Community Driven Artificial Intelligence Development for Robotics

Csaba Kertész and Markku Turunen
Tampere Unit for Computer-Human Interaction (TAUCHI), University of Tampere, Kalevantie 4, Tampere, Finland

1 RESEARCH PROBLEM

The robotics have been advanced greatly along with the computer technologies since the 20th century and the improvements in hardware components enabled the development of complicated mobile robots. Grey Walter and Valentino Braitenburg (Iizuka, 2004) developed simple wheeled robots to traverse their environment with light sensors, but these systems were inflexible and non-reprogrammable. To overcome the closed world assumption in planner systems in the 1970s, the behavior-based robotics were invented by Rodney Brooks (Brooks, 1991) to shift the emphasize to sensing and acting within the environment. The recent consumer robots have been followed Brooks' architecture, but the increasing behavior complexity of these systems is hard to maintain with the human resources in a company. The online community-based artificial intelligence development can solve this problem by distributing the efforts across multiple parties (Hibbard, 2008), however, comprehensive studies have not been not available about the requirements and good practices for this process to the best knowledge of the authors.

AIBO was a brand of entertainment robots, but Sony discontinued in 2006. It was a very successful robot platform both in research labs and consumer market. An active community of enthusiastic AIBO owners is still around the world with their own repair services for hardware faults and damages what shows how strong is this platform. This PhD research attempts to create a special online community for building a new artificial intelligence (AI) software for Sony ERS-7 robots (Figure 1) as a case study to explore new ways of modern AI development.

Figure 1: Sony ERS-7 robot.

The following question is addressed in this research:

- How can an online community be built and involved to develop an artificial intelligence for robotics?

2 OUTLINE OF OBJECTIVES

Two objectives are linked to online community involvements in artificial intelligence creation:

1. Find working strategies to build a successful community for AI development with the open source model, crowdsourcing and other state of the art methods.
2. Search for work distribution methods to involve community members for developing certain AI parts according to their skills.

3 STATE OF THE ART

Although the online AI forums were examined earlier to understand the owner attitudes toward their robots (Friedman et al, 2003; Kahn et al, 2004) and the performance of distributed volunteer project techniques was researched (Krafft, 2005; Porter et al, 2006), but the literature lacks a specific study to cover AI development with online communities.

Open source software (OSS) is a classic case of community based innovation with more than 20 years-long history. An OSS project consists of a volunteer group of geographically distributed experts who contribute to a software project based on internet communication. The normal users are the largest fraction of a community and some active lead contributors drive the new value creation. The software industry has been recognized the opportunity to organize online communities around

Kertész C. and Turunen M..
Community Driven Artificial Intelligence Development for Robotics.

their products where the people are involved in discussion of new innovative ideas (Franke and Shah, 2003), solving product related problems or getting product design concepts (Von Hippel and Katz, 2002).

Yan and Wang (Yan and Wang, 2013) proposed a Crowdsourcing Based Community Development (CBCD) model for a Chinese software company (Huazisoft) to involve online community members into the implementation and testing phases of several commercial projects. Although they run a selection procedure for candidates, the community team was simply hired in their free time for lower wages similar to outsourcing. Their model had copyright risks and the community members sometimes left the project because of their loose connections with the company.

The OSS is an appealing choice for the research projects. OpenCog, an open source AI framework aims to develop an artificial general intelligence as an emergent phenomenon of multiple subsystems (Hart and Goertzel, 2008). This project was started six years ago and their code repository contains around 148000 lines of C++ codes. Since their system components rely on complex theories, the contribution is not a scope for anybody, but AI researchers and professionals. The free Stratagus engine (Ponsen et al, 2005) is another specialized software for AI researchers and game developers to create real time strategy games. The software of a successful humanoid robot was open sourced by the Aldebaran Robotics in 2011 (Nao, 2011), but the community contributions to Nao's AI engine have not been analyzed by researchers yet.

