MODEL BASED CONTINUAL PLANNING AND CONTROL FRAMEWORK FOR ASSISTIVE ROBOTS

A. Anier and J. Vain
Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia

Keywords: Assistive robotics, Model based control, Continual planning, Cognitive architecture.

Abstract: The paper presents a model based robot planning and control framework for human assistive robots of medical domain, namely for Scrub Nurse Robots. We focus on endoscopic surgery as one of the most relevant surgery type for the use of robotic assistants. We demonstrate that our framework provides means for seamless integration of sensor data capture, cognitive functions, model based continual planning and direct actuator control. The novel component of the architecture is a distributed continual planning system implemented based on the Uppaal model checking and testing tool suite. The distributed and the contract-based modular architecture of proposed framework enables flexible online reconfiguration and adaptability to various applications, but also safe installation of new software components on-the-fly.

1 INTRODUCTION

The assistive robotics sets high standards to cognitive capabilities, autonomy and movement precision of robots. Intuitively, it means understanding human intention and adequate reaction to it. Technically, it means human-in-the-loop collaborative action control, fusion of various sensor information, high accuracy actuation and reliable software implementation. Safety issues of action and trajectory planning become critical in the conditions where the robot shares user's working envelope and the contact due to the expected physical interaction for transfer of objects is frequent. This paper presents a new software integration framework for a *Scrub Nurse Robot* (SNR)(Miyawaki et al., 2005) focusing on distributed model based continual planning and control issues. The goal of a SNR is to learn the interactions between a surgeon and a scrub nurse during a laparoscopic surgery and to replace the (human) nurse on demand. One of the important aspects of incorporating the SNR in the collaborative action, e.g. when the human scrub nurse is occupied with other tasks avoiding the need for the surgeon to re-adapt to the robot partner and preserving the "original feel". An example of surgical scene with scrub nurse waiting for right moment of handing over an instrument to surgeon is depicted in Fig. 1.

A scrub nurse must hand a surgical instrument to a surgeon as soon as it is requested. If the scrub nurse has to spend time searching for the instrument after a request, the procedure is interrupted, valuable time is lost and an unnecessary burden is placed on the surgeon. That possibly reduces the quality and effectiveness of the operation. To avoid such delays the scrub nurse must be fully attentive to the activity in the operative field and anticipate accurately what a surgeon will need. For this to be possible the scrub nurse not only needs to know the surgical procedure as well as the surgeon does, but must also be highly disciplined. The "ideal" scrub nurse (if one exists) is able to pass a surgeon whatever is needed without any verbal order at the moment that the surgeon's hand is extended to receive it.

Figure 1: SNR intraoperative scene.(Miyawaki et al., 2005)

The goal of SNR software project is to develop a human-adaptive SNR capable of adapting to surgeons

Anier A. and Vain J..
MODEL BASED CONTINUAL PLANNING AND CONTROL FRAMEWORK FOR ASSISTIVE ROBOTS.
DOI: 10.5220/0003827104030406
In *Proceedings of the 2nd International Conference on Pervasive Embedded Computing and Communication Systems* (PECCS-2012), pages 403-406
ISBN: 978-989-8565-00-6

with various levels of skill and experience, and even to different personalities (and moods). In other words, the SNR should be able to function as an "ideal" scrub nurse. To attain this ideal, highly developed cognitive faculties such as machine vision and speech recognition as well as adaptive robotic arm path planning and targeting are required.

In conventional surgical operations a scrub nurse frequently has to handle an array of different instruments. It is very difficult to make the SNR adaptive to such busy operations. Therefore, the SNR prototype has been designed for endoscopic surgery which does not need many types of surgical instruments. The adaptivity of the SNR requires unsupervised learning by observing skilled nurses' actions and behavior during surgical operations.

On-line recognition and anticipating surgeon's motions while operating is essential to classify which motions are common to all surgeons and which are specific to individuals. This, in turn, will aid in anticipating a surgeon's needs and in adapting to the changes of procedure. On the other hand, the results of the investigation of intraoperative behavior have to be abstracted and memorized in the form of mathematical and/or formal models in order to reproduce the variety of motion trajectories that can be expected from various combinations of surgical procedures and varying external factors. The model of a nurse's behaviors as he or she reacts to other surgical staff (surgeon, assistant and others) serves as a high-level behavior specification for the SNR action planning.

The SNR's control architecture depicted in Fig. 2 comprises following components: MotionAnalysis Hawk position tracking system that is capable of measuring the position-tracking markers coordinates with precision more than 1 mm with sampling rate up to 200 frames per second. The surgeon's hand movement sampling data is passed to gesture recognition module that uses multiple recognition methods and voting automata(Vain et al., 2009)for detection of surgeon's current motion.The identified motion and its parameters are inputs for reactive motion planning that compares the observed movement of surgeon's hand with that of predicted by surgeon's behavior model and surgery scenario model, and makes the correction of the current model state, if necessary. By the corrected state information and surgery scenario model the next SNR action is planned and the resulting control parameters are transferred to the motion control unit. The information about surgeon's possible reactions predicted by the surgeon model is returned to the motion recognition module to restrict the decisions space when new movement is being recognized.

