

Integration of humanoid robots in collaborative working environment: a case study on motion generation

Olivier Stasse, Rudolf Ruland, Florent Lamiraux, Abderrahmane Kheddar, Kazuhito Yokoi, Wolfgang Prinz

▶ **To cite this version:**

Olivier Stasse, Rudolf Ruland, Florent Lamiraux, Abderrahmane Kheddar, Kazuhito Yokoi, et al.. Integration of humanoid robots in collaborative working environment: a case study on motion generation. Intelligent Service Robotics, 2009, 2 (3), pp.153-160. 10.1007/s11370-009-0045-8 . hal-00451346

HAL Id: hal-00451346
https://hal.science/hal-00451346

Submitted on 28 Jan 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Intelligent Service Robotics manuscript No.
(will be inserted by the editor)

Integration of Humanoid Robots in Collaborative Working Environment: A case study on motion generation

O. Stasse, R.Ruland, F. Lamiraux, A. Kheddar, K. Yokoi · W. Prinz

Received: date / Accepted: date

Abstract This paper illustrates through a practical example an integration of a humanoid robotic architecture, with an open-platform collaborative working environment called BSCW (Be Smart - Cooperate Worldwide). BSCW is primarily designed to advocate a futuristic shared workspace system for humans. We exemplify how a complex robotic system (such as a humanoid robot) can be integrated as a proactive collaborative agent which provides services and interacts with other agents sharing the same collaborative environment workspace. Indeed, the robot is seen as a 'user' of the BSCW which is able to handle simple tasks and reports on their achievement status. We emphasis on the importance of using standard software such as CORBA (Common Object Request Broker Architecture) in order to easily build interfaces between several interacting complex software layers, namely from real-time constraints up to basic Internet data exchange.

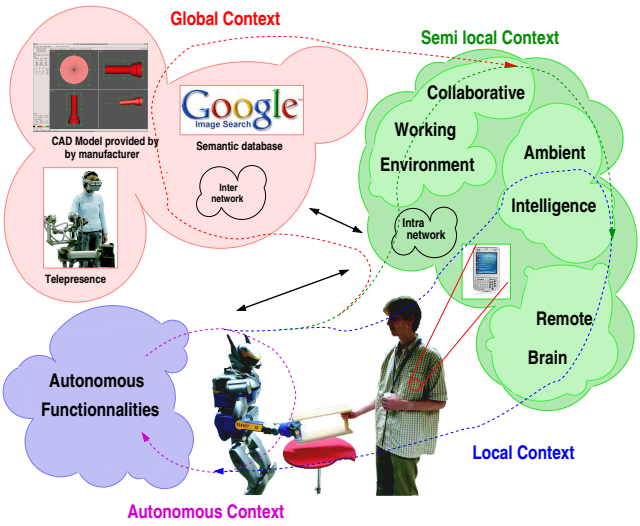


Fig. 1 Four contexts of task realization in the physical common workspace

1 Introduction

Humanoid robots are currently considered in several applications ranging from the house maid robot able to clean [1] or even cook [2], to multi-purpose robotic system flexible to fast changing in tasks and product lines in industries. They would be able to manipulate various products, inspect and guard small and middle size companies outside the factories, etc. In these scenarios it is important to integrate, and even to take advantages of the existing IT- infrastructure, to realize robot programming and mission assignments. The purpose of this work is to investigate the introduction of robots –especially humanoid robots– as agents in collaborative working environments. By collaborative environment, we mean futuristic human-centric shared workspace system for advanced collaboration between humans. This work is based on papers [3] [4] describing the current state of the art and limitations related to the integration of humanoid robotic architecture in such environments.

Address(es) of author(s) should be given

2 Context

2.1 Taxonomy of collaborative contexts

This paper is written from the point of view of a software architect designer. We have defined four contexts

of task realization in a physical workspace as depicted in Fig. 1:

1. An *autonomous* context realization when the robot is directly interacting with a human to perform a task, and particularly during physical interaction.
2. A *local* context realization when the robot is using the surrounding network and computer capabilities to expand its functional space. This is typically the case in the presence of ambient intelligence and/or in the context of the remote brain approach [5].
3. A *semi-local* context realization when the robot is interacting with a collaborative working application targeted for an application or for a structure such as a company. It is semi-local because its semantic scope is local, but can be geographically spread over several locations.
4. A *global* context realization when the robot is interacting with services external to its *semi-local* structure for instance Google Images services, manufacturer product specification, etc.

