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

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

To cite this version:

Olivier Stasse, Rudolf Ruland, Florent Lamiraux, Abderrahmane Kheddar, Yokoi Kazuhito, et al.. Integration of Humanoid Robots in Collaborative Working Environment: A case study on motion generation Anonymous. Intelligent Service Robotics, Springer Verlag, 2009, pp.153-160. hal-01117974

HAL Id: hal-01117974
https://hal.archives-ouvertes.fr/hal-01117974

Submitted on 18 Feb 2015

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

Anonymous

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. BSCW is primarily designed to advocate a futuristic shared workspace system for humans. We exemplify how such complex robotic systems (such as humanoids) can be integrated as a proactive collaborative agent who provides services and interact 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 in order to easily build interfaces between several interacting complex software layers, namely from real-time constraints up to basic Internet data exchange.

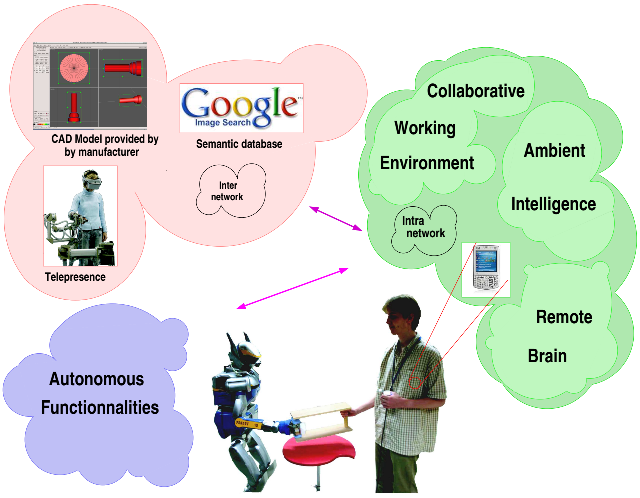


Fig. 1 Four contexts of task realization in the physical common workspace

1 Introduction

Humanoid robots are currently targeted in several applications ranging from the house maid robot able to clean [1] or even cook [2], to industry fields as a multipurpose robotic system which is flexible to fast changing in tasks and product lines, 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 each robot programming and mission assignments. The context of this work is to investigate the introduction of robots – especially humanoids– as agents in collaborative working environment which purpose is to design a futuristic human-centric shared workspace system for advanced collaboration between humans. This work is based on review papers [3] [4] describing the current state of the art and limitations related to the integration of humanoid robotic architecture in such environments.

2 Context

2.1 Taxonomy of collaborative contexts

The view taken in this paper is from the side of a software architect designer. In this context we have defined four contexts of task realization in a physical workspace as depicted in Fig. 1:

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

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 treats more particularly the semi-local context. Recent work by Peer and al. [6] demonstrated how two humans, one in Japan collocated with the robot, and the other one being in Germany, could perform a collaborative tasks using a telepresence system and a humanoid robot. The person in Germany used a telepresence system to teleoperate a humanoid 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 that 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. 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 products lines, the most efficient way to assign robotic tasks missions is to interface the robotic architecture to the specific working one 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.

2.2 Software development approach

The goal of this work is to create the tools necessary to introduce humanoid robots in Collaborative Working Environments. Therefore, from the field of Collaborative Environment and humanoid robotics, requirements and characteristics can be formulated to specify models and abstract platforms. For instance, one of the current characteristic of humanoid robot is to evolve on flat floor because the stability criteria which can be currently computed in real-time is ZMP. A more detailed description of the characteristics regarding human humanoid collaborative work can be found in the two review papers cited previously [3][8]. Based on those characteristics and models we have implemented services such as motion generation and motion planning which can be used by the user to create its application using its own application-models . A more details description of those two services can be found in paragraph 4.2.2 and section 4.3.

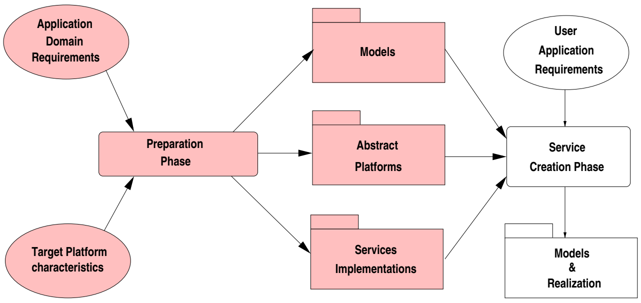


Fig. 2 Development flow to embed a humanoid robot in an application providing a Collaborative Working Environment

