A version of this article appears in JETAI special issue on Autonomy Control Software.

Autonomy Control Software

Henry Hexmoor
Department of Computer Science
University of North Dakota
Grand Forks, ND 58202

David Kortenkamp
NASA Johnson Space Center -- ER2
Houston TX 77058
hexmoor@cs.und.edu kortenkamp@jsc.nasa.gov

1 Introduction

Human beings have always been fascinated with the feeling that a living creature or a device is able to carry out a function that one would ordinarily perform by himself or herself. The methods of employing the creature and ways of designing and using the device vary but to the extent they can do the intended job they are said to be autonomous. An autonomous device is one that can do a job by itself. A thermostat or a sun-dial is autonomous: the former has moving parts, but more or less passively regulates temperature and the latter is completely inert, but casts a shadow onto a surface that can be read as time of day. An automobile uses its own power to move so it is autonomous with respect to mobility. A carrier pigeon delivers messages while avoiding dangers and may interleave delivery with resting or other diversions.

The quality of autonomy for a device or creature (henceforth called an agent) is an attribution by the designer or an observer. From the standpoint of the performing agent, there is usually no need to consider the quality of autonomy unless the device or the creature has certain options such as delegating or ignoring the task. With a task shared among several agents, agents can use social factors to commit to a level of effort to the task as well as synchronization with others [Castelfranchi, 1995, 1997]. The social factors are many such as those stemming from obligation, trust, authority, and conventions. In addition to social factors, an agent may have its own endogenous reasons for selecting a particular level of autonomy with respect to a task that may affect others.

For the purposes of this special issue, we define Autonomy Control Software as a class of software systems that operate with some (possibly varying) level of autonomy in regulating a hardware system's outputs in a continuous manner. Many continuous and long lived operations in space systems, automated factories, and robotics are example domains where autonomy control software is required [Zetoka, 2000]. The quintessential example of autonomy control software is the controller for a mobile robot (see [Kortenkamp et al 1998] for some case studies of autonomous robots), where sensory inputs must be continuously evaluated with respect to goals in order to determine appropriate actions. Increasingly, autonomous control software is being used in non-robotic applications. [Williams and Nayak, 1996] give examples of "immobots", which do not move, but must still process continuous sensory data and achieve goals. [Muscettola et al, 1998] describe one of the more complete and ambitious of recent autonomy control software systems. Of course, there are numerous other examples, from robots to factories to spacecraft to refineries to smart houses. This special issue looks at some of the advances and challenges in autonomy control software.

The last several years have seen enormous strides in the use of autonomous control systems in real-world, complex situations. Examples include the Deep Space 1 spacecraft launched by NASA in 1999 [Muscettola et al 1998], the RoboCup competition [Stone, 2000], and autonomous control of parts of a 90-day life

support test with four crew members [Schreckenghost et al, 1998] and autonomous driving across much of the entire United States (http://www.cs.cmu.edu/Groups/ahs/). These advances have all built upon research into architectures, planning, uncertainty, real-time algorithms and learning. Most importantly, however, these advances have come by integrating a variety of techniques into coherent autonomous control systems that are reliable, adaptable, taskable and adjustable.

2 Definition and Principles of Autonomy

The word "autonomous" is a Greek word for self-governance. In the context of intelligent software, we are literally at the starting point of defining autonomy. It is hard to define autonomy since the colloquial use of it connotes a sense of volition, independence, and self-sufficiency. Below is a partial list of relevant parameters as ingredients for a specification of autonomy. Autonomy is best defined locally in the sense that in a given domain, relevant parameters can be selected and used to gain a sense of autonomy. Examples of these parameters include the:

- Agent's ability with respect to tasks and the myriad of factors that affect its ability such as uncertainties of the world or the constituent components of ability
- Agent's preference with respect to tasks
- Agent's role in a group
- Agent's level of contribution to a team decision
- Agent's relationship to individuals in the team and to the team as a whole. This is a class of parameters that include a large list such as as authority, friendship, cooperation, altruism, etc.
- Agent's responsibility
- Agent's commitment with respect to tasks
- Agent's reasoning
- Agent's emotional makeup and stance such as self-confidence, obedience, boldness, fears

We state several principles that solidify the notion of autonomy. First, autonomy with respect to a task is meaningful only when the agent possesses a nontrivial ability to perform the task or to make it happen by ordering others to do it. Autonomy does not make sense when the agent is not capable of performing the task. Second, autonomy is a relative term. An agent has some autonomy with respect to a task within some constraints such as the goal priority. Third, autonomy can be due primarily to endogenous sources (internal to the agent) or exogenous sources of interaction with other agents.