This paper deals more particularly with the semi-local context. Recent work by Peer et al. [6] demonstrated how two people, one in Japan collocated with the robot, and the other one in Germany, could perform a collaborative task using a telepresence system and a humanoid robot. The person in Germany used a telepresence system to teleoperate a humanoid robot HRP-2 to lift an object with the operator in Japan sharing the same physical space and object with the robot. Although the realization of this experiment requires the use of complex control architecture in order to guarantee stability of the humanoid robot, and of the overall system, the role of the robot was however limited to reproduce the actions of the master operator in Germany.

In a different context, Sagakuchi *et al.* [7] demonstrated how HRP-2 could be used in an intelligent house to perform autonomous actions such as closing the door of a fridge. However if one aims at having humanoid robots used in working offices or flexible SMEs to perform various tasks and adapt quickly to fast changing production lines, the most efficient way to assign robotic tasks to a robot is to interface the robot architecture to the local IT existing facilities if available. We demonstrate in this paper how HRP-2, relying on advanced architecture and planning software, can be smoothly integrated into a real collaborative working environment. We also report on the use of software technology standards to easily build appropriate interfaces.

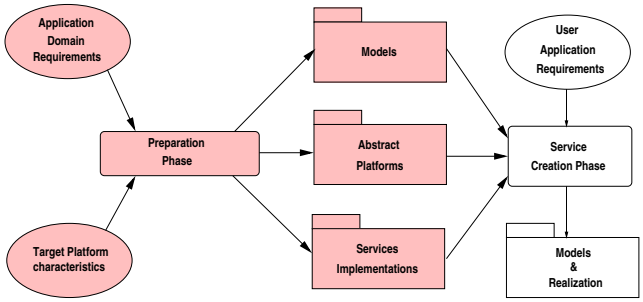


Fig. 2 Development flow to embed a humanoid robot in an application providing a Collaborative Working Environment

2.2 Software development approach

The goal of this work is to create the tools necessary to introduce humanoid robots in Collaborative Working Environments. To this aim, we follow the design principles depicted in Fig. 2 and proposed by Almeida [8]. From the field of Collaborative Environment and humanoid robotics, requirements and characteristics can be expressed to specify models and abstract platforms as depicted in the left side of Fig. 2. For instance, one of the current characteristics of humanoid robots is that they evolve on flat floors because of the stability criterion called ZMP (Zero Momentum Point). A more detailed description of the characteristics regarding human humanoid collaborative work can be found in the two papers cited previously [3][9]. They provide the descriptions of models and abstract platforms on top of which we have implemented services such as motion generation and motion planning.

A more details description of those two services can be found in Paragraph 4.1.2 and Section 4.2. This paper illustrates this approach.

3 The software architecture

The challenging part of this demonstration is the separation between software specific to robotic technology and the overall collaborative technology. This is achieved by raising the functional level of the robot to a level of autonomy sufficient to interact in a human centered environment. The atomic level of understanding on which the robot and the human operator agree is the *task* as commonly understood in the context of collaborative working environment and not in the control sense as introduced in section 4.1.2. Those tasks are defined in the context of BSCW (Be Smart - Cooperate Worldwide), a collaborative working environment used for several European IP projects. The semantic and the ontology of the task is not specified in this context. It has merely a name, and fields. The interpretation of

