

Ein prozessbasiertes Controllerkonzept zur Steuerung eines digitalen OP-Saals am Beispiel von OP:Sense

T. Beyl, L. Schreiter, J. Raczkowsky, H. Wörn

Karlsruher Institut für Technologie,
Institut für Prozessrechenstechnik, Automation und Robotik,
Karlsruhe, Deutschland

Kontakt: tim.beyl@kit.edu

Abstract:

Chirurgische Roboter und Integration von digitalen Operationssälen tragen maßgeblich zur Produktivität und Ergonomie der Systeme, bzw. zum Ergebnis von Operationen bei. Aktuelle kommerzielle Systeme sind entweder nichtstandardisierte integrierte Operationssäle wie der ORI von Karl Storz, oder alleinstehende Robotersysteme wie DaVinci von Intuitive Surgical.

Eine vollständige Integration von im OP eingesetzten Geräten und Prozessen kann dazu dienen sowohl die Produktivität als auch die Ergonomie und in Folge die Akzeptanz von Robotersystemen durch Chirurgen zu erhöhen. Im Rahmen von OP:Sense entwickeln wir neue Methoden zur Integration von Robotern in den Arbeitsablauf (Workflow) während einer OP. Der Roboter soll hierzu flexibel eingesetzt werden und dem Chirurgen zum korrekten Zeitpunkt zur Verfügung stehen und genau die ihm zugeteilten Aufgaben erfüllen. Im Folgenden wird ein wissensbasierter Ansatz beschrieben, der das Verfolgen des Arbeitsablaufs ermöglicht, sowie das System generisch auf Basis dieses Arbeitsablaufes steuern kann. Auch Verifikation und Arbeitsplansynthese sind möglich.

Schlüsselworte: Workflows, Controller, Chirurgische Robotik

1 Problemstellung

Medizinische Robotik entwickelt sich mit stetiger Verbreitung von Systemen und zunehmender Akzeptanz in der chirurgischen Routine zu einem immer wichtiger werdenden Forschungsthema. Systeme wie Intuitive Surgical DaVinci, Mako Rio, oder Hansen Medical Sensai X finden obwohl umstritten immer mehr Anklang im medizinischen Umfeld. Im Gegensatz zu konventionellen Operationsmethoden sind komplexe Kontrollstrategien notwendig. Allen aktuellen Systemen gemein ist, dass der Workflow im Operationssaal wesentlich durch das System selbst bestimmt ist. So bedient sich zum Beispiel DaVinci obgleich ein und dieselbe Operation durchgeführt wird, eines erheblich anderen Arbeitsablaufes verglichen mit normalen laparoskopischen Eingriffen. Der Roboter wird hierzu in einem speziellen Verfahren an den Patienten angedockt. Ähnliche Einschränkungen erfolgen bei fast allen kommerziell erhältlichen Robotersystemen. Zur Verbesserung dieser Situation ist es nach Meinung der Autoren nötig, Wissen über den Prozess selbst, also die durchgeführte Operation, zur Steuerung und Regelung des Systems mit einzubeziehen. Aus beobachteten und dokumentierten Operationen kann Wissen mit Hilfe von Workflows repräsentiert werden. Diese Workflows können als Basis für einen prozessgesteuerten Controller dienen. Im Folgenden wird dieses Konzept näher vorgestellt.

