Citation for published version:
Theodorou, A, Wortham, RH & Bryson, JJ 2017, 'Designing and implementing transparency for real time inspection of autonomous robots', Connection Science, vol. 29, no. 3, pp. 230-241.
https://doi.org/10.1080/09540091.2017.1310182

DOI:
10.1080/09540091.2017.1310182

Publication date:
2017

Document Version
Peer reviewed version

To appear in *Connection Science*
Vol. 00, No. 00, May 2016, 1–11

Designing and implementing transparency for real time inspection of autonomous robots

Andreas Theodorou[a]* Robert H. Wortham[a] and Joanna J. Bryson[a]

[a] *Department of Computer Science, University of Bath, Bath, UK*

(*v1.1 released December 2016*)

The EPSRC's Principles of Robotics advises the implementation of transparency in robotic systems, however research related to AI transparency is in its infancy. This paper introduces the reader of the importance of having transparent inspection of intelligent agents and provides guidance for good practice when developing such agents.

By considering and expanding upon other prominent definitions found in literature, we provide a robust definition of transparency as a mechanism to expose the decision making of a robot. The paper continues by addressing potential design decisions developers need to consider when designing and developing transparent systems. Finally, we describe our new interactive intelligence editor, designed to visualise, develop and debug real-time intelligence.

Keywords: EPOR, transparency, artificial intelligence, ethics, roboethics, robotics

Index to information contained in this article

1. Introduction

Transparency is a key consideration for the ethical design and use of Artificial Intelligence, and has recently become a topic of considerable public interest and debate. We frequently use philosophical, mathematical, and biologically inspired techniques for building artificial, interactive, intelligent agents. Yet despite these well-motivated inspirations, the resulting intelligence is often developed as a black box, communicating no understanding of how the underlying real-time decision making functions.

The black-box nature of intelligent systems, even in relatively simple cases such as context-aware applications, makes interaction limited and often uninformative for the end user (Stumpf, Wong, Burnett, & Kulesza, 2010). Limiting interactions may nega-

*Corresponding author. Email: a.theodorou@bath.ac.uk tively affect the system's performance or even jeopardize the functionality of the system. Consider for example an autonomous robotic system built for providing health-care support to the elderly, who may be afraid of it, or simply distrust it, and in the end refuse to use it.

In such a scenario human well-being could be compromised, as patients may not get their prescribed medical treatment in time, unless a human overseeing the system detects the lack of interaction (or is contacted by the robot) and intervenes. Conversely, if the human user places too much trust in a robot, it could lead to misuse, over-reliance, and ultimately disuse of the system (Parasuraman & Riley, 1997).

In the previous example of a health-care robot, if the robot malfunctions and its patients are unaware of its failure to function, the patients may continue using the robot, risking their health.

Such scenarios violate the EPSRC's third as well as fourth Principle of Robotics. By not making the robots sufficiently transparent, we would also have failed to ensure they were safe in their area of application (Boden et al., 2011).

To avoid such situations, proper calibration of trust between the human users and / or operators and their robots is critically important, if not essential, in high-risk scenarios, such as the usage of robots in the military or for medical purposes (Groom & Nass, 2007). Calibrating trust occurs when the end-user has a mental model of the system and relies on the system within the systems capabilities and is aware of its limitation (Dzindolet, Peterson, Pomranky, Pierce, & Beck, 2003).

We believe that enforcement of transparency is not only beneficial for end-users, but also for intelligent agents' developers. Real-time debugging of a robot's decision making—it's action selection mechanism—could help developers to fix bugs, prevent issues, and explain potential variance in a robot's performance. We envision that with an appropriate implementation of transparency, developers could design, test, and debug their agents in real-time — similar to the way in which software developers work with traditional software development and debugging in an interactive development environment (IDE).

Despite these possible benefits of transparency in intelligent systems, there is little existing research in transparent agents or their implementation. Moreover, there are inconsistencies in the definitions of transparency and no clear criteria for a robot to be considered a transparent system. In this paper, we will present the inconsistent definitions found in the literature, then propose a revised definition which we consider the most useful in the context of the fourth EPSRC Principle of Robotics. Next, we discuss the design decisions a developer needs to consider when designing transparent robotic systems. In the penultimate section, we will present our currently in-development plan editor, ABOD3, and describe its use in the context of developing and debugging AI.