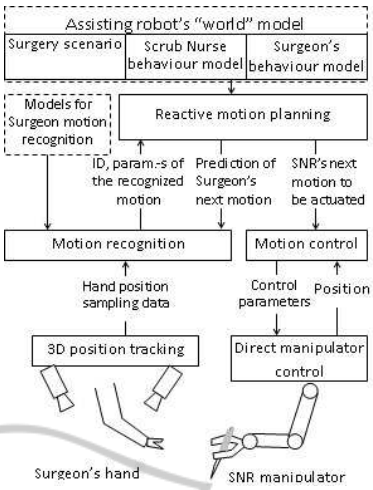

Figure 2: Control loop outline.

2 SOFTWARE ARCHITECTURE

The control architecture described above is implemented on the software framework written in C/C++ and Java and is designed around the Uppaal tool suite.

2.1 Data Acquisition

SNR doesn't have integrated vision but makes use of an external one. MotionAnalysis Hawk near-infrared active 3D measurement system (3DMS) is used for visual feedback. 3DMS is not the only source of information. There are various sensors to monitor the state of the robot and peripheral interfaces that contribute to the overall awareness of the environment.

Software architecture depicted in Fig. 3 unifies 3DMS data with other data acquisition sources for easier integration with upper architecture tiers. This includes capturing RFID information about the instrument positions in use. All the instruments are equipped with ceramic RFID tags that allow easy detection when the instrument is inserted to or removed from trocar cannula. Video image from the laparoscopic camera is used for anticipation of surgeon's motions, but it is not discussed in this paper. Last but not least, the middle-ware enables automated execution of capture and training sessions for algorithm evaluation, optimization and testing.

2.2 Data Analysis and Cognitive Functions

The robot control framework provides a common platform for integration of data acquisition and cog-

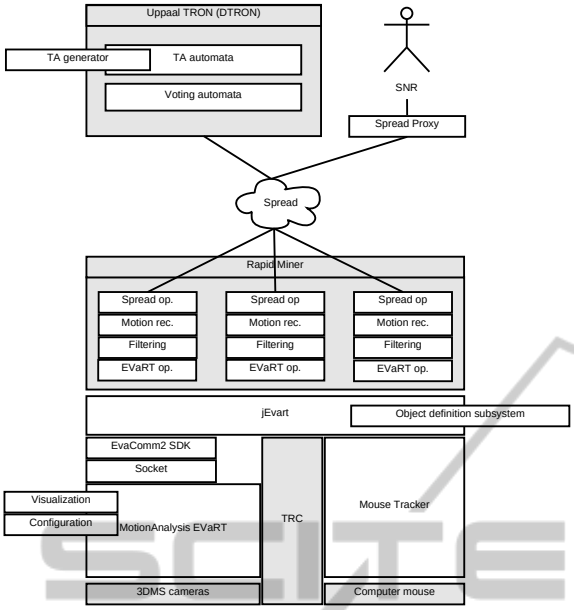

Figure 3: Architecture model.

nitive functions.

Data analysis and cognitive functions are implemented by means of data mining toolkit Rapid Miner. It includes hundreds of algorithms ranging from filtering to machine learning packaged into an integrated development environment. Rapid Miner is inspired by WEKA machine learning toolkit(Hall et al., 2009) but with extensive data visualization and analysis automation tools.

To make the Rapid Miner fit the SNR overall control architecture some custom plug-ins are implemented, specifically, the data acquisition components to capture the data available for analysis and visualization, but also the DTRON plug-in that bridges cognitive functions to deliberative control level functions. The deliberative control including overall safety monitoring, interaction learning and action planning is based on provably correct timed automata models.

3 DISTRIBUTED MODEL BASED CONTROL

The SNR timed automata based action planning and control make use of Uppaal tool suite(Behrmann et al., 2004). That allows manual construction of timed automata similar to visual programming paradigm - using your computer mouse to click around. Plus some limited functionality of C-like functions to be used upon various elements of the automata. Although the functions make it somewhat easy to specify state transitions their usage is prone to state space explosion. The Uppaal toolsuite includes an extension for Testing Real-time systems Online called *TRON*(Hessel et al., 2008)Although TRON was originally developed for conformance testing it also supports the functionality that is relevant for discrete control *To interface the TRON* model-based control stimuli with controllable object requires *"adapters"* on the to be able to intermediate and interpret the signals trafficking to and from the automata. TRON is designed for single *tester-testee* pair, but limits the scaling to $n > 1$ *testers* and $m > 1$ *testees*. The limitation of TRON usage is that it requires an extensive effort for adapter coding. When the adapter-tester pairs are tightly coupled every change in configuration requires re-wiring on both adapter ends .

