MODEL BASED CONTINUAL PLANNING AND CONTROL FOR ASSISTIVE ROBOTS

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

Keywords: Model based control, Continual planning, Cognitive architecture, Online safety monitoring.

Abstract: The paper presents a model-based robot planning and control framework for human assistive robots - namely for Scrub Nurse Robots. We focus on endoscopic surgery as one of the most relevant surgery type for applying robot assistants. We demonstrate that our framework provides means for seamless integration of sensor data capture, cognitive functions for interpretation of sensor data, model based continual planning and actuation control. The novel component of the architecture is a distributed continual planning system implemented based on the Uppaal timed automata model-based verification and control tool suite. The distributed and modular architecture of the framework enables flexible online reconfiguration and easy adaptability to various application contexts. Online learning and safety monitoring functions ensure timely and safe updates of software components on-the-fly.

1 INTRODUCTION

The assistive robotics sets high standards to cognitive capabilities, autonomy and movement precision for robots. Functionally, it means understanding human intention and providing 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. Action and trajectory planning safety issues become critical in the conditions where the robot shares user's working envelope to achieve required physical interaction.

This paper presents a software integration framework for *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. The key aspect for incorporating the SNR in the collaborative action (e.g. when the human scrub nurse has to deal with unexpected emergencies) is to avoid the need for the surgeon to re-adapt to the changed partnerwhile still preserving the "original feel" and the accustomed workflow. A physical scene of a SNR example deployment is shown 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 the 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. The scrub nurse must be fully attentive to the activity in the operative field and anticipate accurately what a surgeon will need to avoid delays. 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 the SNR software project is to develop a human-adaptive SNR capable of adapting to

Anier A. and Vain J..
MODEL BASED CONTINUAL PLANNING AND CONTROL FOR ASSISTIVE ROBOTS.
DOI: 10.5220/0003783503820385
In *Proceedings of the International Conference on Health Informatics* (HEALTHINF-2012), pages 382-385
ISBN: 978-989-8425-88-1

surgeons with various levels of skill and experience, also to different personalities and moods . In other words, the SNR ought to function as an “ideal” scrub nurse. Highly developed cognitive faculties such as machine vision and speech recognition as well as adaptive robotic arm path planning and targeting are required to attain this ideal.

In conventional surgical operations a scrub nurse frequently has to handle an array of different instruments. It is difficult to make the SNR adaptive to such busy operations. Therefore, the SNR prototype has been designed for endoscopic surgery which only needs limited types of surgical instruments. The adaptivity of the SNR requires unsupervised learning by observing skilled nurses’ interactions and behavior during surgical operations.

Online recognition and anticipation of 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 of the following components: 3D position tracking system that is capable of measuring the position-tracking marker’s coordinates with precision more than 1 mm with sampling rate up to 200 fps. The surgeon’s hand movement sampling data is passed to gesture recognition module that uses multiple recognition methods in parallel. These methods of detecting operator’s current motions and the voting mechanism(Vain et al., 2009) maximize the confidence of the recognition. 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.

Such online conformance monitoring allows to correct the current model state with precision of minimum sampling error. By the corrected state information and surgery scenario model the next SNR action is planned and the resulting control parameters are transferred to the actuation control unit of SNR. The information about surgeon’s possible reactions predicted by the surgeon’s model is returned to the

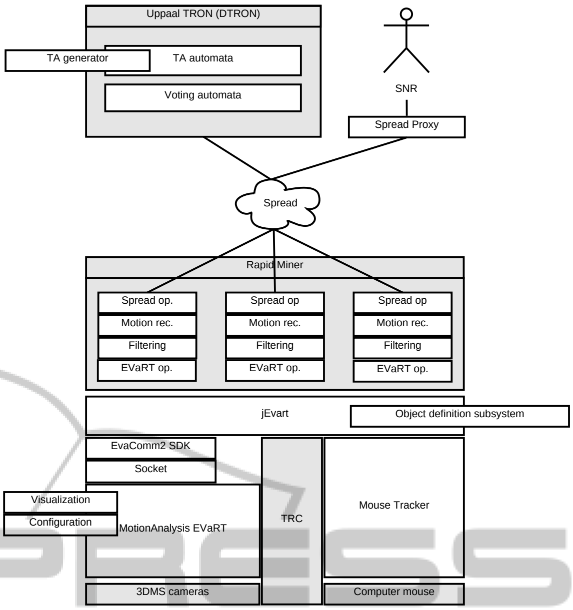

Figure 2: Architecture model.

motion recognition module for discrimination of the decisions space when new movement is being recognized.

The control architecture described above is implemented based on the open middle-ware platform discussed more thoroughly in the following sections.