In the context of this paper, we use the term *intelligent agent* to denote the combination of both the software and hardware of an autonomous robotic system, working together as an actor, experiencing and altering the material world (J. J. Bryson, 2010). Within this paper the word *robot* always implies such agency; we do not discuss less intelligent robotics here.

2. Defining Transparency

Despite the importance assigned to transparency five years ago by the EPSRC Principles of Robotics, research into making systems transparent is still in its infancy. Very few publications have focused on the need of transparent systems and even fewer have attempted to address this need (Lyons, 2013; Novikova & Watts, 2014). Each study provides its own definition of transparency, without excluding others. To date, the transparency concept has been limited to explanations of abnormal behaviour measures of the reliability of the system, and attempts to define the analytic foundations of an intelligent system. We visit each of these in turn. Finally, we provide our meaning of the concept, by building upon the foundations laid by the existing literature.

2.1. *Transparency as Lack of Deception: The EPSRC Principle*

The EPSRC's Principles of Robotics includes transparency in principle four. Its definition there is implied by contrast: "Robots. . . should not be designed in a deceptive way to exploit vulnerable users; instead their machine nature should be transparent."

The EPSRC definition of transparency emphasizes keeping the end-user aware of the manufactured, mechanical, and thus artificial nature of the robot. However, the phrasing used allows us to consider even indirect information, such as online technical documentation, as a sufficient methodology to provide transparency (J. J. Bryson, 2012). Such a solution places at least part of the burden of responsibility with the user, which implies that not all users will find the robot transparent. A user would have to find, read, and understand the documentation or other information provided by the manufacturer, which might be opaque for some user groups.

2.2. *Transparency as a mechanism to report reliability*

One of the earliest publications to define transparency did so in terms of communicating information to the end user, regarding the system's tendency for errors within a given context of data (Dzindolet et al., 2003). While the Dzindolet *et al.* interpretation covers only part of what we think would be desirable in a definition of transparency, the study presents interesting findings concerning the importance of transparent systems. The study shows that providing extra feedback to users regarding system failures, can help participants place their trust in the system. The users knew that the system was not completely reliable, but they were able to calibrate their trust to the autonomous system in the experiment, as they became aware of when they could rely on it and when not to.

Military usage of robotic systems is becoming increasingly widespread, especially in the form of Unmanned Aerial Vehicles (UAVs). Transparency in combat systems is essential for accountability. Consider the situation where an artificial agent identifies a civilian building as a terrorist hideout and decides to take actions against it. Who is responsible? The robot for being unreliable? Or the user, who placed their trust in the system's sensors and decision-making mechanism? While the Principles are intended to ensure that responsibility falls to humans or their organisations, given that the damage done is irreversible accountability needs to be about more than the apportionment of blame. Where errors occur, they must be addressed, in some cases redressed, and in all cases used to reduce future mishaps. Wang, Jamieson, and Hollands (2009) recommend that robots working autonomously to detect and neutralize targets have transparent behaviours, in the sense that their users, who oversee the system, are alerted to contextual factors that affect the system's reliability. The oversees should have constant access to measurements of the system's reliability at its current situation and use such metrics to calibrate their trust towards the system.

2.3. *Transparency as a mechanism to report unexpected behaviour*

Studies by Kim and Hinds (2006) and Stumpf et al. (2010) focus on providing feedback to users regarding unexpected behaviour of an intelligent agent. In these studies, the user is alerted only when the artefact considers its own behaviour to be abnormal. Kim and Hinds' study shows that when increasing autonomy the importance of transparency is also increased, as control shifts from the user to the robot. These results are in line with Kahn et al. (2012), and together demonstrate that humans are more likely to blame a robot for failures than other manufactured artefacts, or human coworkers.

To achieve this sort of transparency it is essential to alert the user when a robot behaves in an unexpected way. In high-risk situations, alerting the user of a system to take control and / or to calibrate their trust appropriately could help save human lives. However, in Kim and Hinds' implementation, the robot alerts the user only when it detects that it behaves in an unexpected way. This implementation might be seen as an attempt to "fix" one black box by replacing it with another, since there is no guarantee that a robot would recognise its own misbehaviour. However, in practice it is often easier to recognise than to diagnose (let alone prevent) misbehaviours (Gat, 1992). For example, most contemporary systems that construct models of their environment can recognise an unexpected context—and even express a measure of its unlikelihood—without necessarily knowing what caused the failure of its models to predict its sensor readings. While ideally transparency could be used to enforce persistent real-time guarantees, in practice the implausible capacity to create such a perfect system might render communication to human users unnecessary. Nevertheless, a system of cognizant error detection does afford one concept of AI transparency: providing at least some ability to detect when something has or might go wrong with a system.