Distributed TRON (*DTRON*) proposed in this paper is a custom framework built around the TRON tool to support multicast messaging between TRON instances running in parallel. In the ISO OSI networking architecture sense it implements the *whiteboard pattern* where the agents publish data and *the subscriber* agents get notified about this. On the other hand, it supports *the dependency injection* programming paradigm to make the *controller* controllable object pairs *loosely coupled* for better scaling. Multicast is a message sent not to one recipient but *n* recipients. *DTRON* is able to intercept the designated transitions within one control agent (practically in its model) and inform the other control agents of interests about it. The designation is defined by predicate on a *synchronized transition of the* controlling agent model.

This synchronization and communication between agents is implemented by means of multicast message passing that allows the agents to join and leave a multicast whenever they want without the need to re-configure existing infrastructure. The control agents do not need to be re-programmed when this happens. It only requires an agreement protocol on how the messages are defined and what data they carry when they traverse the multicast.

4 CONTINUAL PLANNING AND CONTROL

Continual planning(DesJardins et al., 1999) means the planning strategy when all interactions are not fully planned ahead, but reacting to the situation as it emerges. The controller knows the state of the control object it tries to reach, but doesn't have full control over the stimuli and behavior of it. Then the con-

troller stimulates the object step-by-step and reacts to responses of the object to drive it towards the goal.

Timed automata based planning and control suits for this kind of control scenario quite well due to its non-deterministic nature. Observations can easily be mapped to an automata locations and transitions encoding this non-determinism. Uppaal comes with a formal verification engine that can be used to establish weather a "plan" always drives the object to a desired state. An extreme case is a fully non-deterministic automaton that implies that it cannot be guaranteed or estimated which conditions should hold in order to guarantee that a planned target state of the control object is always reachable.

5 REACTIVE PLANNER

For planning and controlling the SNR action in non-deterministic situations reactive planning controller is synthesized on-the-fly based on the interaction model the SNR has learned by observing and recording Scrub Nurse and Surgeon's interactive behavior. The timed automata model learning algorithm used for that has been introduced in [vain2009humanrobot]. For synthesis of the reactive planning controller that guides the SNR action when being active is based on the interaction model the algorithm described in Reactive testing of non-deterministic systems by test purpose directed tester(Vain et al., 2011). Intended control goal of the SNR operation is encoded in the scenario automaton that specifies the sub goals of the control, their temporal order and timing constraints. Whenever one of the sub goals has been reached it triggers the reset of guard conditions in the interaction model and activates driving conditions to reach the subsequent goal or one of the alternatives if multiple equal goals are reachable. In case of violating the timing constraints or blocking an exception handling procedure or reset is activated and diagnostics recorded. Special care has been taken to address the safety precautions in SNR control. An independent safety monitoring process is running to check if all safety invariants are satisfied. Whenever safety violation is detected the emergency stop is activated.

6 CONCLUSIONS

The cognitive robot architecture framework described in this paper supports several innovative aspects needed for implementing assisting robots in different applications. Our experience is based on the Scrub Nurse Robot control architecture and software platform development exercise. We demonstrate that DTRON model-based distributed control framework provides flexible infrastructure for interfacing data acquisition and cognitive functions with the ones of deliberative control level planning and decision making. The architecture merges also a module for learning human interactions and model construction with reactive planning controller generator and runtime execution engine. The timed automata based interaction model learning, on-the-fly reactive planning controller synthesis and online safety monitoring in SNR are steps towards the concept of provably correct robot design.

REFERENCES

Behrmann, G., David, A., and Larsen, K. G. (2004). A tutorial on uppaal. In Bernardo, M. and Corradini, F., editors, *Formal Methods for the Design of Real-Time Systems: 4th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM-RT 2004*, LNCS, page 200–236. Springer–Verlag.

DesJardins, M. E., Durfee, E. H., Ortiz Jr, C. L., and Wolverton, M. J. (1999). A survey of research in distributed, continual planning. *AI Magazine*, 20(4):13.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Hessel, A., Larsen, K., Mikucionis, M., Nielsen, B., Pettersson, P., and Skou, A. (2008). Testing Real-Time systems using UPPAAL. In *Formal Methods and Testing*, page 77–117. Springer-Verlaag.

Miyawaki, F., Masamune, K., Suzuki, S., Yoshimitsu, K., and Vain, J. (2005). Scrub nurse robot system-intraoperative motion analysis of a scrub nurse and timed-automata-based model for surgery. *Industrial Electronics, IEEE Transactions on*, 52(5):1227 – 1235.

Vain, J., Kull, A., Kääramees, M., Maili, M., and Raiend, K. (2011). Reactive testing of nondeterministic systems by test purpose directed tester. In *Model-Based Testing for Embedded Systems.*, Computational Analysis, Synthesis, and Design of Dynamic Systems, pages 425–452. CRC Press - Taylor & Francis Group, Massachusetts, USA.

Vain, J., Miyawaki, F., Nomm, S., Totskaya, T., and Anier, A. (2009). Human-robot interaction learning using timed automata. In *ICCAS-SICE, 2009*, pages 2037 –2042.