3 The software architecture

The challenging part of this demonstration is to maintain separate the software specific to robotic technology from the overall collaborative technology. This is achieved by raising the functional level of the robot to an autonomy sufficient to interact in a human centered environment. The atomic level of understanding on which the robot and the human agree on is the *task* as commonly understood in the context of collaborative working environment and not in the control sense as introduced in section 4.2.2. Those tasks are defined in the context of BSCW, 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 those properties is left to the users to which those tasks are aimed for. Following this line, we give a brief overview of how the robot is able to interpret the task and 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

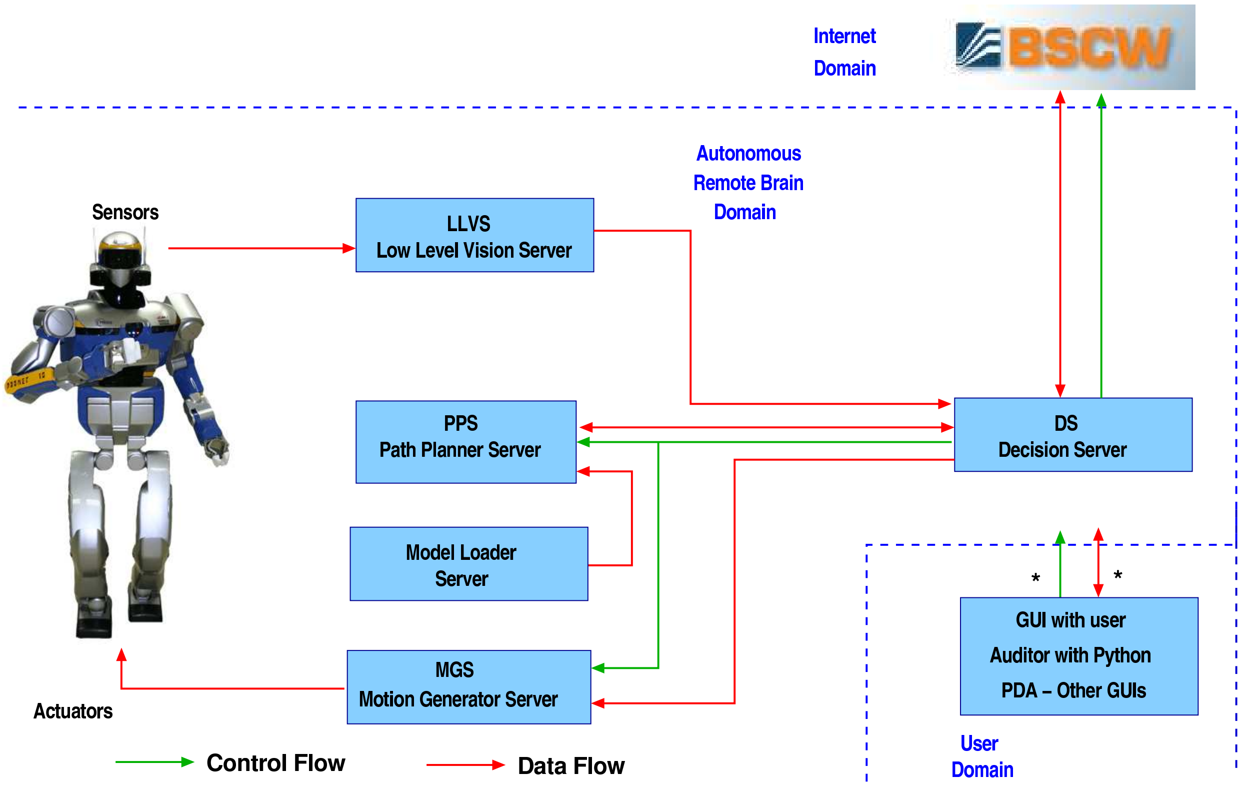


Fig. 3 Functional block implemented as CORBA and OpenRTM servers

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 HRP-2's architecture

The architecture depicted in figure 3 is a functional block oriented architecture. where 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 SLAM. The world model server build 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 needs to perform steps, or 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 state-chart specification of UML. The current extension of the standard template library called boost implement such 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's 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 exists 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 [9] 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 services implementations specific to collaborative working environments.