2.4. *Transparency as a mechanism to expose decision making*

The previous two definitions concerning accountability and reliability both deal only with exceptional circumstances. But the principle of non-deception implies that transparency might better be thought of as a more general characteristic of an intelligence. A fully transparent system may imply a mechanism integral to its intelligence for providing information in real time concerning its operation. This goes significantly beyond (though by no means deprecates) the requirement of providing access to adequate documentation suggested in the commentary provided with the Principles.

We propose that an intelligent agent such as a robot should contain the necessary mechanisms to provide meaningful information to its end users. To consider a robot transparent to inspection, the end user should have the ability to request accurate interpretations of the robot's capabilities, goals, and current progress in relation to its goals, its sensory inputs, and its reliability, as well as reports of any unexpected events. The information provided by the robot should be presented in a human understandable format. Note that the necessary requirement for human understandability requires tradeoffs in detail, as real-time decision-making events may easily occur far faster than humans can discriminate between stimuli (Pöppel, 1994).

A transparent agent, with an inspectable decision-making mechanism, could also be debugged in a similar manner to the way in which traditional, non-intelligent software is commonly debugged. The developer would be able to see which actions the agent is selecting, why this is happening, and how it moves from one action to the other. This is similar to the way in which popular Integrated Development Environments (IDEs) provide options to follow different streams of code with debug points.

3. Designing Transparent Systems

In this section, we discuss the various decisions developers may face while designing a transparent system. To date, prominent research in the field of designing transparent systems focuses in presenting transparency only within the context of human-robot collaboration. Thus, it focuses on designing transparent systems able to build trust between the human participants and the robot (Lyons, 2013). We believe that transparency should be present even in non-collaborative environments, such as human-robot competitions (Kim & Hinds, 2006) or even when robots are used by the military. Developers should strive to develop intelligent agents that can efficiently communicate information to the human end-user and sequentially allow her to develop a better mental model of the system and its behaviour.

3.1. *Usability*

In order to enforce transparency, additional displays or other methods of communication to the end-user must be carefully designed, as they will be integrating potentially complex information. Agent developers need to consider both the actual relevance and level of abstraction of the information they are exposing and how they will present this information.

3.1.1. Relevance of information

Different users may react differently to the information exposed by the robot. Tullio, Dey, Chalecki, and Fogarty (2009) demonstrate that end-users without a technical background neither understand nor retain information from technical inputs such as sensors. In contrast, an agent's developer needs access to such information during both development and testing of the robot to effectively calibrate sensors and to fix any issues found. However, within the same study, they demonstrate that users are able to understand at least basic machine-learning concepts, regardless of a non-technical educational and work-history background.

Tullio et al.'s research establishes a good starting point at understanding what information maybe relevant to the user to help them understand intelligent systems. Nevertheless, further work is needed in other application areas to establish both domain-specific and user-specific trends regarding what information should be considered of importance.

3.1.2. Abstraction of information

Developers of transparent systems will need to question not only *what*, but also *how much* information they will expose to the user by establishing a level of complexity with which users may interact with the transparency-related information. This is particularly important in multi-robot systems.

Multi-robot systems allow the usage of multiple, usually small robots, where a goal is shared among various robots, each with its own sensory input, reliability and progress towards performing its assigned task for the overall system to complete. Recent developments of nature inspired swarm intelligence allow the usage of large quantities of tiny robots working together in such a multi-robot system (Tan & Zheng, 2013). The military is already considering the development of swarms of autonomous tiny robotic soldiers. Implementing transparency in a such system is no trivial task. The developer must make rational choices about when low or high level information is required to be exposed.

By exposing all information at all times, for all types of users, the system may become unusable as the user will be overloaded with information.