The crowdsourcing can obtain services, ideas or contents by collecting contributions from a large community. The Amazon Mechanical Turk service is used by companies and researcher teams to perform simple microtasks in parallel with unqualified people for pecuniary rewards although some critical thoughts were raised that using this method for scientific research with such low payments is not ethical. One of the most successful type of voluntary participation for science is to give the unused CPU and GPU cycles of a desktop computer for scientific computing when it is left in idle. The popularity of this option comes from the easy setup steps and low maintenance needs for the regular users. Over 60 projects operate under the umbrella of BOINC infrastructure from disciplines of astronomy, physics, mathematics, medicine, biology and chemistry (Anderson, 2004). The Zooniverse is a citizen science web portal run by Citizen Science Alliance. Volunteers contribute for free to projects from astronomy, ecology, cell biology, humanities and climate science (Simpson et al, 2014). The latter organization shows a model how people can be involved to sacrifice their leisure time for scientific research driven by enthusiasm and field of interest.

Some examples above represented that the OSS model is very popular and made some traction in some research fields, but the current PhD research is essential because the development involves both technical and non-technical people into the artificial intelligence creation. The open source model and the transparency of the AI development ensure that the design can be examined for flaws by as many intelligent humans as interested in. These are key advantages for open source AI development (Hibbard, 2008) as natural worries can arise in the humans against dangerous robots. The crowdsourcing is an emergent field for several research fields to get great contributions from the public masses, and combined with OSS model, an interesting experiment will be done during this PhD research to build a new AI.

4 METHODOLOGY

4.1 Open Source Software Model

The AI development mus implement an open and transparent process mentioned in the next section as people contribute for OSS with higher chance than to a project with a more restrictive license (Crowston et al, 2012).

Although the practice of open source model has been increased in the research software development for artificial intelligence fields like Dlib in machine learning (King, 2009) and OpenCV in computer vision (Bradski and Kaehler, 2008) as well as this model has obvious advantages, but several empirical studies revealed some challenges. Since the OSS projects emerge from individual needs, most of them are developed by small, isolated teams. By analyzing the projects on sourceforge.net, a study found that many are inactive or never publish any releases and the majority have only two or less developers (Christley and Madey, 2007). The openhub.net claims that half of the OSS projects in their database are individual efforts. Consequently, these projects are mainly developed by one or a few persons rather than by a dynamically changing open community. When the lead users stop the contribution after losing interest or personal reasons, the project can go in hibernation easily until a new developer is attracted to continue the leadership.

4.2 Community Involvement

A review paper (Crowston et al, 2012) examined the heterogeneous OSS contributor motives and they concluded that the reputation and the rewards are important extrinsic motivations while the fun, sharing and learning opportunities are intrinsic. The motives are changing over time because once the personal needs are fulfilled with the initial development, the participation must transform into a hobby to keep up the engagement. Other community properties are also essential to drive a successful project like leadership, interpersonal relationships and taking advantage of the cultural diversity.

The users spend 1.7 hours with an OSS project daily on average, the project leaders invest more time and bug fixing takes less hours (Crowston et al, 2012). The remunerated developers contribute more than others without any material compensations and their commitment is less dependent on the intrinsic motivations.

A community must be engaged to contribute for a project and the key contributors need to know several important factors to lead the project for a longer period:

- What is the size of the community?
- Which capabilities of the members are suitable for contribution?
- Which goals can motivate the members to create new value?
- What kind of actions can keep the community active and alive in long-term?

These questions can be answered by executing questionnaires to get know the community and the existing experiences of the OSS model must be applied in the community building.

The primary meeting point of the AIBO enthusiasts is the forum on aibo-life.org. The PhD research targets the creation of a special community inside aibo-life.org to build a new artificial intelligence for Sony ERS-7 robots. The following ideas are under consideration to reach this goal:

- Run a questionnaire among the forum members to get a view about the community of aibo-life.org and future expectations for a new Sony ERS-7 software.
- Develop a minimal AI for Sony ERS-7 to get the initial attention of the forum members.
- Open and transparent AI development.
- Build reputation among the forum members.
- Quick responses to problems, requests in the forum.
- Get a dedicated area inside the forum where problems of the new AI can be discussed.
- Involve members in the development with tasks, polls, competitions or brainstorming.
- Keep the project going for a longer time.
- Analyze other online robotics communities and integrate their ideas.
- The first author will try to get into the Google Summer of Code[1] (GSoC) in 2016 as a mentor and check how effective is a student contribution.