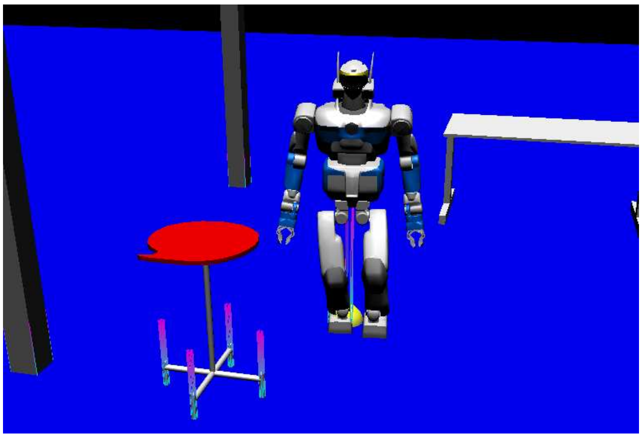


Fig. 4 Dynamical simulation of the steps generated by HPP

4 Motion generation, Planning and high level description

4.1 Introduction

One of the main difficulty with complex redundant robot such as humanoid robot is to find a way to generate motion with commands simple enough to be manageable by a human while maintaining the overall stability of the robot. Neo [10] proposed recently such a system for on-line motion generator in the context of teleoperation. Mansard [11] demonstrated how it is possible to generate motion autonomously using visual information. In the first case, the field of application is mostly cases where human assistance are needed to help people with limited mobility, in disaster situation or in the case of space application. In the second case, the approach is mostly reactive and need to be coupled with higher decisional layer such as motion planner. Motion planner however are limited by the combinatorial explosion when trying to find a trajectory in the configuration space. A key to find trajectories which are dynamically stable is to find simple models for the motion planner which correspond to the control architecture.

4.2 Motion generator

4.2.1 Stability

The stability criteria used in our work is the Zero Momentum Point which assume that both feet are on a flat floor. This criteria is important because it reduces the set of trajectories possible by the robot. Indeed when considering other stability criteria, the range of possible motions might include contact with obstacles and other complex interaction. Moreover to make the problem tractable in the high speed control loop necessary for such robot, supplementary constraints are considered which simplify the numerical resolution, but also constraint the set of trajectories. The current scheme used in general for humanoid robot acts more as dynamically stable reference generator and uses a simpler controller to realize the reference. The algorithms implementation used to generate those reference have been organized in a framework allowing prototyping and multiple modalities.

4.2.2 Generalized Inverse Kinematics

Introduced initially by Nakamura, the generalized inverted kinematics offers a prioritization scheme to associate several controllers together in order to generate motion for a redundant robot. Its equivalent in the force domain is the operational space control. It is receiving a renewed interest for whole-body motion generation in the field of humanoid robotics. Due to current practical limitations, most of the walking humanoid robots are not using a low-level torque control but rather a position-based control. Finding the activation and the prioritization of those controllers is still an open issue. Some work exist to GIK in planning to correct trajectories when considering dynamical stability, For sake of simplicity in the remaining of this paper, it is assume that the underlying GIK provide one solution for one stable reference trajectory provided by the previous module.

4.3 Planning

Following the previous remark, current fast planning relies on simplified model which are known to be realized by the control architecture presented in section 4.2.2. A popular solution is to discretize the set of feasible footsteps of the robot, and perform an A^* [12][13] search in the environment. One problem is the possibility of stepping in some situations. In this work we propose a different approach where the robot is seen as a box with a behavior similar to a mobile robot. This model is used to connect configurations chosen by a probabilistic roadmap configuration shooter. The software used in this paper is is based upon KineoWorks a product commercialized by Kineo, a spin-off company based on the work of the second co-author. This software provide the overall architecture to apply probabilistic roadmap with various robot models. In this specific application, the robot has three degree of freedom the position on the ground and the orientation. Once the configurations are connected with the chosen smooth function the steps are placed along the trajectory. To generate

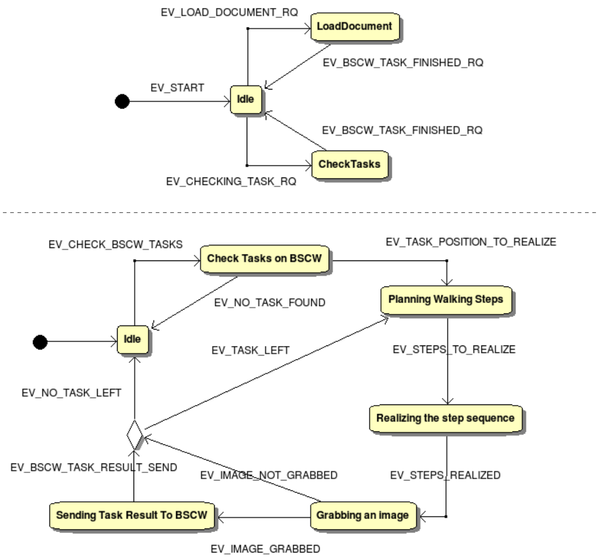


Fig. 5 Statechart model of the case study

the complete robot configuration, the framework uses the same walking pattern generator used to generate the motion of HRP-2 in the control architecture. This allows to check the collisions. When no trajectory is found, and when the environment limits are known, the system is able to return a failure message.