We believe that different users will require different levels of information abstraction to avoid information overload. Higher levels of abstractions could concentrate on presenting only an overview of the system. Instead of having the progress of a system towards a goal, by showing the current actions the system is taking in relation to achieve the said goal, it could simply present a completion bar. Moreover, in a multi-robot system, lower level information could also include the goal, sensor, goal-process, and overall behaviour of individual agents in a detailed manner. Conversely, a high-level overview could display all robots as one entity, stating averages from each machine. Intelligent agents with a design based on a cognitive architecture, such as Behaviour Oriented Design (BOD) (J. Bryson, 2002), could present only high level plan elements if an overview of the system is needed. In the case of an agent designed with BOD, users may prefer to see and become informed about the states of Drives or Competencies but not individual Actions. Other users may want to see only parts of the plan in detail and other parts as a high level overview.

A good implementation of transparency should provide the user with the options described above, providing individuals or potential user-groups with both flexible and preset configurations in order to cater for a wide range of potential users' needs. We hypothesize that the level of abstraction an individual needs is dependent on a number of factors including, but not limited to, the demographic background of the user.

(1) User: We have already discussed the way in which different users tend to react differently to information regarding the current state of a robot. Similarly, we can expect that various users will respond in a similar manner to the various levels of abstraction based on their usage of the system. End-users, especially non-specialists, will prefer a high-level overview of the information available, while we expect developers to expect access to lower level of information.
(2) Type of robotic system: As discussed in our examples above, a multi-robot system is most likely to require a higher level of abstraction, to avoid infobesity of the end-user. A system with a single agent would require much less abstraction, as less data are displayed to its user.
(3) Purpose of the robotic system: The intended purpose of the system should be taken into account when designing a transparent agent. For example, a military robot is much more likely to be used with a professional user in or on the loop and due to its high-risk operation, there is much greater need to display and capture as much information about the agent's behaviour as possible. On the other hand, a robotic receptionist or personal assistant is more likely to be used by non-technical users, who may prefer a simplified overview of the robot's behaviour.

3.1.3. Presentation of information

Developers needs to consider how to present to the user any of the additional information regarding the behaviour of the agent they will expose. Previous studies used visual or audio representation of the information. To our knowledge, there are no prior studies comparing the different approaches.

Autonomous robotic systems may make many different decisions per second. If the agent is using a reactive plan, such as a POSH plan (J. J. Bryson, Caulfield, & Drugowitsch, 2005), the agent may make thousands of call per minute to the different plan elements. This amount of information is hard to handle with systems providing only audio output.

Visualizing the information, i.e. by providing a graphical representation of the agent's plan where the different plan elements blink as they are called, should make the system self-explanatory and easy to follow by less-technical users. Finally, a graph visualization as a means to provide transparency-related information has the additional benefits in debugging the application.

The developer should be able to focus on a specific element and determine why it has been activated by following a trace of the different plan elements called and viewing the sensory input that triggered them.

3.2. *Utility of the system*

So far in this paper we have expanded upon the importance of transparency and the design choices regarding the implementation of it. However, we believe the developer also needs to consider whether implementing transparency may actually damage the utility of a system. (Wortham, Theodorou, & Bryson, 2016a) argues that in certain applications the the utility of an agent may increase with the degree to which it is trusted. Increasing transparency may reduce its utility. This might, for example, have a negative effect for a companion or health-care robot designed to assist children. In such cases, the system is designed without regards for the EPSRC Principles of Robotics, since it is trying to actively exploit the users feelings to increase its utility and performance on its set task.

Another important design decision which effects the system is the physical transparency of the system. The physical appearance of an agent may increase its usability (Fischer, 2011), but also it may conflict with transparency by hiding its mechanical nature. Back in our companionship robot example, a humanoid or animal-like robot may be preferred over an agent where its mechanisms and internals are exposed, revealing its manufactured nature (Goetz, Kiesler, & Powers, 2003).

Discussing the trade-offs between utility and transparency is far beyond the scope of this paper. However, developers should be aware of this trade–off as they design and develop robots.

3.3. *Security and Privacy*

It will become increasingly important that AI algorithms be robust against external, malicious manipulation. For example, a machine vision system in an autonomous weapon can be hacked to target friendly targets instead of hostiles. In line with well-established computer security practices; "security through obscurity is no security", transparency may improve the overall security of a system. Transparency can help us trace such incidents, even as they occur, as we can have a clear, real-time understanding of the goals and actions of the agent.