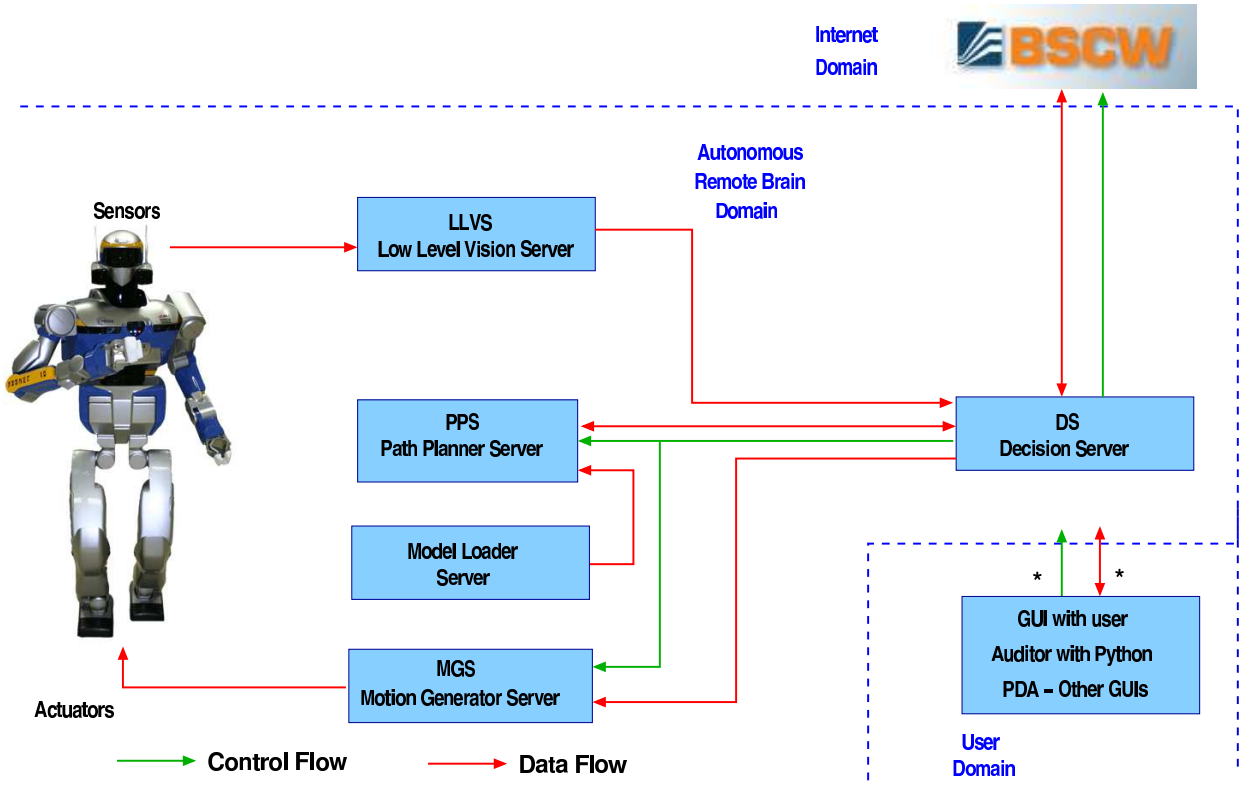


Fig. 3 Functional block implemented as CORBA and OpenRTM servers

those properties is left to the users to which those tasks are aimed for. Following this guideline, we give a brief overview of how the robot is able to interpret tasks and to reply with the appropriate answer. To give a flavor of what the robot is able to do, we give an integrated overview of the experiments we have been able to achieve so far in the context of the project. Later on, a case study is described on a surveillance task for HRP-2 in a known environment together with experiments.

3.1 Architecture of HRP-2

The architecture depicted in figure 3 is a functional block oriented architecture. Each block is implemented by a CORBA server. The Low Level Vision server aims at providing early vision processing such as segmentation, optical flow, edge detection and real-time Self Localization and map-building. The world model server builds a representation of the world by an accumulation of disparity maps associated with a location. The object visual model server is in charge of building and looking for an object. The path planner server provides the steps to perform to go from one point of the environment to another. The visual attention server finds the next best view in order to search for an object in an unknown environment. The motion generator generates and realizes a dynamically stable motion when the robot is required to perform steps, or to perform some tasks with its end-effectors. The decision layer is based upon the classical Hierarchical Finite State Machine paradigm. More precisely we are using the statechart specification of UML. The current extension of the standard template library called boost implements such a specification. We have used it to realize the Decisional block. This part can be easily specified by a user using today's UML state-chart modeler. In this paper, we will mostly describe the step planner server and the decision layer.

3.2 BSCW

BSCW is a cooperation platform on the Internet which allows to share documents, organize team work by assigning tasks, organize meetings, create communities, allow direct communication or information distribution such as e-mail or RSS feeds. This creation of the Fraunhofer Institute for Applied Information has been existing since the mid-1990s, and is currently supported by a spin-off company called OrbiTeam. BSCW is being extended in the frame of the European Integrated Project called Ecospace [10] to develop a collaborative

environment for eProfessionals. BSCW can be freely downloaded for academic research purposes. BSCW in this work provides the model and the service implementations specific to collaborative working environments.