As it can be seen above, multiple tasks must be accomplished to build a successful community.

4.3 Modern Software Development Tools and Practices

Existing software solutions have been adapted to the robot dog, but many functions had to be reimplemented. A challenge in this situation that the AIBO owners must be attracted with never-seen features compared to the official AIBO Mind software to gain interest for contribution.

The software industry invented a number of technologies and processes to ensure the quality of complex applications, but these techniques are not often used in the research labs. When a robot is developed with many behaviors, any small change can trigger unexpected failures in the operation and this problem will be addressed during the PhD research with good practices from the software industry:

- Task list with priorities and estimations.
- Modular software design.
- Tests for the core modules.
- Test the robot AI as a complete system.
- Follow some source code metrics over time.
- Check the source code with static code analysis tools.
- Use tools to detect memory handling errors.
- Check the compilation for each supported platform.

4.4 Crowdsourcing

The first author believes in a more strict meaning of crowdsourcing than the CBCD model (Yan and Wang, 2013), namely, driving a team selection process in a community for a specific job is far from a distributed task execution. Therefore, the crowdsourcing in this research is a voluntary data

1 https://www.google-melange.com

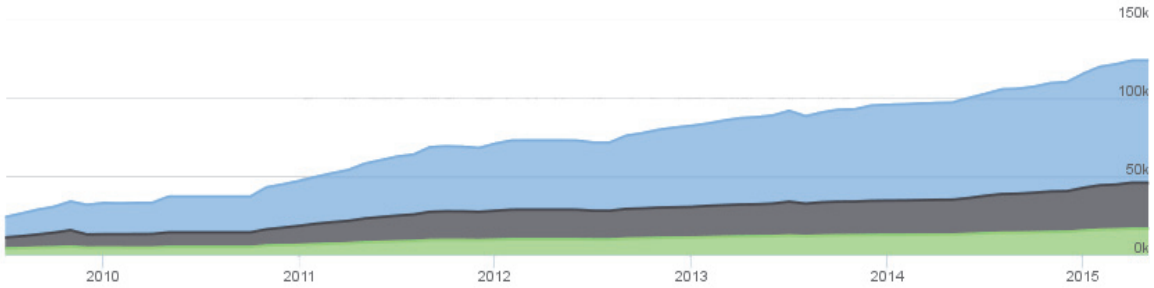

Figure 2: History of source code lines (blue), comments (dark gray) and blank lines (green) in the AiBO+ project. The diagram was created by the Open HUB service.

contribution of sample data to solve machine learning problems.

The main idea for this direction is that the robot owners will run the new AI software on their robot dogs to record samples for different problems and the collected data are shared with the author to carry out scientific analysis. The user will get a notification on an application user interface when these contributions are uploaded, but reward mechanisms like point systems are not under considerations yet.

5 EXPECTED OUTCOME

This PhD research will be successful if multiple AIBO community members can be involved in the artificial intelligence development and their work will be included into the AiBO+ project. An exceptional result would be to find members with long standing commitments which would ensure the sustainable development after PhD studies.

The long-term experiences of this study can be used in artificial intelligence development for new products in the future to distribute the efforts between company employees and online volunteers. This work sharing is required because of the high complexity of the robots nowadays. The expected outcome is the case study of practical processes and methods to involve online community members with diverse skills for build a new AI.

6 STAGE OF THE RESEARCH

6.1 Preliminary Work

An established software basis and a minimal AI engine had to be developed before this PhD research to have enough offering for community building.

The first author started the work with a Sony ERS-7 robot five years ago and he registered the AiBO+ project[2] on the sourceforge.net. After examining the available developer options, the Open-R SDK with C++ support was chosen as development platform which was released by the Sony with gcc toolchain for low level AIBO programming. Although a Open-R developer has full access to the sensors, motors and wireless connection, the Sony did not include any higher level functions (e.g. face recognition, locomotion or environment mapping). An other problem was the old gcc 3.3 in the original Open-R SDK what made cumbersome to compile modern C++ sources without a proper compiler support for 1998 ISO C++ standard.