4.4 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 usually done in an arbitrary manner. Recent works [14][15][16] is trying to create automatically this mapping by grouping set of trajectories of human activities. There is an important issue here in making accessible the interfaces provided by the block depicted in 3 with a collaborative environment. In our case, this is filtered out by the decision layer.

One way could be to use the interface description of the component and expose them through Web Service Description Language.

Fortunately the link described previously between the planning and the control layer allow the humanoid robot's high level system decide by itself if a motion asked to the robot is feasible or not. Such capability facilitate the user programming of the robot behavior, and free the Collaborative Working Environment to have any knowledge on the robot.

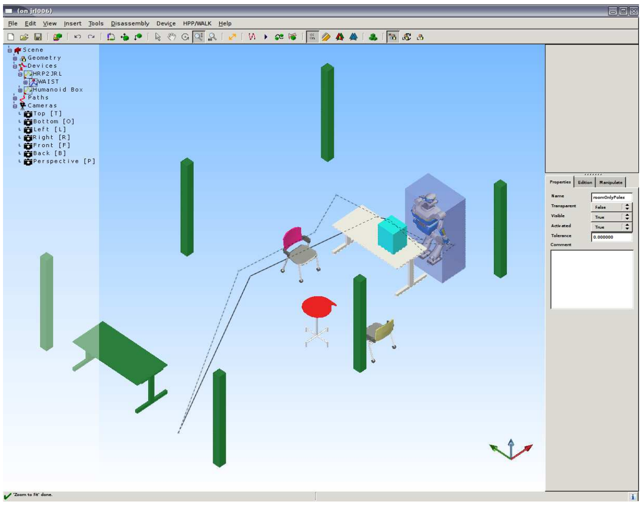
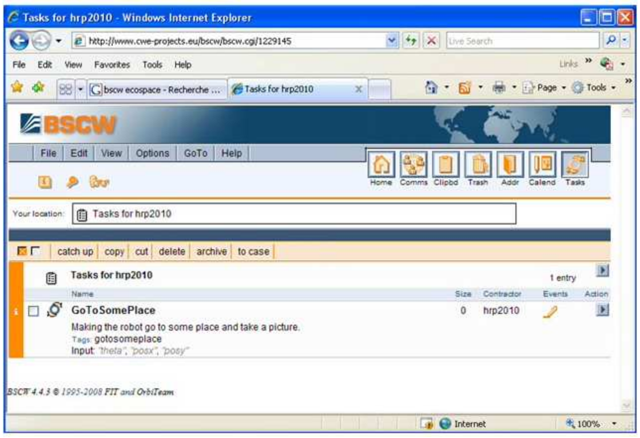


Fig. 6 HPP solving a more complex situation



Tasks for hrp2010 - Windows Internet Explorer

http://www.cwe-projects.eu/bscw/bscw.cgi/1229145

bscw ecospace - Recherche ...

Tasks for hrp2010

Your location: Tasks for hrp2010

Tasks for hrp2010 1 entry

Name	Size	Contractor	Events	Action
GoToSomePlace Making the robot go to some place and take a picture. Tags: gotosomeplace Input: "theta", "posx", "posy"	0	hrp2010		

BSCW 4.4.3 © 1995-2008 FIT and OrbiTeam

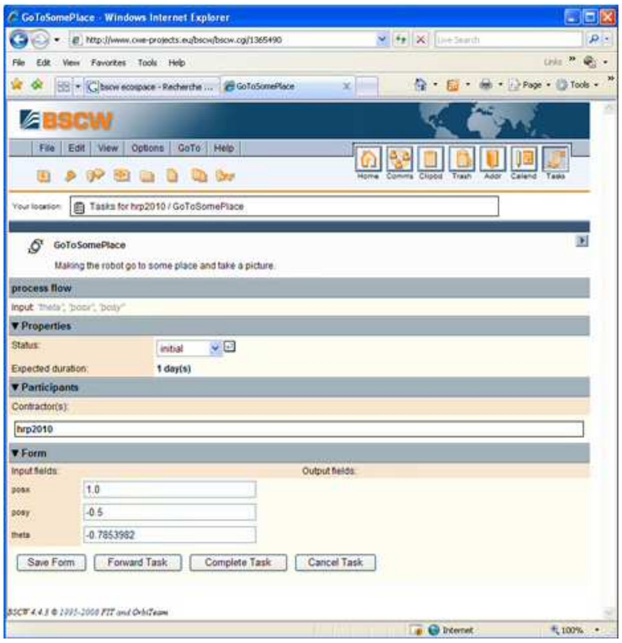
Fig. 7 List of tasks assigned to HRP-2 by others BSCW users.