So far we have been discussing treatment of concept of Autonomy explicitly; however, complex control systems may treat qualities that provide autonomy implicitly. Some of these qualities are Timeliness, Purposefulness, Robustness and Failure Tolerance, Coherence, Flexibility and Adaptability, and Reliability, Interruption, and Learning and Adaptation. We will not discuss implicit notion of Autonomy, nor will give the relation between these qualities and autonomy. We envision that we will see much work in quantifying these qualities as a step toward empirical evaluation of agent systems [Hexmoor, Lafary, and Trosen 1999].

A version of this article appears in JETAI special issue on Autonomy Control Software.

In the remainder of this article we will discuss the challenges related to autonomy, make a few predictions for future, review the articles in this special issue, and then draw a few conclusions.

3 Challenges

There are enormous challenges in the area of building autonomous control systems. Some of these challenges are unique to autonomous control systems as a subset of more general autonomous agents. The papers in this special issue address a number of these challenges, including:

- **Varying the level of autonomy**: Often fully autonomous control systems are not possible or not desirable. For example, in crewed spacecraft the crew will want to be able to work with and possibly override the autonomous control system [Dorais et al, 1998]. Another example is when the autonomous control system itself adjusts its autonomy in response to other agents or to the situation. This capability is often called adjustable autonomy or human-centered autonomy. This capability does not imply an overall loss of autonomy, for example a musician in an orchestra adjusts her autonomy to play within the confines of the score but is not less autonomous. In this issue, the paper by **Suzanne Barber**, **Anuj Goel**, and **Cheryl Martin** defines autonomy categories such as Command-driven, Consensus, and Locally autonomous/Master. Within these broad categories agents dynamically sense their level of contribution to a team's decision making as their autonomy level. Also in this issue, the short paper by **Rino Falcone** and **Cristiano Castelfranchi** relate the notion of autonomy to delegation and trust.

- **Dealing with uncertainty**: Because an autonomous control system is defined as controlling hardware there is inherent uncertainty in both sensor input and command output. Dealing with this uncertainty is a major component of an autonomous control system. In this issue, the paper by **James Gunderson** and **Worthy Martin** looks at uncertainty with respect to planning and execution. They focus on experimentally determining how different kinds of uncertainty can have different effects on goal satisfaction. Agent's introspection about its abilities and the underlying uncertainties is used in determining an appropriate level of Autonomy.

- **Architectures and verification**: Architectures provide framework within which to program autonomous control systems. They can also allow for easier verification of the resulting autonomous control system. Verification is a significant hurdle in fielding autonomous control systems, especially in mission critical or life critical applications. An agent's changing level of autonomy will affect its predictability, i.e., it will become more non-deterministic. How can we make sure the agent's actions are safe? Extensions of techniques such as [Gordon, 1998] will be useful. The issue of architecture has already been addressed in a previous special issue of this same journal (Volume 9, Numbers 1 and 2). Inclusion of methods for adjusting autonomy affects design of agent software. The principles of design have been studied and reported in articles that discuss architectures. In this special issue we have an article by **Joanna Bryson** that looks at analyzing different architectures as design methodologies. We have an article by **Paul Scerri** and **Nancy Reed** that takes an engineering approach to autonomous control architectures and presents a real-world application of them. In her article in this issue, **Lynne Parker** provides a list of application specific metrics in applying the ALLIANCE architecture. She hopes that by averaging these metrics across multiple application domains, high-level metrics can be developed that quantifies application-independent architectural characteristics.

