Chapter 2

Supervision for Human-Robot Interaction

Contents

2.1 Role of the supervisor in the global architecture

One of the goals of our research group at LAAS-CNRS is to build a fully autonomous robot which interacts and performs Joint Actions with humans. To do so, an architecture for human-robot interaction has been developed and is constantly improved by the group. This architecture is composed of several modules and a simplified scheme of it can be found in Fig. 2.1.

Sensorimotor layer: The lower level of the architecture is composed of modules which allow to communicate and control sensors and actuators. Among others, this layer is composed of modules interpreting sensors data to detect humans and objects and a module allowing to execute given trajectories by calling the adequate actuators.

Situation Assessment: The situation assessment is done by a software called TOASTER [Milliez 2016]. One of the functionalities of TOASTER is to build and maintain a consistent world state based on data coming from the sensorimotor layer.

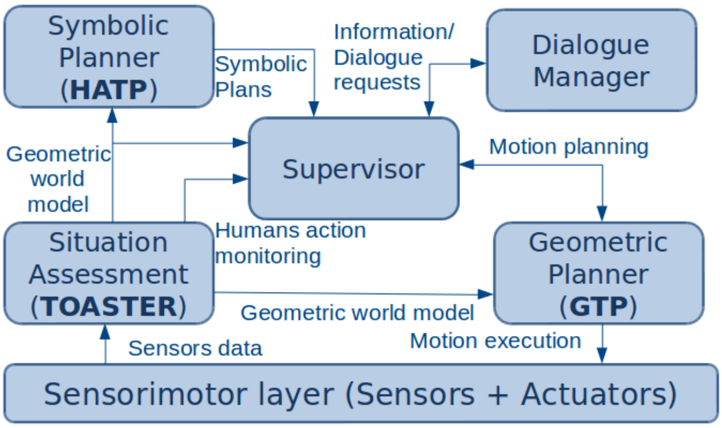

Figure 2.1: The global architecture for human-robot interaction implemented at LAAS-CNRS.

Geometric computations are done on this world state to compute symbolic predicates concerning the environment (e.g. *<object, isOn, support>*, *<object, isIn, box>*) and agents abilities and behaviors (e.g. *<object, isVisibleBy, human>*, *<object, isReachableBy, robot>*, *human, isLookingToward, object*). TOASTER is also in charge of perspective taking: the previous predicates are constantly estimated and maintained not only from the robot's point of view but also from the point of view of all humans concerned and perceived in the current context. All the data concerning the world state are stored and accessible through a database.

Geometric Planner: In order to perform actions and movements adapted to the human proximity, our architecture is equipped with a geometric task and motion planner called GTP [Waldhart 2016]. GTP allows to compute trajectories as well as objects placements and grasp. It does that at a level that is human understandable and readable by giving access to high level tasks such as Pick or Place while taking into account the human safety and comfort.

Symbolic Planner: For the robot to be able to synthesize Shared Plans, our architecture is equipped with HATP (Human-Aware Task Planner), a human-aware HTN (Hierarchical Task Network) task planner which allows the robot to compute and refine a plan both for itself and its humans partners, taking into account a number of social rules [Lallement 2014]. HATP has been specially designed to integrate a number of features that are meant to promote the synthesis of plans that are acceptable by humans and easily if not trivially understandable by them. It allows to specify the humans and robot capabilities in terms of actions they can execute. Several aspects such as human preferences and comfort, estimation of human effort to achieve a task in a given context and "social rules" are used in a cost-based approach to build "sufficiently good" human-robot Shared Plans. An example of a plan computed by HATP can be found in Fig. 2.2.

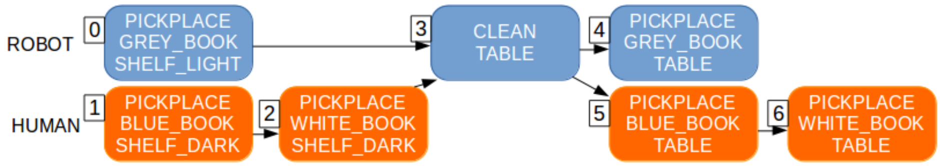

Figure 2.2: An example of a Shared Plan computed by HATP. This plan allows a human and a robot to clean a table by removing all objects on it, cleaning it and then putting back all previous objects.

