to end users.

4.2 Testing Graphical Robot Transparency for Observers with ABOD3

As first presented by Wortham et al [29], our experiment with ABOD3 took place over three days at the At-Bristol Science Learning Centre, Bristol, UK. The robot operated within an enclosed pen as a special interactive exhibit within the main exhibition area, see Figure 6.

Visitors, both adults and children, were invited to sit and observe the robot in operation for several minutes whilst the robot moved around the pen and interacted with the researchers. Subjects were expected to watch the robot for at least three minutes before being handed a paper questionnaire. The questions sought to investigate the understanding of the robot by the participants, in terms of of both its intelligence and cognitive capacity, and its objectives and means of achieving them. Half of the participants had access to the ABOD3 real-time display, whilst the other half did not (the monitor was simply switched off). Full details are available in Wortham et al[29].

4.3 Plan Execution Vocalisation - iPEV

The R5 robot's transparency feed contains a single line of output each time an Instinct plan element is executed. It also contains another line to indicate whether the plan ele-

Fig. 6. The arrangement of the R5 Robot experiment using ABOD3 at the At-Bristol Science Centre. Obstacles visible include a yellow rubber duck, a blue bucket and a potted plant. The position and orientation of the transparency display is shown [29, Figure 6].

ment execution was successful, failed, still pending or resulted in an error. Since plans are hierarchical, the execution notification is received on the way down the hierarchy, and the other notifications are received on the way up. A stack structure is used to track this tree traversal, and thus it is known whether this is the first, or subsequent invocation of the element. This information is then used, together with the element type and name, to form a candidate sentence about the processing of the element. These sentences are generated far too quickly to all be vocalised, and so a filtering mechanism is used based on the following factors

1. The plan element type : Drive, Competence, Action Pattern, Action, Competence Element, Action pattern Element.
2. The event type : Execution, Success, Failure, In-progress, Error.
3. The elapsed time since the sentence was generated. If a sentence waits too long before being vocalised then it is discarded.
4. Whether the sentence is being repeated within a given time interval.

These factors can be set using the robot command line interface and are stored within the robot EEPROM. Following extensive usage testing by the designers, including feedback from user testing with students, the default parameters give priority to sentences relating to Drive execution (the highest level in the hierarchy) and the success or failure of Competences, Action Patterns and Actions. These parameters were then used for the formal experiments to test the Instinct Plan Execution Vocalisation (iPEV) at the AtBristol Science Centre, see Figure 7.

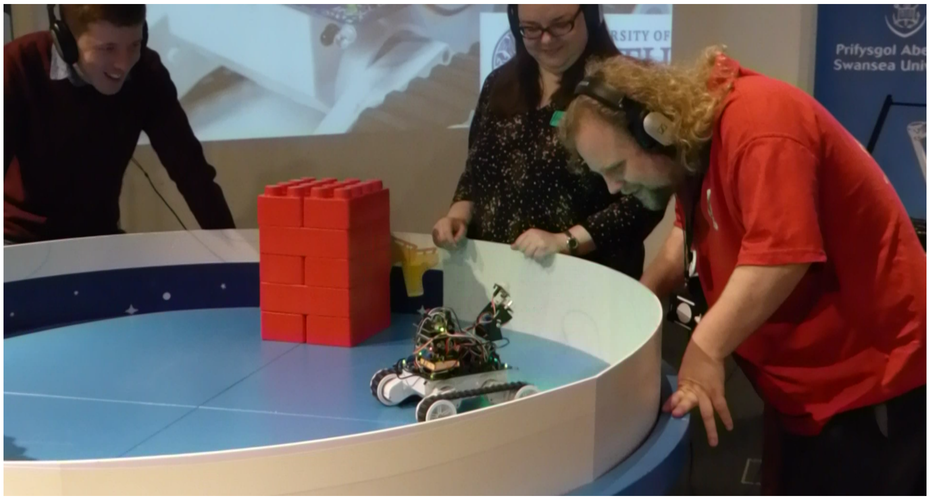

Fig. 7. A frame from a video of experiments with the R5 Robot at the AtBristol Science Centre, Bristol, UK. Due the high background noise level, participants are wearing headphones connected to the iPEV audio output of R5 via blue-tooth. This enabled them to clearly hear the robot's audio output. The robot is shown successfully detecting a human, to the delight of participants [26].

4.4 Testing Audible Robot Transparency for Observers with iPEV

These experiments were similarly conducted to those using ABOD3, and a very similar questionnaire was used, differing only in that it attempted to collect more data about the reported emotional response of participants to the robot. However in this experiment the robot operated on a large round tabletop rather than on the floor. This enabled participants to hear the robot more clearly, and interact with it at arms reach whilst standing, see Figure 7. We now consider the benefits of transparency for both robot designers and naive observers of the robot.