- **Learning**: If an autonomous control system is to regulate a hardware system for a lengthy period of time (say years) with as little human input as possible then the system will need to learn. Learning will be necessary for several reasons: First, the underlying hardware system being controlled will

change over time due to degradation or drift; second, the environment in which the autonomous control system is situated will change over time; and finally, the autonomous control system designers may not have provided completely accurate models or knowledge due to lack of information. In this special issue the paper by **Larry Pyeatt** and **Adele Howe** looks at learning in the context of a robotic architecture. They describe experiments that test their architecture for reliability and generalization. Although they do not directly include the concept of autonomy, they discuss learning actions while preserving reliability.

- **Real-time autonomous control**: One significant aspect of autonomous control systems that is often unique from other autonomous systems is that there are deadlines by which actions must be taken. This is true of robotics, spacecraft, factories, refineries, etc. Many of the key research questions in the area of real-time autonomous control are described in a survey article by Musliner et al (1995). In our special issue, the problems of real-time control are discussed in the paper by **Maurizio Piaggio**, **Antonio Sgorbissa**, and **Renatto Zaccaria**, which addresses the quality of timeliness through a presentation of scheduling concurrent robotic activities.

5. Human Tolerance of Autonomy

As autonomous control systems are increasingly used in applications such as smart homes, smart vehicles and medical applications it will be necessary for humans to have a degree of confidence and acceptance of them. There are several ways to achieve this. First, the autonomous control system must be tested and verified. Testing and formal verification of autonomous control systems is an active research area and is being pushed by the use of autonomous control systems in mission critical applications. Second, the motivations and actions of the autonomous control system must be understandable to a non-expert human user. This involves research into user interfaces and user interaction. Finally, the autonomous control system must be adjustable; that is, it must allow for humans to intervene and take more or less control depending on the situation.

6. Conclusion

Autonomous control systems pose their own unique set of problems within the autonomous agents domain. This special issue looks at many of these problems and at the current state-of-the-art in autonomous control systems.

References

Castelfranchi. C. Modeling Social Action for AI Agents. (Invited paper) 1997. In International Joint Conference of Artificial Intelligence **- IJCAI'97**, Nagoya, August 23-29, pp.1567-76)

Castelfranchi, C., 1995. Guarantees for Autonomy in Cognitive Agent Architecture, in N. Jennings and M. Wooldridge (eds.) Agent Theories, Architectures, and Languages (**ATAL-95**), Heidelberg, Spinger/Verlag.

Dorais, G. and R. P. Bonasso, D. Kortenkamp, B. Pell and D. Schreckenghost, Adjustable Autonomy for Human-Centered Autonomous Systems on Mars, Mars Society Conference, 1998.

Gordon, G., 1998. Well-behaved Borgs, Bolos, and Bersekers, In the Proceedings of the 15th International Conference on Machine Learning (**ICML-98**).

Hexmoor, H. 2000. Case Studies of Autonomy, In the proceedings of **FLAIRS 2000**, Orlando, FL.

A version of this article appears in JETAI special issue on Autonomy Control Software.

Hexmoor, H., Lafary, M., Trosen, M., 1999. Towards Empirical Evaluation of Agent Architecture Qualities, Agent Theories, Architectures, and Languages (**ATAL-99)**, Orlando, Florida.

Muscettola, N., P. P. Nayak, B. Pell and B. Williams, Remote Agent: To Boldly Go Where No AI System Has Gone Before, Artificial Intelligence, 103(1), 5-47, 1998.

Musliner, D., J. A. Hendler, A. K. Agrawala, E. H. Durfee, J. K. Strosnider, and C. J. Paul, The Challenges of Real-Time AI, *IEEE Computer*, 28(1), 1995.

Schreckenghost, D. K., D. Ryan, C. Thronesbery, R. P. Bonasso and D. Poirot, Intelligent Control of Life Support Systems for Space Habitats, Proceedings of the Conference on Innovative Apllications of Artificial Intelligence, 1998.

Stone, P. *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*, MIT Press, Cambridge MA, 2000.

Williams, B. C., and P. P. Nayak, Immobile Robots: AI in the New Millennium, AI Magazine, 17(3), 16-35, 1996.

Zetoka, P., 2000. Satellite Cluster Command and Control, In **IEEE Aerospace Conference**, Montana. (http://www.aeroconf.org/)