Dialogue Manager: In order for the robot to communicate with humans, a basic dialogue manager has been integrated to the architecture. This module allows to give humans information concerning the environment (it verbalize predicates), ask basic questions (as asking if a human want to perform an action) and understand basic answers (mainly yes or no answers, the user can answer with buttons as there is no speech recognition in the system).

Supervisor: The last module of the architecture is the supervisor. It is the one in charge of controlling collaborative activities. It chooses the robot goals and monitors the Shared Plan execution. To do so, it estimates humans mental states concerning the Shared Plan and takes them into account to decide when to perform actions or to communicate (verbally and/or non-verbally). It also interprets the information coming from the Situation Assessment module in order to recognize human actions like Pick or Place with regard to the Shared Plan. This module is an extension of [Clodic 2009] and [Fiore 2016] and is the major technical contribution of this thesis. Its internal architecture will be detailed in the next section.

2.2 The supervisor architecture

The supervisor is composed of several modules and is fully implemented in ROS[1]. The complete scheme of its architecture can be found in Fig. 2.3, however, since the figure is quite complex, each composing part of the supervisor will be represented and described individually in the next subsections.

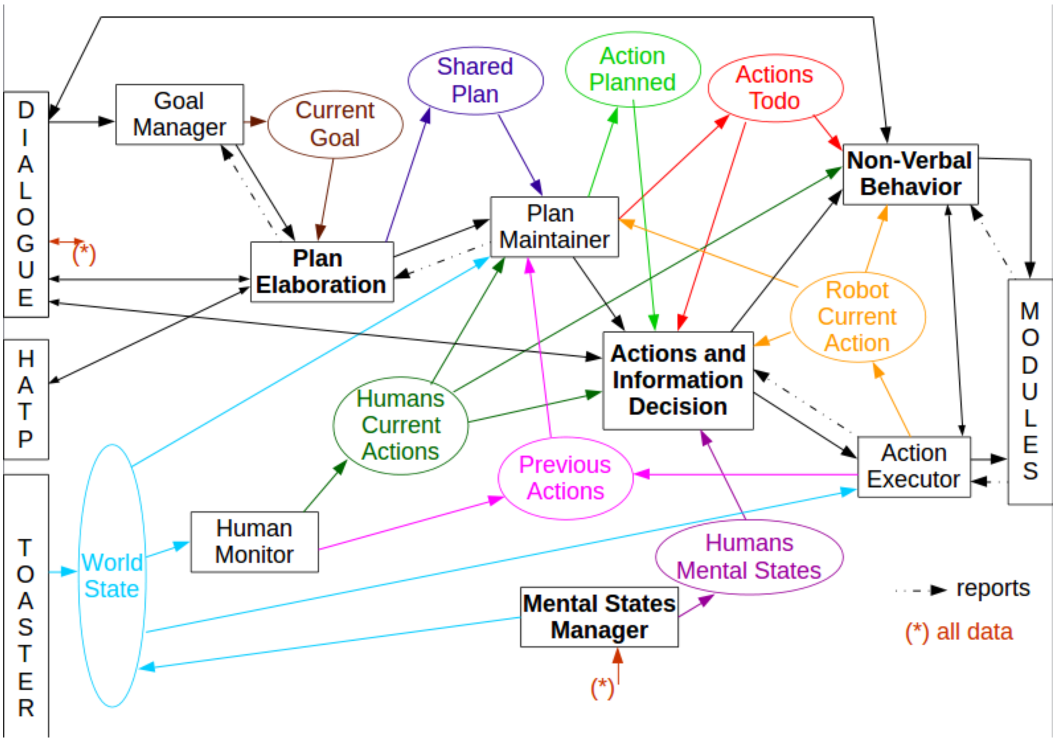

Figure 2.3: Architecture of the supervisor. The modules in bolt are the ones developed in this manuscript.