4.5 Transparency for Designers

During development of the R5 robot, we can report anecdotal experience of the value of offline analysis of textual transparency data, and the use of ABOD3 in its recorded mode. These tools enabled us to quickly diagnose and correct problems with the reactive plan that were unforeseen during initial plan creation. These problems were not so much 'bugs' as unforeseen interactions between the robot's various Drives and Competences, and the interaction of the robot with its environment. As such these unforeseen interactions would have been extremely hard to predict. This reinforces our assertion that iterative behaviour oriented design (BOD) is an effective and appropriate method to achieve a robust final design. The BOD development methodology, combined with the R5 Robot hardware and the Instinct Planner has proved to be a very effective combination. The R5 Robot is robust and reliable, proven over weeks of sustained use during both field experiments and demonstrations. The iterative approach of BOD was productive and successful, and the robot designers report increased productivity resulting from use of the Instinct transparency feed and the ABOD3 tool.

4.6 Transparency for Observers

The experiment using R5 with ABOD3 investigates whether seeing a real-time graphical representation of the robot's internal state and action selection processes helps naive observers to form a more useful mental model of the robot. By 'useful' we mean a model that is more closely aligned with the robot's actual capabilities, intentions, goals and limitations. Full details of this experiment and the results obtained can be found in Wortham et al [29]. However, to summarise, subjects show a significant improvement in the accuracy of their mental model of the robot if they also see the accompanying ABOD3 display. ABOD3 visualisation of the robot's intelligence makes the machine nature of the robot more transparent.

The experiment using R5 with iPEV (vocalisation) investigates whether hearing a real-time audible representation of the robot's internal state and action selection processes similarly helps observers to form a more useful mental model of the robot. Our preliminary analysis of results indicate that this is indeed the case, and that a significant improvement in the observers mental model of the robot was achieved ($N=68$, $t(66)=2.86$, $p=0.0057$). The addition of vocalisation made no significant difference the users report of their emotional responses, nor to their report of perceived robot intelligence or capacity of the robot to 'think'. Interestingly it also made no significant difference to the users self report of their ability to understand the purpose of the robot's behaviour.

5 Conclusions and Further Work

Development frameworks for autonomous robots are an important contribution to assist robot designers in their work. Today these frameworks are typically based on a behaviour based model, with reactive planning controlling the immediate behaviour of the robot, and a higher, deliberative level serving to interact with the reactive layer to achieve longer term goals. These action selection mechanisms within robots are therefore amenable to the addition of various transparency measures by simply displaying or otherwise communicating the real time control state. Our programme of research indicates that by making transparency an important design consideration for a development framework, it becomes straightforward to productively design and deliver a reliable, useful robot. The R5 robot exhibits robust behaviour over prolonged time periods in varying environments. We are also developing a body of evidence to indicate that robot designers should consider transparency as a fundamental and important trait of a robot. Taking steps to improve robot transparency significantly improves understanding of a robot in naive observers.

We plan to develop ABOD3, adding features such as "fast-forward" debug functions in pre-recorded log files, the ability to set conditional breakpoints and additional views of the reactive plan hierarchy. A fuller analysis of the results of data obtained during the robot vocalisation experiments is being prepared for subsequent publication[26]. We also intend to run workshops for robot designers, where we will further evaluate the Instinct Planner, iVDL plan authoring, the ABOD3 debugger and iPEV plan execution vocalisation.