However, to implement transparency sensitive data captured by the sensors and regarding the internal state of the robot need to be made retrievable, thus, traceable. Such data are prone to be targets of third-party unauthorised hackers and may even be misused by corporations and governments for user profiling, raising privacy concerns. Developers of robotics systems should cater to address such concerns by not only securing any data collected, but also by providing the users of their systems with a clear overview on which data are collected and how the data are used.

While it is beyond the scope of this article to argue and propose methods to develop secure systems, in our view, Artificial Intelligence researchers and developers should start thinking not only improving the performance of their solutions, but also of their security.

4. ABOD3

Current software for AI development requires the use of programming languages to develop intelligent agents. This can be disadvantageous for AI designers, as their work needs to be debugged and treated as a generic piece of software code. Moreover, such approaches are designed for experts, requiring a steep initial learning curve, as they are tailored for programmers. This can be disadvantageous for implementing transparent inspection of agents, as additional work is needed to expose and represent information.

Graph visualisation tools solve the problem described above by allowing the easy design of reactive plans as part of the development of intelligent systems.

ABODE (Brom et al., 2006) 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 and POSHsharp (Gaudl, Davies, & Bryson, 2013). AI designers working with ABODE and a robust planner don't have to worry about programming-related mistakes, as any mistakes are limited to the logical level of the plan.

The main drawback of using the current version of ABODE, similar to other plan editors, is its time-consuming workflow. The plan needs to be saved, imported into the planner, tested in the planner -usually by using log files- and then the developer needs to switch back to ABODE to correct the plan. Creating a complex agent can take numerous iterations of the above described process, making the process take longer than desired.

Currently, we are working towards the development of a new editor, ABOD3, shown in fig. 1 (Theodorou & Bryson, 2016). It allows the graphical visualisation of BOD-based plans, including its two major derivatives: POSH and Instinct Wortham, Gaudl, and Bryson (2016). 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 time required to develop an agent. This allows the development and testing of plans from a same application.

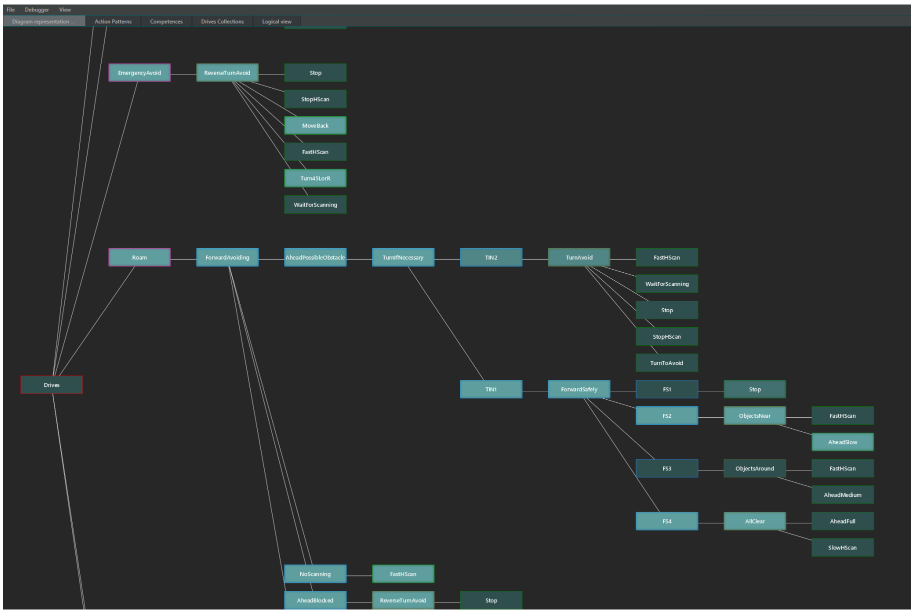

Figure 1. The upcoming ABOD3 editor in debug mode

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-house, using Java the software development language and the JavaFX GUI-framework, to ensure cross-platform compatibility. An API allows adding support for additional BOD derivatives, other than those already supported.

A public Application Programming Interface (API) allows the editor to connect with planners, presenting debugging information in real time. Plan elements flash as they are called by the planner and glow based on the number of calls they were used by the agents. Plan elements without any recent calls start dimming down, over a user-define interval, until they return back to their initial state offering backtracking of the calls. Sense information and progress towards a goal are displayed. Finally, videos of the agent in action are supported. ABOD3 can render these videos concurrently with display of pre-recorded log-files.