2 SOFTWARE ARCHITECTURE

2.1 Data Acquisition

SNR doesn’t have integrated vision. Instead, the visual feedback control is implemented by means of external MotionAnalysis Hawk near-infrared active 3D measurement system (3DMS). 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 situation and context awareness. For instance, the position data of surgical instruments is backed by RFID readings of ceramic RFID tag positions that are attached to the instruments. Abdominal video imaging from laparoscopic camera provides more accurate information about the course of surgery.

Middle-ware *jEvart* unifies 3DMS data with other data acquisition sources and passes to data analysis and cognitive modules implemented by means of Rapid Miner tool - www.rapid-i.com.

2.2 Data Analysis and Cognitive Functions

The robot control framework and middle-ware provide a common platform for integration of data acquisition and cognitive functions.

Data analysis and cognitive functions are implemented by means of data mining toolkit Rapid Miner. It includes hundreds of algorithms ranging from filtering and clustering to machine learning packaged into an integrated development environment. Rapid Miner is inspired by WEKA machine learning toolkit(Hall et al., 2009) improved 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, it concerns 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 is based on provably correct timed automata models executed symbolically by DTRON tool.

3 DISTRIBUTED TRON

The SNR timed automata based action planning and control make use of Uppaal tool suite(Behrmann et al., 2004). Uppaal editor allows manual construction of timed automata in a way of visual programming paradigm. Limited functionality of various elements of the automata can be encoded using C-like functions. Although those functions make it somewhat easy to specify state transitions, their usage is prone to state space explosion. The Uppaal tool-suite includes an extension for *Testing Real-time systems Online* (*TRON*)(Hessel et al., 2008). Although TRON was originally developed for conformance testing, it also supports the functionality relevant to model-based discrete control. To interface the *TRON* model-based control module with controllable object requires "*adapters*" on the object side. Adapters intermediate and interpret the signals trafficking between the Uppaal automata and the control object. TRON was originally designed for single *tester-testee* pair and does not scale well with $n > 1$ *testers* and $m > 1$ *testees*. So it does not easily scale to distributed control applications. The main limitation of TRON usage is that it requires an extensive effort for adapter coding between controllers and control objects. When the adapter-controller pairs are tightly coupled every change in configuration requires re-wiring on both adapter ends.

Distributed TRON (*DTRON*) proposed in this paper is a 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 *publishers* publish data and *subscribers* get notified about this. On the other hand, it embraces *the dependency injection* programming paradigm to make the *controller*-controllable object pairs *loosely coupled* for much better scaling.

To multicast is to send a message not to one recipient but to *n* recipients. *DTRON* is able to intercept the designated transitions within one control agent (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. The synchronization and communication between agents is implemented by means of multicast message passing that allows the agents (dynamically) to join and leave a multicast whenever they want without the need to re-configure existing infrastructure. It only requires an agreement or protocol how messages are defined and what data they carry when they traverse the multicast.

4 CONTINUAL PLANNING AND CONTROL

Continual planning(DesJardins et al., 1999) denotes a planning strategy where the interactions between the controller and controllable object cannot be planned deterministically up front. The control signals have to be chosen depending on the situation as it emerges. The controller "knows" the state of the control object it tries to reach, but has limited control over stimuli or limited observation power of the control object behavior. The continual planning controller stimulates the object by limited set of stimuli step-by-step driving it towards the control goal by adjusting the stimuli to the control object responses.

Timed automata based planning and control suits for continual control due to its non-deterministic nature. Observations are mapped to automata structure and transition guards that encode the selection of stimuli to guide the (possibly) non-deterministic moves of the controllable object.

Uppaal comes with a formal verification engine that is used to establish weather a "plan" always drives the object to a desired state, provided the object responses are (at least partially) known. An extreme case would be a fully non-deterministic object that implies that it cannot be guaranteed or estimated which conditions should hold in order to reach the target state. This sets practical limits to the controllability for the SNR. If major deviations from prespecified scenario model occur the SNR would safely disengage human interaction from the working envelope and switches to manual override.

5 REACTIVE PLANNER

For continual planning and control the SNR actions in nondeterminstic situations are synthesized on-the-fly. The synthesis is 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 (Vain et al., 2009). The synthesis of reactive planning controller(Vain et al., 2011), that guides the SNR action when being active is based on the interaction model learned. The 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 resets on guard conditions of 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 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 disengagement procedure from continual planning unit 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 demonstrated 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 also incorporates 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 are steps towards the concept of provably correct robot design of *cognitive assisting robots*.

ACKNOWLEDGMENTS

This work was partially supported by the Estonian Science Foundation under grant No. 7667 and by Centre of Research Excellence in Dependable Embedded Systems - CREDES.

We want to thank Fuji Miyawaki for the productive discussions and suggestions on this subject.

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.

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.