4 Motion generation, Planning and high level description

One of the main challenges with complex redundant robots such as humanoid robots is to generate motions with commands simple enough to be manageable by a human while maintaining the overall stability of the robot. Motion planners are limited by the combinatorial explosion when trying to find a trajectory in configuration spaces of high dimension. To overcome this difficulty, the motion planner reasons on a simplified model of the robot.

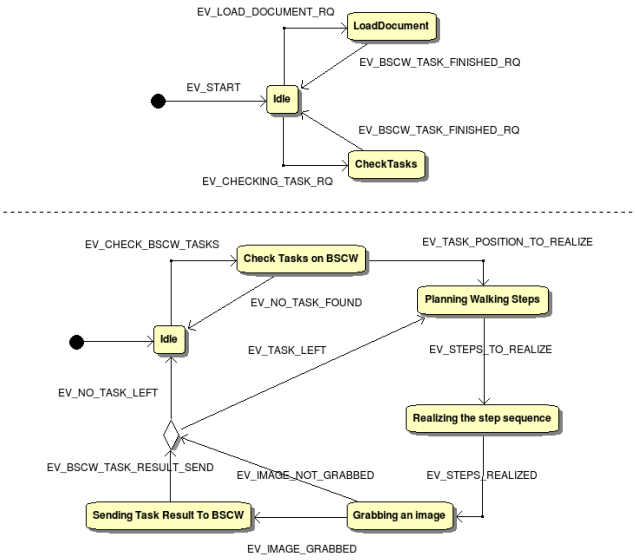


Fig. 4 Statechart model of the case study

4.1 Motion generator

4.1.1 Stability

The stability criterion used in our work is the Zero Momentum Point (ZMP) which assume that both feet are on a flat floor. This criterion, together with a linearization of the dynamic model of the robot make real-time trajectory generation possible. The algorithms implementation used to generate those reference have been organized in a framework allowing prototyping and multiple modalities.

4.1.2 Generalized Inverse Kinematics

Introduced initially by Nakamura et al.[11], generalized inverse kinematics (GIK) offers a prioritization scheme to associate several controllers together in order to generate motion for a redundant robot. Finding the activation and the prioritization of those controllers is still an open issue.

4.2 Motion planning

Following the previous remark, current fast motion planning relies on simplified model which are known to be realized by the control architecture presented in Section 4.1.2. A popular solution is to discretize the set of feasible foot-steps of the robot, and to perform an A^* [12][13] search in the environment. One problem is the possibility of stepping in some situations [13]. In this work we propose a different approach. In a first step, a collision-free path is planned for the bounding box of the robot moving in the plane. In a second step, footsteps are sampled along the path of the bounding box. This sequence of footsteps is then given as input to the same walking pattern generator which generates the motion of HRP-2 in the control architecture.

4.3 Decision Layer

As it is done classically we used a Hierarchical Finite State Machine to map a discrete semantic with a set of controllers and parameters. This mapping is done here in an arbitrary manner. Recent works [14][15][16] is trying to create automatically this mapping by grouping set of trajectories of human activities. Making the interfaces provided by the blocks depicted in 3 accessible to a collaborative environment is an important issue. In our case, this is filtered out by the decision layer.

Exposing the the interface description of the components through Web Service Description Language would clearly be inefficient. The level of autonomy reached through these components makes robot task specification much more simpler to the user.

5 Simulations and experiments

We are presenting our current status in trying to integrate HRP-2 in a full-size CWE.

5.1 Setup description

In order to achieve our integration of HRP-2 in a CWE, the hierarchical finite state machine depicted in figure

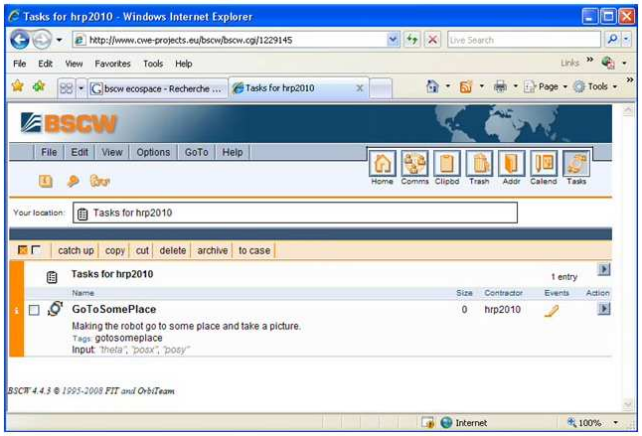


Fig. 5 List of tasks assigned to HRP-2 by others BSCW users.