2.2.1 Goal Manager

The **Goal Manager** allows the robot to select and prioritize goals. It maintains a priority list of goals to perform. This list is updated with *insert*, *abort* or *halt* commands from dialogue or command line.

The chosen goal is published in order for the **Plan Elaboration** module to compute a Shared Plan to satisfy it. The **Goal Manager** sends *stop* and *suspend* orders to the **Plan Elaboration** from which it receives reports concerning the success of the plan or the impossibility to find a plan.

This module is, for now, really basic. An interesting extension would be to integrate data coming from an intention recognition module concerning humans

[1] http://www.ros.org/

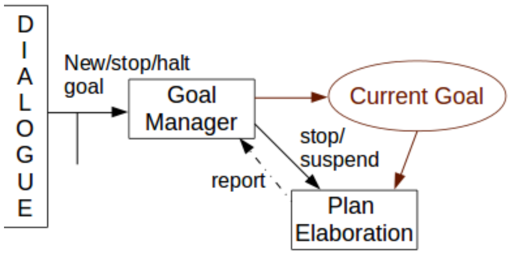

Figure 2.4: Interaction of the Goal Manager with the rest of the supervisor.

activities. This will allow the robot to choose if it should proactively offer its help based on this data and the goal orders it received.

2.2.2 Plan elaboration

Once a goal received from the **Goal Manager**, the **Plan Elaboration** module is in charge of finding a Shared Plan to perform it. To do so, the module is able to call HATP (the Human Aware Task Planner described in Sec. 2.1) to compute a plan and the dialogue module to validate the plan or ask for missing information. One of the contributions of this thesis concerns the elaboration of more flexible Shared Plans where some decisions are left to the execution. This work is in part done in this module and will be developed in Chapter **??**.

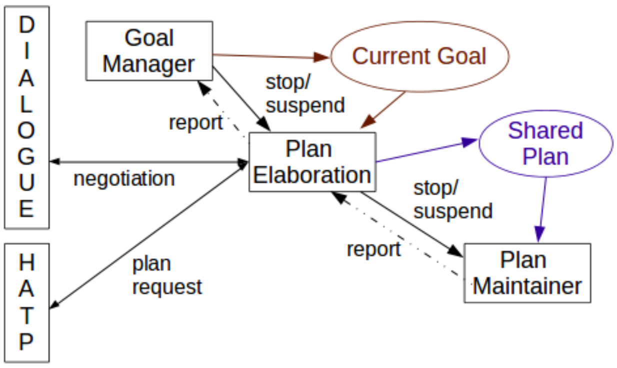

Figure 2.5: Interaction of the Plan Elaboration with the rest of the supervisor.

The computed Shared Plan is then published in order for the **Plan Maintainer** module to deal with it. The *stop* and *suspend* orders received from the **Goal Manager** are transmitted to the **Plan Maintainer** module from which it receives reports concerning the success, failure or need of adaptation of the Shared Plan.

2.2.3 Plan Maintainer

The **Plan Maintainer** module is in charge of monitoring the execution of the Shared Plan based on the current world state and current and past actions. It publishes the list of actions from the Shared Plan which need to be performed at a given moment and the list of actions from the Shared Plan which need to be done later. It also checks the consistency of the plan and reports to the **Plan Elaboration** module in case of failure or unexpected situations.

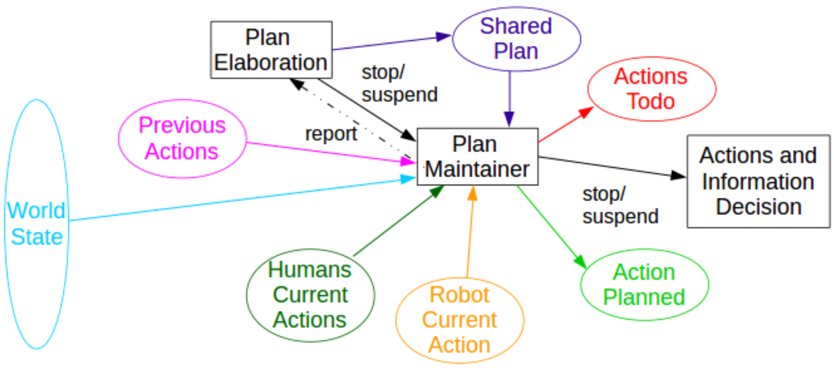

Figure 2.6: Interaction of the Plan Maintainer with the rest of the supervisor.