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 5 has been implemented to provide a simple decision layer. At first the robot system connect to BSCW and identify itself. It checks in its list of tasks, Fig. 7 if there is a task named *GoToSomePlace*, Fig. 8 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 and assuming that the environment is fully known and static, HPP tries to plan a trajectory. If such a trajectory exists



Your location: Tasks for hrp2010 / GoToSomePlace

GoToSomePlace

Making the robot go to some place and take a picture.

process flow

Input: "theta", "posx", "posy"

▼ Properties

Status: initial

Expected duration: 1 day(s)

▼ Participants

Contractor(s):

hrp2010

▼ Form

Input fields: Output fields:

posx 1.0

posy -0.5

theta -0.7853982

BSCW 4.4.3 © 1995-2008 FIT and OrbiTeam

Fig. 8 The task *GoToSomePlace* specified in BSCW.

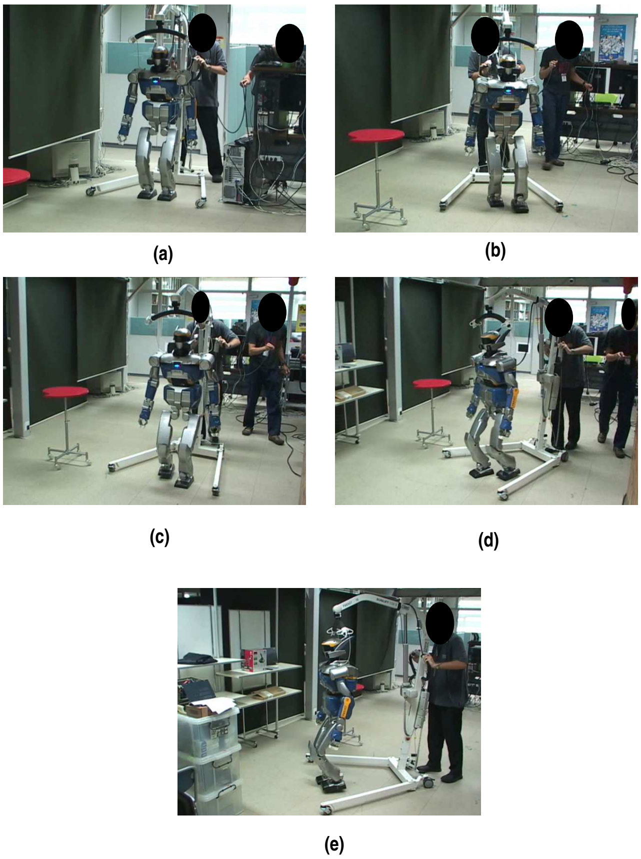

(a)
(b)
(c)
(d)
(e)

Fig. 9 Real life experiments with HRP-2

the steps found are send to the control architecture to realize the motion. Once the steps are realized the robot take a picture and upload it back to BSCW. Figure 4 shows a dynamical simulation of the steps generated by HPP with a rather simple situation. Figure 6 displays a more complex situation handled by HPP. The simple situation has been executed on the real platform, and some snapshots of the experiment are depicted in figure 9.

5.2 Software consideration

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 instead to concentrate on new concepts. 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. Because 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 to fill the void. HPP, our planning framework, has been used together with this technology 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[19] allows to avoid a dependency on interfaces and a graph can be constructed by an external client. When computational time is constant, it is possible with appropriately specified scheduling properties to perform model checking. The decision layer follows the UML statechart rationale and is implemented using the boost::statechart library [20]. 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 to decide 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, p. accepted.
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, 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. 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.
9. 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.
10. E. S. Neo, K. Yokoi, S. Kajita, and K. Tanie, “Whole-body motion generation integrating operator’s intention and robot’s autonomy in controlling humanoid robots,” *IEEE Transactions on Robotics*, vol. 23, pp. 763–775, 2007.
11. N. Mansard, O. Stasse, F. Chaumette, and K. Yokoi, “Visually-guided grasping while walking on a humanoid robot,” in *IEEE ICRA*, 2007, pp. 3041–3047.
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, 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.
20. A. H. Dnni, “The boost statechart library.” [Online]. Available: http://www.boost.org/doc/libs/1_34_0/libs/statechart/doc/index.html