An upgrade to the Open-R toolchain was necessary to solve these problems, but the Sony robots do not have a Unix-like system and the application loader is encrypted into the Aperios environment. After five months of trial and error, the gcc suite was updated to version 4.1.2 (Kertész, 2013) which can build state of the art software like the latest versions of OpenCV (Bradski and Kaehler, 2008) and Boost (Schaeling, 2014).

The lower level functions were implemented in the past five years:

- AIBO (Open-R SDK), Windows (MinGW), Ubuntu Linux (gcc, clang) and Android (NDK) support.
- Wireless communication between a robot and a PC to transfer camera image, sensor data and log messages. The received data is recorded for further analysis and testing (Kertész, 2010).
- A behavior-based architecture was developed to control the motor joints, LEDs and implement higher level functions (Kertész, 2012).

2 http://aiboplus.sf.net

- Floor surface detection was explored based on accelerometer and paw sensors (Kertész, 2014).
- An Ubuntu PPA[3] was set up to grant free and easy access to the updated Open-R SDK and AiBO+ components.
- OpenCV and many machine learning methods (classifiers and regression methods) are integrated.
- An AMD FX 8350-based eight-core desktop PC was built to run machine learning tasks.

6.2 Community Building

The Nao humanoid robot of Aldebaran Robotics has an online community which was created for similar purposes like this PhD research for Sony ERS-7. Inasmuch as there is no study about the community experiences of the Aldebaran Robotics, the first author will contact the company to get some knowledge by interviewing their employees.

The author run a questionnaire in the forum members of aibo-life.org in April 2014 to get an impression about the present state of the AIBO community. 50 responses were received and the preliminary results suggest the following:

- Majority of the users do not have any technical or engineering background thus they are not potential contributors to develop the C++ sources.
- Half of the users are interested in to create new tricks for the robot if a motion editor software is available. These contributions require little technical knowledge, but more creativity, imagination and patience.

Taken into account these points, the author worked on a first public release of the AiBO+ project to attract the community whose members do not care about the technical details, but the visible AI capabilities and connectivity options. This initial offering was implemented with these features:

- Transitions between basic poses.
- Locomotion: walk, turn.
- Turn over from upside down.
- Floor surface recognition.
- Happy responses for petting.
- Angry responses for poking.
- Avoid downward stairways and objects in front.
- Intelligent LED brightness levels and volume level control.
- A dedicated application under Windows, Android and Ubuntu Linux.

The artificial intelligence can provide these functions based on 71000 lines of C++ codes (Figure 2) and 1400 lines of Qt Modeling Language (QML) as of April 2015. According to the analysis of openhub.net, the AiBO+ project has an established, mature codebase and an estimated 20 years of effort with COCOMO model. Nevertheless, the first author is the sole developer of the AiBO+ hosted on sourceforge.net and the project can go in hibernation when he stops the contribution. The sources are developed in a public Git version control from the beginning to minimize this risk and C++ sources are well documented, 29% of the code lines are comments.

An earlier review paper (Crowston et al, 2012) found that the OSS developers are predominantly from USA and Europe, and similarly, the questionnaire for the English speaking AIBO community found that vast majority of the members live in the Western countries (60% in North-America, 32% in Europe), a small percentage is in Asia and Australia. This homogeneous cultural background can ease the planning of a community development according to Section 4.2. Since most Japanese people do not speak English at all, but they have great traditions in the robotics, a Japanese version of the survey will be executed in the future to compare the differences between Japanese and Westerners.

Some results can be concluded about the downloads of the initial offering. Although the SourceForge site reports 11 downloads for both the Windows installer and the memory stick image after two weeks, the installations can not be followed for Linux because of the Ubuntu PPAs lack these statistics. An interesting finding was that the Windows installer was mostly downloaded from Asia (45%) and even Europe (36%) predominated the American downloads (18%). Contrary to these results, the memory stick image was downloaded mainly from Europe (66%) and North-America (34%), but nobody downloaded from Asia. The AiBO+ Client application for Android was downloaded from Google Play to a few devices (Figure 3) and the installations are divided by half between Europe and North-America. These low download numbers can be explained so that the robot owners do not put much efforts to set up the wireless network for AIBO because the official AIBO Mind software requires the connection only