2 Material und Methoden

Im Rahmen von OP:Sense [1] entwickeln wir eine flexibel einzusetzende Forschungsplattform die abhängig von den Bedürfnissen des Chirurgen und der Operation konfiguriert werden kann. In OP:Sense werden Leichtbauroboter der Firma KUKA (Augsburg, Deutschland) als Assistenzsysteme direkt am Operationstisch eingesetzt. Diese Roboter verfügen über redundante Gelenke und können so ein und dieselbe Endeffektorposition unter Einnahme vielfältiger Gelenkstellungen einnehmen. Die hohe Nutzlast (7 kg) ermöglicht die Montage verschiedenartiger Werkzeuge. In verschiedenen Projekten wurden Versuche mit Schwingsägen, Frässystemen sowie minimalinvasiven Instrumenten durchgeführt. Die unterschiedlichen Verfahren wurden dabei entsprechend an die Anforderungen der Operation angepasst. OP:Sense liegt eine modulare Architektur auf Basis von ROS zugrunde. Die einzelnen Komponenten können über Konfigurationsdateien miteinander direkt zum Systemstart verbunden werden. Dies ermöglicht den Einsatz des Systems sowohl für minimalinvasive als auch für offene Interventionen. Ein weiterer Fokus von OP:Sense liegt auf der Überwa-

chung des Operationssaals, einerseits zur weitest gehenden Gewährleistung der Sicherheit, andererseits zur Verfolgung des chirurgischen Prozesses selbst.

Eine generische Steuerung auf Basis von Workflows wurde bisher nicht realisiert. Ein Workflow beschreibt den Ablauf eines Prozesses innerhalb einer bestimmten Domäne. Workflows im Operationssaal können unterschiedlicher Natur sein. So können sowohl pre-, post-, intra- als auch perioperative Workflows erstellt werden. Letztere können als Basis für einen Controller verwendet werden um sichere und intuitive Systeminteraktion zu realisieren. Während intraoperative Workflows die Operation selbst beschreiben und versuchen feingranulare Prozesse abzubilden, beschreiben perioperative Workflows die Prozesse innerhalb des Operationssaals zur Laufzeit der Operation sowie kurz davor und kurz danach. Dies umfasst Einsatz und Konfiguration von Geräten, das Erkennen der Positionen von Geräten und Personen sowie das Erkennen getätigter Aktionen. Zur Reduktion der Komplexität und zur Steigerung der Sicherheit und Intuitivität schlagen wir vor diesen Prozess als Basis für einen generischen Prozessbasierten Controller einzusetzen.

Workflowmanagement Systeme wie z.B. IBM Lotus, oder das hier eingesetzte YAWL [2] bieten Möglichkeiten zur direkten Interaktion mittels externer Software und damit auch zu den vorhandenen Geräten. In YAWL können sowohl externe Dienste direkt angesteuert werden, als auch Workflows über externe Dienste in die Laufzeitumgebung von YAWL geladen werden. Zusätzlich sind Verifikationsmethoden einsetzbar um die Korrektheit der auszuführenden Workflows also auch die Kompositionalität des Systems zu prüfen [3].

Unser prozessbasiertes Controllerkonzept basiert hierbei auf der Möglichkeit Systemkomponenten zur Laufzeit neu zu konfigurieren und auch die Kommunikation zwischen den einzelnen Komponenten dynamisch anzupassen. Hierzu werden Prozesse in YAWL regelbasiert abgearbeitet. Jeder elementare Bestandteil des Prozesses (im Folgenden: Task) kann mit einer, oder mehreren Aktionen auf dem System verknüpft werden. So können z.B. zur Laufzeit des Systems Trokarpunkte versetzt werden, oder von einer laparoskopischen zu einer offenen Operation gewechselt werden. Das System soll, wenn gewünscht verschiedene Kollisionsvermeidungsstrategien verfolgen, und ist in der Lage während der Laufzeit Funktionsüberprüfungen einzelner Komponenten durchführen.

Basis des vorgestellten Controllerkonzepts ist die Laufzeitumgebung von YAWL. Der Prozess wird hierbei über eine interaktive grafische Benutzerschnittstelle ausgewählt und entsprechend der vorhandenen Systembestandteile ausgeführt bzw. modifiziert. Der Prozess wird anschließend durch die Laufzeitumgebung von YAWL ausgeführt. Bausteine des Systems selbst sollen nicht statisch modelliert, sondern in einer Wissensbasis, repräsentiert als Ontologie enthalten sein. Auf diesem Wege können sowohl neue Komponenten und Arbeitsschritte, als auch komplette Prozesse in einer gemeinsamen Datenbank vorgehalten werden.