2.2.4 Human Monitor

The **Human Monitor** module allows to interpret the current world state which contains humans activity information in order to recognize basic humans actions like Pick or Place. This module is, for now, really basic as it is based mainly on distances between humans and objects. However, there is room for improvements by taking into account the context (e.g. the action the agent is supposed to perform) during action recognition or using probabilistic models.

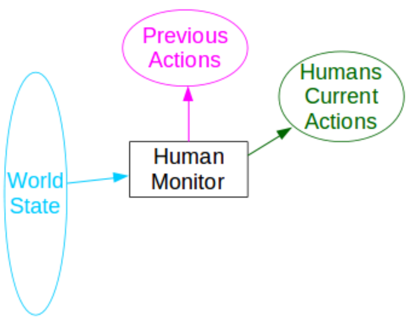

Figure 2.7: Interaction of the Human Monitor with the rest of the supervisor.

2.2.5 Mental State Manager

The **Mental State Manager** estimates the humans mental states concerning the current goal and Shared Plan. It bases its reasoning on all data published by the other supervisor modules and on the world states from all agents point of view given by TOASTER (see Sec. 2.1). This work is one of the thesis contributions and will be developed in Chapter **??**. The composition of the estimated mental states will be given in Sec. 2.3.

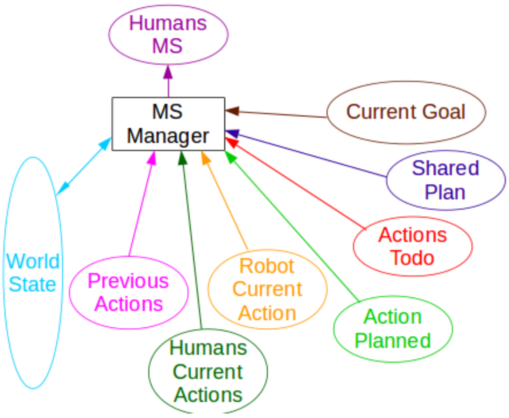

Figure 2.8: Interaction of the Mental State Manager with the rest of the supervisor.

2.2.6 Actions and Information Decision

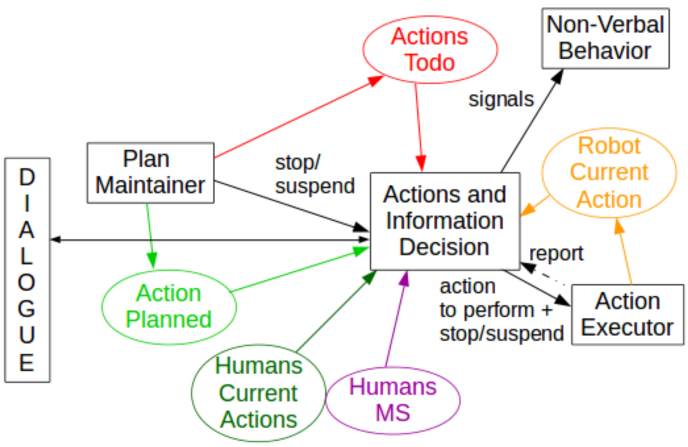

Figure 2.9: Interaction of the Actions and Information Decision module with the rest of the supervisor.

The **Actions and Information Decision** module allows the robot to decide which action to execute and which information to give. Its decisions are based on the lists of current, planned and to execute actions as well as on the humans mental states. The way the robot uses these mental states to give pertinent information to humans is one of the thesis contribution and will be developed in Chapter **??**. The decision of which action to execute has also been studied in this thesis and will be developed in Chapter **??**.