3 https://launchpad.net/~csaba-kertesz/+archive/aiboplus

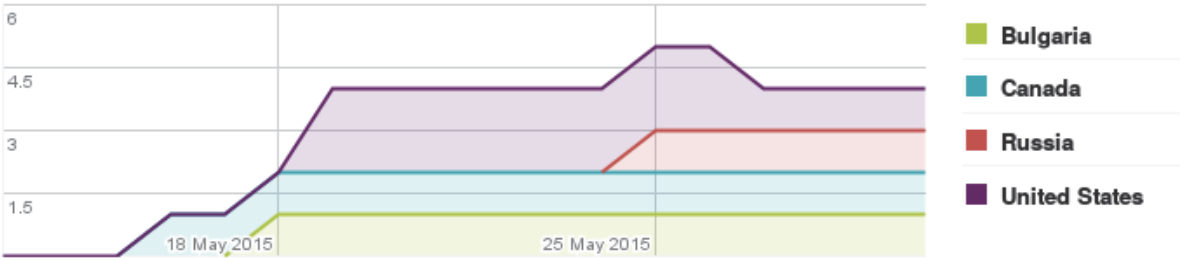

Figure 3: Installation history of AiBO+ Client on Google Play.

for extra features. Secondly, the current AiBO+ software is not released for Apple electronics (iOS, Mac OSX) though these products are popular in North-America where the majority of the AIBO owners live. The iOS support was a strong wish in the responses to the AIBO questionnaire, but the Qt Framework, which is the software basis for the graphical user interface, is not mature on iOS yet. To summarize, the initial download counts are good in overall, compared to the small size of the AIBO community.

Despite the small amount of users, the AIBO forum members have been involved in several activities so far:

- An AiBO+ compatible version of a community developed motion editor software (Skitter[4]) for Sony robot dogs.
- Polls were executed before the AiBO+ initial software release about some software features and how the owners use their Sony ERS-7s.
- Voice acting for the AiBO+ software.
- 3D printing accessories (Figure 4) for the Sony robots.

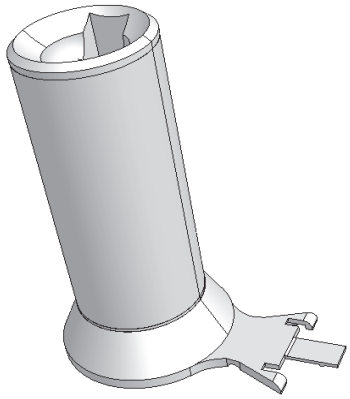

Figure 4: 3D printing model of Sony ERS-7 station pole[5].

6.3 Software Development Practices

Usually, the OSS projects have TODOs or they use their bug tracker to have a list of feature requests with severity levels. The first author schedules the tasks of AiBO+ with an agile management tool (PivotalTracker). The bugs and upcoming features are represented as stories in one week-long sprints, relative points show the time estimation for each story and the story listing order determinates the priorities.

The modular software design is necessary to maintain a growing project and add new features with minimal efforts. The AiBO+ is coded in C++ language and the components are split by functionalities:

- libmindcommon: This library contains the common functions for the other modules.
- libmindaibo: The behavior based robot brain for AIBO with motor control and machine learning.
- libmindsession/libmindpiece: These modules can connect to AIBO, transfer the robot state to PC and simulate the robot brain with the downloaded data on the laptop later.
- libmindeye: Image processing library.

The average OSS projects do not put emphasize on the software quality except manual testing or unit tests because they are time consuming practices. Since the non-Unix operating system (Aperios) in Sony ERS-7 crashes easily and the memory resources are limited (64MB), rigorous testing practices and software validation were introduced in the AiBO+ development process.

Due to the project supports the multiplatform compilation, all components can be built and run on the host computer excluding a glue layer for the robot. *Component tests* are written for every

4 http://www.dogsbodynet.com/skitter.html

5 http://kecsapblog.blogspot.fi/2014/09/sony-ers-7-charging-station-pole-model.html

software module and executed on the host computer. As mentioned in Section 6.1, the robot states of a runtime session can be recorded, transferred to a computer via wireless network and used for simulation. As more complex behaviors are implemented for AIBO, new test cases (*session tests*) are defined how the robot reacts in a certain situation. When a new session test is acted by the robot, the runtime is saved and the subsequent AI simulations with the recorded data must produce the same outcome on the host computer. The component and session tests verify that new changes do not break the existing behaviors.