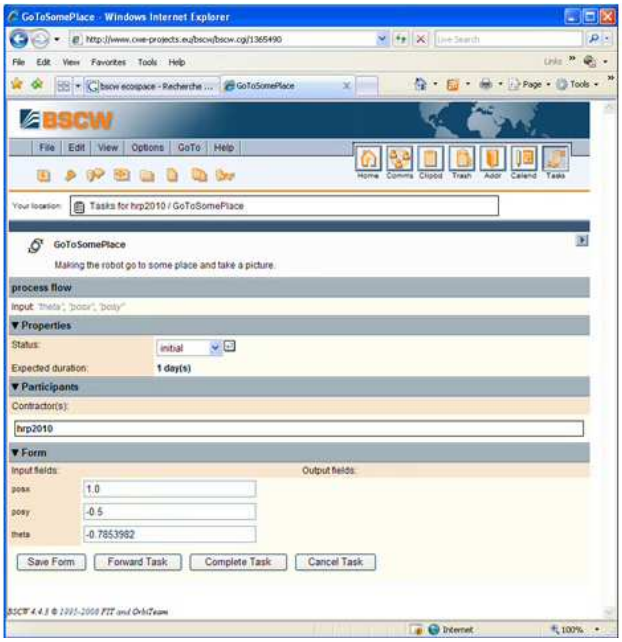


Fig. 6 The task *GoToSomePlace* specified in BSCW.

4 has been implemented to provide a simple decision layer. At first the robot system connects and register to BSCW. From its list of tasks, depicted in Fig. 5, it checks if there is a task named *GoToSomePlace*. Such task, depicted in Fig. 6, is assigned by another user of the system. It extracts from this task the fields specifying the target position and orientation of the robot. From this target position the motion planner tries to plan a trajectory assuming that the environment is fully known and static. If such a trajectory exists the steps found are send to the control architecture to realize the motion. Once the steps are realized the robot takes a picture and upload it back to BSCW. Fig. 7 shows a dynamical simulation of the steps generated by the motion planner with a rather simple situation. Fig. 9