References

1. Arduino: Arduino Website. https://www.arduino.cc/ (2016), https://www.arduino.cc/
2. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley, Reading, MA (2000)
3. Boden, M., Bryson, J., Caldwell, D., Dautenhahn, K., Edwards, L., Kember, S., Newman, P., Parry, V., Pegman, G., Rodden, T., Sorell, T., Wallis, M., Whitby, B., Winfield, A.: Principles of robotics. The United Kingdom's Engineering and Physical Sciences Research Council (EPSRC) (April 2011), web publication
4. Boden, M.a.: Creativity and artificial intelligence. Artificial Intelligence 103(1-2), 347–356 (1998)
5. Brom, C., Gemrot, J., Bída, M., Burkert, O., Partington, S.J., Bryson, J.J.: POSH tools for game agent development by students and non-programmers. In: Mehdi, Q., Mtenzi, F., Duggan, B., McAtamney, H. (eds.) The Ninth International Computer Games Conference: AI, Mobile, Educational and Serious Games. pp. 126–133. University of Wolverhampton (November 2006)
6. Brom, C., Gemrot, J., Michal, B., Ondrej, B., Partington, S.J., Bryson, J.J.: Posh tools for game agent development by students and non-programmers. ... on Computer Games: ... pp. 1–8 (2006)
7. Brooks, R.A.: Intelligence Without Representation. Artificial Intelligence 47(1), 139–159 (1991)
8. Bryson, J.: Cross-paradigm analysis of autonomous agent architecture. Journal of Experimental & Theoretical Artificial Intelligence 12(2), 165–189 (apr 2000), http://www.tandfonline.com/doi/abs/10.1080/095281300409829
9. Bryson, J.J.: The study of sequential and hierarchical organisation of behaviour via artificial mechanisms of action selection. M. phil., University of Edinburgh (2000), https://www.cs.bath.ac.uk/~jjb/ftp/mphil.pdf
10. Bryson, J.J.: Intelligence by Design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agents. Ph.D. thesis, MIT, Department of EECS, Cambridge, MA (June 2001), AI Technical Report 2001-003
11. Bryson, J.J.: Advanced Behavior Oriented Design Environment (ABODE) (2013), http://www.cs.bath.ac.uk/~jjb/web/BOD/abode.html
12. Bryson, J.J., Gaudl, S.: jyPOSH (2013), https://sourceforge.net/projects/jyposh/
13. David E., W., Karen L., M., John D, L., WESLEY, L.P.: Planning and reacting in uncertain and dynamic environments. Journal of Experimental & Theoretical Artificial Intelligence 7(1), 121–152 (1995), http://dx.doi.org/10.1080/09528139508953802
14. Finzi, A., Ingrand, F., Muscettola, N.: Model-based executive control through reactive planning for autonomous rovers. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566) 1, 879–884 (2004), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1389463
15. Fodor, J.A.: The Modularity of Mind: An Essay on Faculty Psychology. A Bradford book, MIT Press, Cambridge, Mass. (1983), http://books.google.co.uk/books?id=e7nrSeibJZYC

16. Gaudl, S., Bryson, J.J.: The Extended Ramp Goal Module: Low-Cost Behaviour Arbitration for Real-Time Controllers based on Biological Models of Dopamine Cells. Computational Intelligence in Games 2014 (2014), http://opus.bath.ac.uk/40056/
17. Gaudl, S., Davies, S., Bryson, J.: Behaviour oriented design for real-time-strategy games: An approach on iterative development for STARCRAFT AI. Foundations of Digital Games Conference pp. 198–205 (2013)
18. Lyons, J.B.: Being Transparent about Transparency : A Model for Human-Robot Interaction. Trust and Autonomous Systems: Papers from the 2013 AAAI Spring Symposium pp. 48–53 (2013)
19. Minsky, M.: The Society of Mind. Simon and Schuster Inc., New York, NY (1985)
20. Ruckert, J.H., Kahn, P.H., Kanda, T., Ishiguro, H., Shen, S., Gary, H.E.: Designing for sociality in HRI by means of multiple personas in robots. ACM/IEEE International Conference on Human-Robot Interaction pp. 217–218 (2013)
21. Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., Fujimura, K.: The intelligent ASIMO: system overview and integration. IEEE/RSJ International Conference on Intelligent Robots and System 3(October), 2478–2483 (2002)
22. Shingo, S., Dillon, A.P.: A Study of the Toyota Production System: From an Industrial Engineering Viewpoint. Produce What Is Needed, When It's Needed, Taylor & Francis, New York, USA (1989), https://books.google.co.uk/books?id=RKWU7WElJ7oC
23. Theodorou, A.: Abod3: A graphical visualization and real-time debugging tool for bod agents. In: EUCognition 2016. Vienna, Austria (December 2016), http://opus.bath.ac.uk/53506/
24. Theodorou, A., Wortham, R., Bryson, J.J.: Designing and implementing transparency for real time inspection of autonomous robots. Connection Science 29 (mar 2017), http://opus.bath.ac.uk/55250/
25. Wortham, R.H., Gaudl, S.E., Bryson, J.J.: Instinct : A Biologically Inspired Reactive Planner for Embedded Environments. In: Proceedings of ICAPS 2016 PlanRob Workshop. London, UK (2016), http://icaps16.icaps-conference.org/proceedings/planrob16.pdf
26. Wortham, R.H., Rogers, V.E.: The Muttering Robot: Improving Robot Transparency Though Vocalisation of Reactive Plan Execution {in prep} (2017)
27. Wortham, R.H., Theodorou, A.: Robot transparency, trust and utility. Connection Science 29 (2017)
28. Wortham, R.H., Theodorou, A., Bryson, J.J.: What Does the Robot Think? Transparency as a Fundamental Design Requirement for Intelligent Systems. In: IJCAI-2016 Ethics for Artificial Intelligence Workshop. New York, USA (2016), http://opus.bath.ac.uk/50294/1/WorthamTheodorouBryson_EFAI16.pdf
29. Wortham, R.H., Theodorou, A., Bryson, J.J.: Improving Robot Transparency : Real-Time Visualisation of Robot AI Substantially Improves Understanding in Naive Observers {submitted} (2017)