Open source programs can check the source codes for correctness. Valgrind[6] is a programming tool for memory debugging, memory leak detection and profiling. In the AiBO+ development, the component and session tests are run with the valgrind's memchecker module to avoid any invalid read or write in memory accesses and memory leaks. Other memory tools were also integrated in the project successfully, including AddressSanitizer[7] tool for Clang compiler and callgrind, helgrind in valgrind. One of the best available free static code analyzer is the Clang Static Analyzer[8] to keep the sources error-free and adding a new tool is planned for checking the includes in the C++ sources in the future. A candidate is the open source include-what-you-use[9], a backend for the Clang compiler.

A continuous integration system[10] (Jenkins) was installed on a computer to validate the new modifications with multiple steps before merging into the main software stack. First, the new revision is compiled for all supported platforms to detect any compilation problem, the component and session tests are run with the previously mentioned memory checks to avoid any memory handling error, and finally, the source code metrics are calculated for trend diagrams in Jenkins.

After the initial AiBO+ software release was published in April 2015, only two software bugs were found by the community members. The software crashed when the WLAN switch was off on the Sony ERS-7 and the Mac OSX could corrupt the directory names when the AiBO+ software image was copied to a memory stick. Both problems were fixed in a few days and a new version with hotfixes was uploaded to the project site on SourceForge. These quick response times build the trust among the community members.

6.4 Crowdsourcing for Machine Learning

The machine learning samples are collected on the robot and they are transferred to an AiBO+ Client (Figure 5) application running on Windows, Linux or Android via the local wireless network. These samples are completed with the owner name and email address and they are uploaded to the AiBO+ Cloud what is stored inside a noSQL Enginio database in the Qt Cloud Services. Finally, the samples are downloaded by the author and they are removed from the cloud.

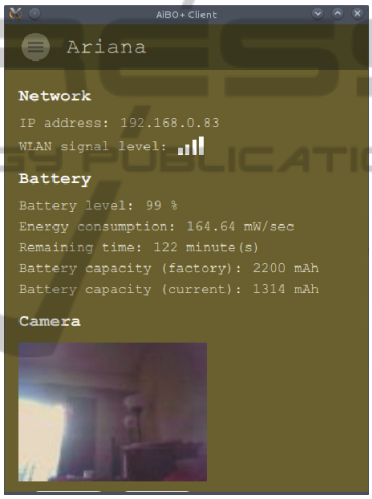

Figure 5: AiBO+ Client application under Ubuntu Linux.

The above mentioned process must be implemented with transparency and guaranteeing the privacy of the contributing users. Since all AiBO+ components are open source if any credential for the cloud services are included in the sources, anybody can get unauthorized access to the contributed data. The problem was solved as follows:

1. The sensitive data (name, email address) are encoded with RSA algorithm with a shared public key before uploading to the AiBO+ Cloud. The private key, which must be available for decoding, is kept private by the first author to avoid any 3rd party access.
2. The regular users are restricted to upload new data to the cloud, but the first author can delete the downloaded samples. The credentials for this privilege are secret.

A preliminary work was done on floor surface

6 http://valgrind.org
7 https://code.google.com/p/address-sanitizer
8 http://clang-analyzer.llvm.org/
9 https://code.google.com/p/include-what-you-use
10 http://aiboplusci.privatedns.org

detection before starting the PhD studies (Kertész, 2014) and these results were polished and included in the AiBO+ software. The recognition rate is around 90% and the samples are being collected to improve the model and verify the accuracy in real environments.

In the future, the crowdsourcing is planned for power savings in the robot functions, anomaly detection during locomotion and sound event recognition.

REFERENCES

Anderson, D. P., 2004. *BOINC: A System for Public-Resource Computing and Storage*. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing.

Bradski, G. R., Kaehler, A., 2008. *Learning OpenCV*. 1st Edition, O'Reilly Media.