The **Actions and Information Decision** module sends commands to the **Action Executor** from which it receives reports. It also communicates with the dialogue module and the **Non-Verbal Behavior** module to give the correct information.

2.2.7 Action Executor

The **Action Executor** is in charge of supervising robot actions. It receives actions to execute and *stop* or *suspend* orders from the **Actions and Information Decision** and calls lower level modules to perform the given action in the best possible way.

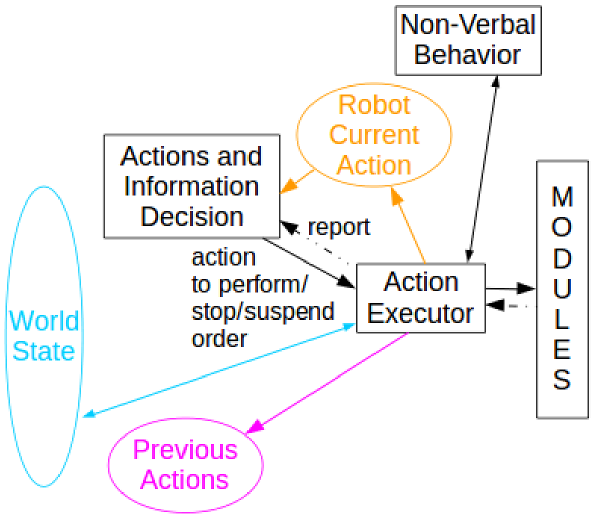

Figure 2.10: Interaction of the Action Executor with the rest of the supervisor.

2.2.8 Non-Verbal Behavior

This module allows to control the non-verbal behavior of the robot. In the current supervisor version, only the robot head behavior is concerned, but other types of non-verbal behaviors can be envisioned. The principles behind this module will be developed in Chapter **??**.

The robot head behavior is based on the current robot action, the humans activities and the actions to perform. The module communicates with the dialogue module in order to coordinate and calls lower modules to control the robot.

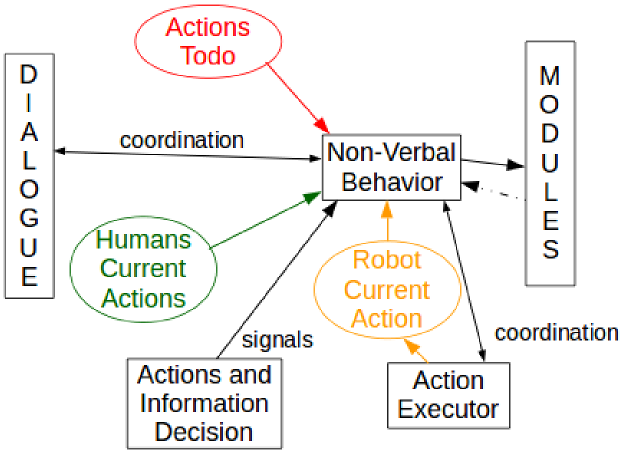

Figure 2.11: Interaction of the Non-Verbal Behavior with the rest of the supervisor.

2.3 Data representation

As seen in the previous section, several types of data are produced and used by the supervisor to take decisions. We will see now how we represent this data and this formalization will be used in the next chapters of the thesis.

The current state of the world from the robot point of view WS is composed of a set of predicates p:

$$p =< entity, attribute, value >$$

For example, the fact that an object is on a table will be represented as

$$< object, isOn, table >$$

.

A goal g is represented as:

$$g =< Name_g, Actors_g, Params_g, Obj_g >$$

where $Name_g$ allows to identify the goal, $Actors_g$ are the agents involved in the goal achievement, $Params_g$ are entities (agents or objects) used to define precisely the goal and Obj_g is a set of predicates representing the objective of the goal. For example, if the robot has for goal to clean the table of the kitchen in collaboration with Bob by removing all items on it, this goal will be represented as:

$$< Clean, < Robot, Bob >, < Kitchen\_table >, << NULL, isOn, Kitchen\_table >>>$$

. Finally, at its end, each goal g is stored and associated with a label noted $label_g$ which can be equal either to DONE or ABORTED.