The editor provides a user-customisable user interface (UI) aimed at supporting both the development and debug of agents. The intention is to provide a platform to facilitate agent design in accordance with the good practices established in the previous section. Plan elements, their subtrees, and debugging-related information can be hidden, to allow different levels of abstraction and present only relevant information. The graphical representation of the plan can be generated automatically, and the user can override its default layout by moving elements to suit his needs and preferences. Users and fellow developers can, by editing a simple CSS file, change the color scheme of the editor. Developers can provide additional views by implementing our UI API.

We plan to continue developing this new editor, implementing debug functions such as "fast-forward" in pre-recorded log files and usage of breakpoints in real-time. Moreover, we will enhance its plan design capabilities by introducing new views, to view and edit specific types of plan-elements and through a public alpha testing to gather feedback by both experienced and inexperienced AI developers.

The simple UI and customisation allows the editor to be employed not only as a developer's tool, but also to present transparency information to the end-user. We have already used ABOD3 in experiments to determine the effects of transparency on the mental models formed by humans (Wortham, Theodorou, & Bryson, 2016b). Our experiments consisted of a non-humanoid robot, powered by the BOD-based Instinct reactive planner (Wortham, Gaudl, & Bryson, 2016). The technology used to construct the experimental system was found to be reliable, robust, and straightforward to use. Despite using an early pre-alpha version of ABOD3 in the experiment, it confirmed its usefulness both as a tool during robot plan debugging and to provide transparency information to untrained observers of the robot. ABOD3 visualisation of the robot's intelligence does indeed make the machine nature of the robot more transparent, as subjects can show marked improvement in the accuracy of their mental model of a robot, if they also see an accompanying display of the robot's real-time decision making as provided by ABOD3. The IDE was able to display a sufficient amount of information to casual observers, with and without prior knowledge of robotics. Its ability to hide complex subtrees, while providing a high level overview, proved valuable at displaying the plan in a single monitor and to inexperienced users. We concluded that providing transparency information by using ABOD3 does help users, regardless of their demographic background, to understand the behaviour of the robot and calibrating their expectations.

5. Conclusion

In this paper we have reviewed the concept of transparency, both as used in the EPSRC Principles of Robotics, and as used elsewhere in the AI literature. We have determined that the Principle requires the accessibility of a robot's ordinary decision-making, not only in situations of accountability, collaboration, or cognizant error-detection. Artificial intelligence is defined by the fact it is authored, and as such needs never be the kind of mystery evolution provides us.

We believe the implementation and usage of intelligent systems which are fundamentally transparent can help not only with debugging AI, but also with its public understanding, hopefully removing the potentially-frightening mystery around "why that robot behaves like that". Transparency should also allow a better understanding of an agent's emergent behaviour. In this paper we redefined transparency as an always-available mechanism able to report a system's behaviour, reliability, senses, and goals. Such information should help us understand an autonomous system's behaviour.

Further work is needed to test whether our beliefs are well-founded, and to establish good practices regarding the implementation of transparency within the robotics community. Considering the potential benefits of transparent systems, we strongly suggest the promotion of this key principle by research councils such as the EPSRC, and other academic communities.

References

Boden, M., Bryson, J., Caldwell, D., Dautenhahn, K., Edwards, L., Kember, S., ... Winfield, A. (2011). *Principles of robotics.* The United Kingdom's Engineering and Physical Sciences Research Council (EPSRC). (web publication)

Brom, C., Gemrot, J., Bída, M., Burkert, O., Partington, S. J., & Bryson, J. J. (2006, November). POSH tools for game agent development by students and non-programmers. In Q. Mehdi, F. Mtenzi, B. Duggan, & H. McAtamney (Eds.), *The ninth international computer games conference: AI, mobile, educational and serious games* (pp. 126–133). University of Wolverhampton.

Bryson, J. (2002). The behavior-oriented design of modular agent intelligence. In *System* (Vol. 2592, pp. 61–76). doi:

Bryson, J. J. (2010). Robots should be slaves. In Y. Wilks (Ed.), *Close engagements with artificial companions: Key social, psychological, ethical and design issues* (pp. 63–74). Amsterdam: John Benjamins.