Die Ontologie ist hierbei in mehreren Abstraktionsstufen aufgebaut:

- Prozesslevel: Beschreibung der Prozesse, Äquivalente Prozesse und Transitionen zwischen einzelnen Tasks
- Tasklevel: Einzelne Tasks, Verknüpfte Systembestandteile, Prä- und Postoperative Bedingungen
- Systemlevel: Dienste und Konfigurationen der einzelnen Systembestandteile

Durch den Einsatz von Protegé (Ontologiesystem) ergeben sich weitere Möglichkeiten. So können z.B. Klassen von Komponenten definiert werden und die Verfügbarkeiten dieser mit einbezogen werden. Arbeitsabläufe könnten speziell für die Situation synthetisiert werden. Bei entsprechend großen Datenbanken könnten auch Notfalllösungen für Ausfälle einzelner Komponenten definiert und ausgeführt werden. In Ontologien werden hierzu Reasoner verwendet die auf Basis der Objekteigenschaften ähnliche, identische, oder unterschiedliche Objekte definieren können.

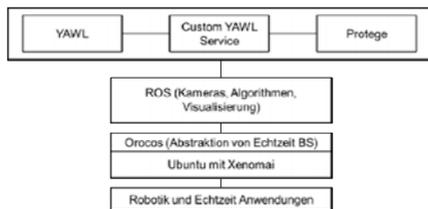


Abbildung 1: Softwarearchitektur in OP:Sense

Ein Problem ist die korrekte Modellierung der Ontologie. Es muss sichergestellt werden, dass nur kompatible Komponenten miteinander verknüpft werden können, sowie alle Vorbedingungen ausgeführt wurden bzw. alle Nachbedingungen noch ausgeführt werden. So kann ein Roboter seine Arbeit z.B. nicht ohne vorhergehende Kalibrierung und Registrierung aufnehmen. Abb. 1 zeigt die Softwarearchitektur, die für OP:Sense zum Einsatz kommt.

Mehrere Abstraktionsebenen wurden zur Übersichtlichkeit eingeführt. So werden auf der untersten Abstraktionsebene Echtzeitanwendungen realisiert, unter die klassischerweise viele Aufgaben aus der Robotik fallen. Als Echtzeitbetriebs-

system wird *Ubuntu* mit *Xenomai* verwendet. Um die Entwicklung zu vereinfachen wird *Orocos* als Abstraktionsebene von der Echtzeitprogrammierung verwendet. Dieses ist direkt an *ROS* angebunden und kann damit über Standardmechanismen von *ROS* angesteuert werden. In *ROS* selbst sind schließlich Algorithmen, Filter und die Module selbst, die die Anwendung ergeben realisiert. In der obersten Abstraktionsebene wird schließlich der prozessgesteuerte Controller umgesetzt. Zur Umsetzung des Controllers wird ein *Custom YAWL Service* verwendet, der die Anbindung an das Laufzeitsystem von *YAWL* ermöglicht, zusätzlich aber auch die Anbindung an *Protegé* und *ROS* realisiert.

Ein Ablaufplan wird wie folgt ausgeführt:

1. Laden des Ablaufplans bzw. Synthetisieren eines Ablaufplans aus der Datenbank
2. Verifikation des Ablaufplans und gegebenenfalls Ablehnung
3. Hochladen des Ablaufplans ins Laufzeitsystem von *YAWL*
4. Abarbeitung der einzelnen Tasks
5. Bei Start bzw. bei Beendigung eines Tasks wird der Custom *YAWL Service* informiert. Dieser startet Abfragen an *Protegé* um etwaige Aktionen auf *ROS*-Komponenten durchzuführen
6. Die *ROS* Komponenten selbst werden durch den *YAWL Custom Service* über *ROS* Mechanismen aktiviert bzw. deren Konfiguration geändert.

Diese Vorgehensweise ermöglicht eine flexible Abarbeitung nahezu aller auf dem System durchführbaren Aktionen.