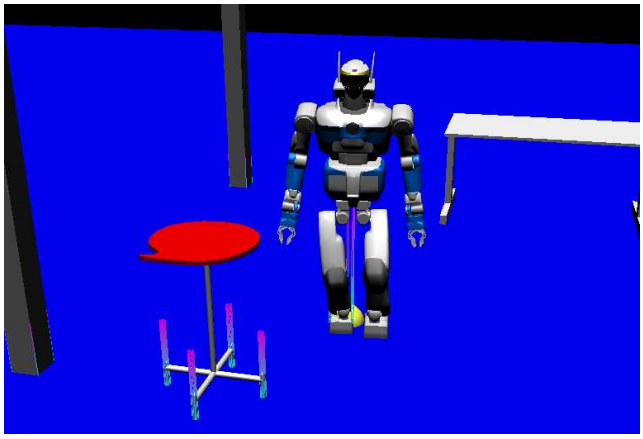


Fig. 7 Dynamical simulation of the steps generated by the motion planner

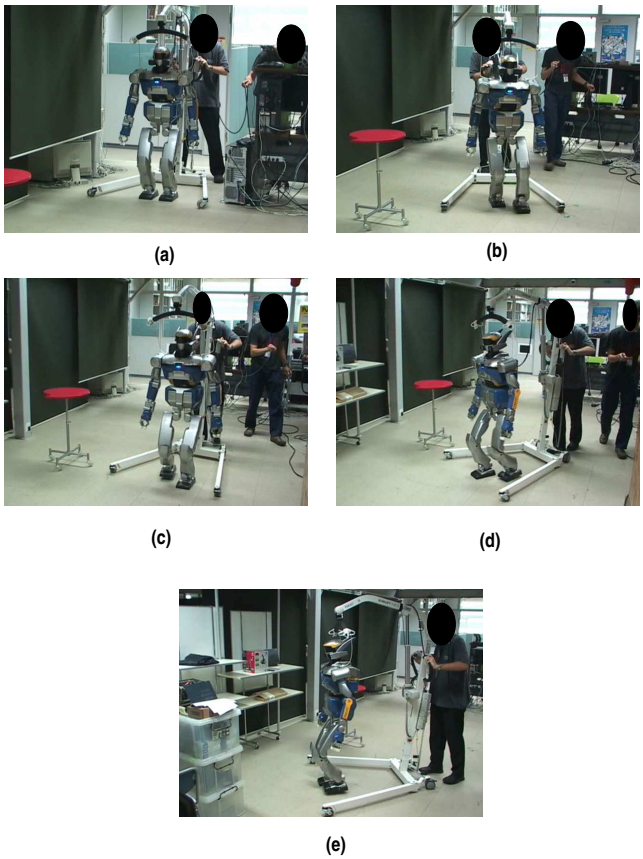


Fig. 8 Real life experiments with HRP-2

displays a more complex situation handled by the motion planner. The simple situation has been executed on the real platform, and some snapshots of the experiment are depicted in figure 8.

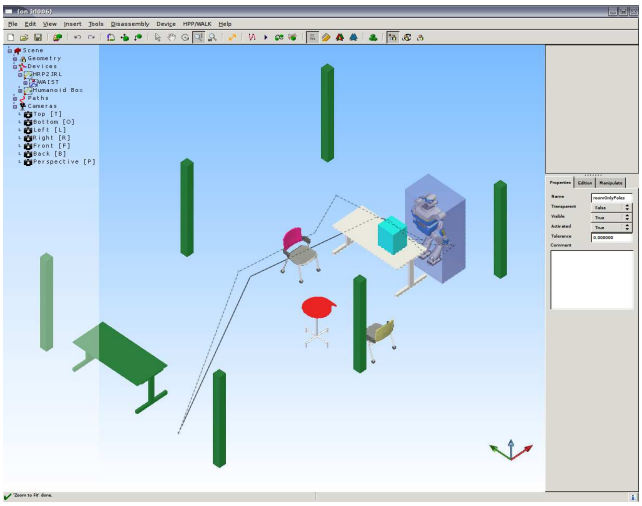


Fig. 9 The motion planner solving a more complex situation

5.2 Using software standards

To maximize the compatibility and the reuse of the software components, we have tried to use as much possible standards, software tools and design patterns. The control system and the physical simulation are realized using OpenHRP [17] which is currently supported by the Japanese government to become a national platform. Because HRP-2 [18] embeds advanced CPU systems we are using mostly CORBA to handle the middleware issues. As CORBA does not integrate any way to specify data flow, scheduling properties, control and interface parts of a component, a new OMG standard has been proposed called Robot Technology Middleware (RTM)[19] to fill the void. HPP (Humanoid Path Planner), our planning framework, has been used together with these technologies in this paper. CORBA and RTM made possible to use 4 machines with several cores to make the computation in a seaming-less manner. With data flow structure, RTM [20] allows to avoid a dependency on interfaces and a graph can be constructed by an external client. The decision layer follows the UML statechart rationale and is implemented using the boost::statechart library [21]. We hope to move forward with an automatic code generation from model description.

The connection with BSCW is realized with XML-RPC, which allows to use libraries already available to access the remote application. The open definition of a task in BSCW allows the robot in deciding autonomously if the task is understood and feasible.

6 Conclusion

We have presented our current work in trying to include a humanoid robot in a real collaborative environment using standard software and robotic technologies. With a sufficient level of functionalities, the robot is able to act as an autonomous user interpreting simple command and sending a feedback on this collaborative environment. They are open issues with the mapping of the capabilities of such a robot in a company collaborative tools. We believe that raising the range of functionalities of such robot while using software standards is the good direction to tackle this issue.