Then, a Shared Plan SP is represented as:

$$SP =< id_p, A_p, L_p >$$

where id_p is used to identify the plan, A_p are the actions composing the plan and L_p the links representing the order the actions should be executed (causal links).

A link $l \in L_p$ is described as:

$$l = \langle prev_l,\ next_l \rangle$$

where $prev_l$ is the id of the action which needs to be achieved before the action with the id $next_l$ is performed.

The actions composing the plan A_p can be decomposed as:

$$A_p =< A_{prev}, A_{cur}, A_{next}, A_{later} >$$

where A_{prev} are the actions of the plan already executed, A_{cur} the actions currently executed, A_{next} the actions which can be performed according to causal links and actions preconditions and A_{later} the actions to be executed in the future. Each of the set of actions previously introduced can be decomposed as:

$$< A = A^R, A^H, A^X >$$

where A^R are the actions assigned to the robot, A^H the actions assigned to the human and A^X the actions not yet assigned. Indeed, we will see in Chapter **??** that not all actions are assigned to an actor during plan elaboration.

Finally, each action a in A_{prev} is associated with a label noted $label_a$ which can be equal either to DONE, FAILED or ABORTED.

An action a is represented as:

$$a =< id_a,\ Name_a,\ Ag_a,\ Params_a,\ Precs_a,\ Effects_a >$$

Where id_a is the action identifier and $Name_a$ represents its name. $Actors_a$ are the actors of the actions and $Params_a$ a set of parameters (objects or agents) which allows to define precisely the action. $Precs_a$ and $Effects_a$ are sets of predicates representing respectively the action preconditions and effects. For example the action for the robot to place an object on a support will be defined as:

$$< 0, place, Robot, < object, support >, << object, isHoldBy, Robot >>,$$

$$<< object, isOn, support >>>$$

.

The information concerning the state of the task is grouped in what will be called the Task State TS:

$$TS =< g_R, SP, WS >$$

with g_R the current goal of the robot, SP the current Shared Plan and WS the current world state from the robot point of view.

The robot also have a representation of humans mental state. The representation of the mental state $MS(H)$ of a human H is represented as:

$$MS(H) =< g_H, g_R(H), SP(H), WS(H) >$$

where g_H is the goal the robot estimates the human is engaged in, $g_R(H)$ is the goal the robot estimates the human thinks the robot is performing, and $SP(H)$ and $WS(H)$ are the estimation of the Shared Plan and the World State from the human point of view. $SP(H)$ is represented in the same way as the robot Shared Plan.

All these terms are reminded in the Appendix **??**

Bibliography

[Clodic 2009] Aurélie Clodic, Hung Cao, Samir Alili, Vincent Montreuil, Rachid Alami and Raja Chatila. *Shary: a supervision system adapted to human-robot interaction*. In Experimental Robotics, pages 229–238. Springer, 2009. (Cited in page 3.)

[Fiore 2016] Michelangelo Fiore, Aurélie Clodic and Rachid Alami. *On planning and task achievement modalities for human-robot collaboration*. In Experimental Robotics, pages 293–306. Springer, 2016. (Cited in page 3.)

[Lallement 2014] Raphaël Lallement, Lavindra de Silva and Rachid Alami. *HATP: An HTN Planner for Robotics*. CoRR, 2014. (Cited in page 2.)

[Milliez 2016] Grégoire Milliez. *Raisonnement sur le contexte et les croyances pour l'interaction homme-robot*. PhD thesis, INP-T, LAAS-CNRS, 2016. (Cited in page 1.)

[Waldhart 2016] Jules Waldhart, Mamoun Gharbi and Rachid Alami. *A Novel Software Combining Task and Motion Planning for Human-Robot Interaction*. In 2016 AAAI Fall Symposium Series, 2016. (Cited in page 2.)