3 Ergebnisse

Zur Veranschaulichung der Vorgehensweise wird im Folgendem ein sehr einfach strukturierter Workflow auf Basis einer realen Operation entsprechend für *OP:Sense* modifiziert. In [4] haben Mönlich et.al einen ausschließlich systembezogenen Workflow modelliert um erste Versuche mit workflowbasierten Controllern durchzuführen. Diese Workflows spiegeln zwar die Konfiguration des Robotersystems auf Basis fiktiver Workflows wider, können aber nicht ohne deutliche Modifikation auf klinische Szenarien angewendet werden. Aktuelle Ansätze zur Workflowmodellierung verwenden Videoaufzeichnungen zur Phasenidentifikation sowie probabilistische Ansätze zur online Identifikation während bzw. nach der Operation. In [5] wird ein Workflow von einer Videoaufzeichnung einer laparoskopisch durchgeführten Gallenblasenoperation abgeleitet. Abbildung 2 zeigt den schematischen Ablauf des identifizierten Workflows, modelliert in *YAWL*. Deutlich zu sehen ist der lineare Ablauf der Operation, der charakteristisch für viele chirurgische Interventionen ist. In einer Notfallsituation kann es jedoch vorkommen dass der laparoskopische Eingriff beendet werden muss und zu einer offenen Operation gewechselt wird.

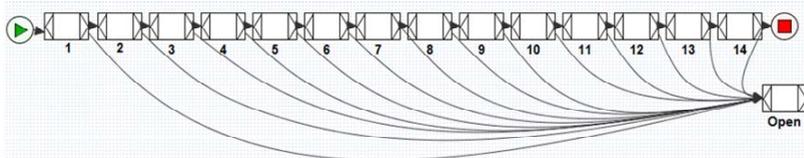


Abbildung 2: Workflow (1=C02 Insuflierung, 2=Trokar setzen, 3=Dissektion 1, 4=Klammern/Schneiden 1, 5=Dissektion 2, 6= Klammern/Schneiden 2, 7=Auslösung der Gallenblase, 8=Koagulation am Leberbett, 9=Verpacken der Gallenblase, 10=Entfernung der Gallenblase aus dem Körper, 11=Externe Säuberung, 12= Koagulation am Leberbett 2, 13=Trokar Entfernung, 14=Nähen)

Viele Schritte des Intra-operativen Workflows sind für die Steuerung des Operationssaals nicht notwendig und schwer automatisiert auswertbar. So können z.B. Interventionsschritte selbst, die keinerlei Interaktion mit Geräten, wie z.B. Ultraschallgeräten, erfordern vernachlässigt werden. Im Falle der Cochlecystomie reduziert sich der Workflow für eine robotergestützte Intervention damit auf Abbildung 3. Dieser Workflow spiegelt aufgrund der niedrigen Komplexität der OP ein sehr einfach auszuführendes Szenario wider. Dabei wird am Anfang einer robotergestützten OP ein Trokar eingesetzt und anschließend der Roboter eingeführt. In der Phase des „operieren“ wird auf den Intraoperativen Workflow zurückgegriffen. Sollte es zu einer Notfallsituation kommen, kann jederzeit in den Modus der offenen OP gewechselt werden, was wiederum einen neuen Workflow initialisieren würde, der hier nicht modelliert wurde.

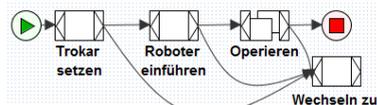


Abbildung 2: Perioperativer Workflow

Deutlich wird hierbei, dass sich der Workflow der Operation häufig vom Workflow der zur Steuerung des Systems notwendig ist unterscheidet. Das Ableiten eines systemspezifischen Workflows kann sich je nach Operation schwierig gestalten. So bestehen für viele Operationen keine Erfahrungen, wie der Einsatz eines Robotersystems sich auf den Workflow auswirkt. Je häufiger Assistenzsysteme während einer Operation eingesetzt werden sollen, desto komplexer gestaltet sich der Workflow, der auf den Controller zu applizieren ist. Je weniger eindeutig Entscheidungen im Workflow sind, desto komplexere probabilistische Ansätze müssen verwendet werden um eine korrekte Klassifikation zu erzielen. Elementare Bestandteile eines Workflows können für mehrere Abläufe verwendet werden, da viele Operationen sich vom Kontrollschema kaum unterscheiden (z.B. Telemanipulation in der Weichteilchirurgie ohne preoperative Bildgebung).