Bryson, J. J. (2012). The Making of the EPSRC Principles of Robotics. , *133*(133), 14–15.

Bryson, J. J., Caulfield, T. J., & Drugowitsch, J. (2005, October). Integrating life-like action selection into cycle-based agent simulation environments. In M. North, D. L. Sallach, & C. Macal (Eds.), *Proceedings of Agent 2005: Generative social processes, models, and mechanisms* (pp. 67–81). Chicago: Argonne National Laboratory.

Dzindolet, M. T., Peterson, S. A., Pomranky, R. A., Pierce, L. G., & Beck, H. P. (2003). The role of trust in automation reliance. *International Journal of Human Computer Studies*, *58*(6), 697–718. doi:

Fischer, K. (2011). How People Talk with Robots: Designing Dialogue to Reduce User Uncertainty. *AI Magazine*, *32*(4), 31–38. doi:

Gat, E. (1992). Integrating planning and reaction in a heterogeneous asynchronous architecture for controlling mobile robots. In *Proceedings of the tenth national conference on artificial intelligence (aaai92).*

Gaudl, S., Davies, S., & Bryson, J. (2013). Behaviour oriented design for real-time-strategy games: An approach on iterative development for STARCRAFT AI. *Foundations of Digital Games Conference*, 198–205.

Goetz, J., Kiesler, S., & Powers, A. (2003). Matching robot appearance and behavior to tasks to improve human-robot cooperation. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, 55–60. doi:

Groom, V., & Nass, C. (2007). Can robots be teammates? *Interaction Studies*, *8*(3), 483–500. doi:

Kahn, P. H., Severson, R. L., Kanda, T., Ishiguro, H., Gill, B. T., Ruckert, J. H., ... Freier, N. G. (2012). Do people hold a humanoid robot morally accountable for the harm it causes? *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI '12*(February 2016), 33. doi:

Kim, T., & Hinds, P. (2006). Who should I blame? Effects of autonomy and transparency on attributions in human-robot interaction. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, 80–85. doi:

Lyons, J. B. (2013). Being Transparent about Transparency : A Model for Human-Robot Interaction. *Trust and Autonomous Systems: Papers from the 2013 AAAI Spring Symposium*, 48–53.

Novikova, J., & Watts, L. (2014). Towards artificial emotions to assist social coordination in HRI. *International Journal of Social Robotics*, 1–12. Retrieved from `http://dx.doi.org/10.1007/s12369-014-0254-y` (in press) doi:

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, *39*(2), 230–253. doi:

Pöppel, E. (1994). Temporal mechanisms in perception. *International Review of Neurobiology*, *37*, 185–202.

Stumpf, S., Wong, W.-k., Burnett, M., & Kulesza, T. (2010). Making intelligent systems understandable and controllable by end users. , 10–11.

Tan, Y., & Zheng, Z.-y. (2013, 3). Research Advance in Swarm Robotics. *Defence Technology*, *9*(1), 18–39. doi:

Theodorou, A., & Bryson, J. J. (2016). Abod3: A graphical visualization and real-time debugging tool for bod agents.. Retrieved from `http://opus.bath.ac.uk/53506/`

Tullio, J., Dey, A. K., Chalecki, J., & Fogarty, J. (2009). How it works: a field study of Non-technical users interacting with an intelligent system. *SIGCHI conference on Human factors in computing systems (CHI'07)*, 31–40. doi:

Wang, L., Jamieson, G. a., & Hollands, J. G. (2009). Trust and reliance on an automated combat identification system. *Human factors*, *51*(3), 281–291. doi:

Wortham, R. H., Gaudl, S. E., & Bryson, J. J. (2016). Instinct : A Biologically Inspired Reactive Planner for Embedded Environments. In *Proceedings of icaps 2016 planrob workshop.*

Wortham, R. H., Theodorou, A., & Bryson, J. J. (2016a). Robot Transparency , Trust and Utility. In *Asib 2016: Epsrc principles of robotics.*

Wortham, R. H., Theodorou, A., & Bryson, J. J. (2016b). What Does the Robot Think? Transparency as a Fundamental Design Requirement for Intelligent System. In *Ijcai-2016 ethics for artificial intelligence workshop.*