References

1. K. Okada, M. Kojima, Y. Sagawa, T. Ichino, K. Sato, and M. Inaba, "Vision based behavior verification system of humanoid robot for daily environment tasks," in *IEEE/RAS International Conference on Humanoid Robots*, 2006, pp. 7–12.
2. F. Gravot, A. Haneda, K. Okada, and M. Inaba, "Cooking for humanoid robot, a task that needs symbolic and geometric reasonings," in *IEEE/RAS Int. Conf. on Robotics and Automation*, 2006, pp. 462–467.
3. A. Bauer, D. Wollherr, and M. Buss, "Human-robot collaboration: A survey," *International Journal of Humanoid Robotics*, vol. 4, pp. 47–66, 2008.
4. O. Stasse, E. S. Neo, F. Lamiraux, A. Kheddar, and K. Yokoi, "Architectures and models for humanoid robots in collaborative working environments," in *International Symposium on Robotics*, 2008, pp. 354–359.
5. M. Inaba, T. Ninomiya, Y. Hoshino, K. Nagasaka, S. Kagami, and H. Inoue, "A remote-brained full-body humanoid with multisensor imaging systemof binocular viewer, ears, wrist force and tactile sensor suit," in *IEEE/ICRA*, vol. 3, 1997, pp. 2497–2502.
6. A. Peer, S. Hirche, C. Weber, I. Krause, M. Buss, S. Miossec, P. Evrard, O. Stasse, E. S. Neo, A. Kheddar, and K. Yokoi, "Intercontinental multimodal tele-cooperation using a humanoid robot," in *IEEE/RSJ International Conference on Intelligent RObots and Systems*, 2008, pp. 405–411.
7. T. Sakaguchi, T. Ujiie, S. Tsunoo, K. Oohara, E. Neo, and K. Yokoi, "Intelligent-ambience robot cooperation - closing door with humanoid robot-," in *4th International Conference on Ubiquituous Robots and Ambient Intelligence (URAI)*, 2007, pp. 383–388.
8. *Distributed Applications and Interoperable Systems*. Springer-Verlag, 2006, ch. Model-Driven Development of Context-Aware Services, pp. 213–227.
9. O. Stasse, N. E. Sian, F. Lamiraux, T. Sakaguchi, A. Kheddar, and K. Yoikoi, "Architectures and models for humanoid robots in collaborative working environments," in *International Symposium on Robotics*, 2008, pp. 354–358.
10. W. Prinz, H. Loh, M. Pallot, H. Schaffers, A. Skarmeta, and S. Decker, "Ecospace - towards an integrated collaboration space for eprofessionals," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006.*, 2006, pp. 1–7.
11. Y. Nakamura and H. Hanafusa, "Inverse kinematics solutions with singularity robustness for robot manipulator control," *Trans. ASME Journal of Dynamic System, Measures and Control*, vol. 108, pp. 163–171, 1986.

12. J. Chesnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the honda asimo humanoid," in *IEEE/ICRA*, 2005, pp. 631–636.
13. J.-M. Bourgeot, N. Cislo, and B. Espiau, "Path-planning and tracking in a 3D complex environment for an anthropomorphic biped robot," in *IEEE/RSJ IROS*, 2002, pp. 2509–2514.
14. Y. N. W. Takano, K. Yamane, "Capture database through symbolization, recognition and generation of motion patterns," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 3092–3097.
15. O. C. Jenkins and M. J. Matarić, "Performance-derived behavior vocabularies: Data-driven acqusition of skills from motion," *International Journal of Humanoid Robotics*, vol. 1, no. 2, pp. 237–288, Jun 2004.
16. E. Drumwright, V. Ng-Thow-Hing, and M. Mataric, "Toward a vocabulary of primitive task programs for humanoid robots," in *International Conference on Development and Learning (ICDL)*.
17. F. Kanehiro, H. Hirukawa, and S. Kajita, "Openhrp: Open architecture humanoid robotics platform," *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.
18. K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, "Humanoid robot hrp-2," in *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, vol. 2, 2004, pp. 1083–1090.
19. N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.-K. Yoon, "Composite component framework for rt-middleware (robot technology middleware)," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2005, pp. 1330–1335.
20. N. Ando, T. Suehiro, K. Kitagaki, and T. Kotoku, "Rt(robot technology)-component and its standardization- towards component based networked robot systems development - ," in *SICE-ICASE International Joint Conference 2006 (SICE-ICCAS 2006)*, 2006, pp. 2633–2638.
21. A. H. Dnni, "The boost statechart library." [Online]. Available: http://www.boost.org/doc/libs/1_34_0/libs/statechart/doc/index.html