4 Diskussion

Noch nicht inbegriffen in den Workflow sind die Fehleridentifikation und -behandlung, sowie die notwendigen Schritte zum Wechsel zur offenen Operation. Komplexere Interventionen wie Epilepsiechirurgie, wie sie z.B. im EU-Projekt ACTIVE verfolgt wird, sehen verschiedenste Betriebsmodus und Sicherheitsmechanismen vor, die eine intuitive und sichere Interaktion gewährleisten sollen. So muss der Roboter jederzeit im korrekten Betriebsmodus (Telemanipulation, handgeführt, automatisch) sein und dabei die jeweils relevanten Sicherheitskriterien einhalten. Auch sind Übergänge nicht unbedingt einfach zu identifizieren und benötigen zusätzlich zu regelbasierten Systemen unter Umständen probabilistische Ansätze zur eventuell fehlerbehafteten Identifikation der Transitionen.

Das hier vorgestellte System ist nicht ausschließlich beschränkt auf den Einsatz von robotergestützten Operationen, sondern ist in der Lage verschiedenste Komponenten, solange sie in ROS eingebunden sind und in der Ontologie modelliert sind anzusteuern. So ist es denkbar grafische Benutzeroberflächen während der Laufzeit an die Bedürfnisse des Chirurgen anzupassen, Bildaufnahmeparameter für bildgebende Geräte zu setzen, oder auch bestimmte Aspekte der Operation zur späteren Wiedergabe aufzunehmen. Dennoch ist für eine vollständige Applizierbarkeit des Systems eine genaue Anforderungsanalyse erforderlich. Die zur Ausführung notwendigen Anteile der Operation müssen vollständig modelliert sein und Kompatibilität der verwendeten Komponenten muss gewährleistet sein. Damit ergibt sich ein flexibel konfigurierbares System, dessen Ausführungsqualität aber maßgeblich von der Qualität der Anforderungsanalyse und der Modellierung der Komponenten in der Ontologie abhängt.

6 Danksagungen

Diese Arbeit wurde durch das FP7 Framework Programm der Europäischen Union innerhalb der Projekte „EuroSurge“ unter Grant Nr. 288233 und „Active Constraints Technologies for Ill-defined or Volatile Environments (ACTIVE)“ unter Grant Nr. 270460 unterstützt und gefördert.

7 Referenzen

- [1] P. Nicolai , T. Brennecke , M. Kunze , L. Schreiter , T. Beyl , Y. Zhang , J. Mintenbeck , J. Raczkowski , and H. Wörn, *The OP:Sense surgical robotics platform: first feasibility studies and current research*, International Journal of Computer Assisted Radiology and Surgery, pp. 136-137, 2013.
- [2] W. M. Van Der Aalst and A. H. Ter Hofstede, *YAWL: yet another workflow language*, Information Systems, vol. 30, pp. 245-275, 2005.
- [3] M. Capiluppi, L. Schreiter, P. Fiorini, J. Raczkowski, and H. Woern, *Modeling and Verification of a Robotic Surgical System using Hybrid Input/Output Automata*, European Control Conference2013.
- [4] H. Mönlich, J. Raczkowski, and H. Wörn, *Model Checking for Robotic Guided Surgery*, in Electronic Healthcare, ed Heidelberg Springer 2010.
- [5] T. Blum, H. Feußner, and N. Navab, *Modeling and segmentation of surgical workflow from laparoscopic video*, in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010, ed: Springer, 2010, pp. 400-407.