Brooks, R. A., 1991. *Intelligence Without Representation*. Journal of Artificial Intelligence, Vol. 47, pp. 139-159.

Christley, S., Madey, G., 2007. *Analysis of Activity in the Open Source Software Development Community*. In Proceedings of the 40th Annual Hawaii International Conference on System Sciences.

Crowston, K., Wei, K., Howison, J., and Wiggins, A.: *Free/Libre open-source software development: What we know and what we do not know*, Journal of ACM Computing Surveys, Vol. 44(2), pp. 7, 2012.

Franke, N., Shah, S., 2003. *How communities support innovative activities: an exploration of assistance and sharing among end-users*. Journal of Research Policy, Vol. 32(1), pp. 157-179.

Friedman, B., Kahn, P. H., and Hagman, J., 2003. *Hardware companions?: what online AIBO discussion forums reveal about the human-robotic relationship*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 273-280.

Hart, D., Goertzel, B., 2008. OpenCog: *A Software Framework for Integrative Artificial General Intelligence*. Proceedings of the First AGI Conference, pp. 468-472.

Hibbard, B., 2008. *Open Source AI*. Frontiers in Artificial Intelligence and Applications, pp. 473-477.

Iizuka, H., Ikegami, T., 2004. *Adaptability and diversity in simulated turn-taking behavior*. Journal of Artificial Life, Vol. 10(4), pp. 361-378.

Kahn, P. H., Freier, N. G., Friedman, B., Severson, R. L., and Feldman, E. N., 2004. *Social and moral relationships with robotic others?*. 13th IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), pp. 545–550.

Kertész, C., 2010. *A synchronized system concept and a reference implementation for a robot dog*. 14th Finnish Artificial Intelligence Conference (STeP), pp. 46-53.

Kertész, C., 2012. *Dynamic behavior network*. IEEE 10th Jubilee International Symposium on Applied Machine Intelligence and Informatics (SAMI), pp. 207-212.

Kertész, C., 2013. *Improvements in the native development environment for Sony AIBO*. International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI), Special Issue on Improvements in Information Systems and Technologies, Vol. 2, No. 3, pp. 50-54.

Kertész, C., 2014. *Exploring Surface Detection for a Quadruped Robot in Households*. 14th IEEE International Conference on Autonomous Robot Systems and Competitions (IEEE-ICARSC 2014), pp. 152-157.

King, D. E., 2009. *Dlib-ml: A Machine Learning Toolkit, Journal of Machine Learning Research.* Vol. 10, pp. 1755-1758.

Krafft, F. M., 2005. *Workflow in distributed volunteer projects — Intuitive approaches to modern Debian package development*. PhD Thesis, University of Limerick, Ireland.

Nao, 2011. *NAO Open Source – ALDEBARAN Robotics shares source code with its community*. Aldebaran Robotics, Press Release.

Ponsen, M. J. V., Lee-Urban, S., Muñoz-Avila, H., Aha, D. W., and Molineaux, M., 2005. *Stratagus: An Open-Source Game Engine for Research in Real-Time Strategy Games*. IJCAI Workshop on Reasoning, Representation and Learning in Computer Games.

Porter, A., Yilmaz, C., Memon, A. M., Krishna, A. S., Schmidt, D. C., and Gokhale, A., 2006. *Techniques and processes for improving the quality and performance of open□source software*. Journal of Software Process: Improvement and Practice, Vol. 11(2), pp. 163-176.

Schaeling, B., 2014. *The Boost C++ Libraries*. Second Edition, XML Press.

Simpson, R., Page, K. R., and Roure, D. D., 2014. *Zooniverse: Observing the World's Largest Citizen Science Platform*. Proceedings of the 23rd International Conference on World Wide Web, pp. 1049-1054.

Von Hippel, E., Katz, R., 2002. *Shifting Innovation to Users via Toolkits.* Journal of Management Science , Vol. 48(7), pp. 1-13.

Yan, J., Wang, X., 2013. *From Open Source to Commercial Software Development - the Community Based Software Development Model.* Proceedings of 34th International Conference on Information Systems, pp. 1-20.