

1 Thema

InProX: Industrial Programming by Example

2 Zeitraum

Sommersemester 2014 und Wintersemester 2014/2015

3 Veranstalter

Dipl.-Inf. Michael Lybecait <michael.lybecait@tu-dortmund.de>, Tel. 7756

Dipl.-Inf. Stefan Naujokat <stefan.naujokat@tu-dortmund.de>, Tel. 7734

Prof. Dr. Bernhard Steffen <steffen@cs.tu-dortmund.de>, Tel. 5801

Informatik Lehrstuhl 5, Otto-Hahn-Straße 14, Raum 130/131

4 Aufgabe

Verpackungsautomation hat sich zu einer eindrucksvollen Disziplin entwickelt, bei der spezialisierte Roboter in rasender Geschwindigkeit komplexe Verpackungsformen mit verschiedenen Produkten befüllen. Ein konkretes Beispiel hierfür sind Deltaroboter (siehe Abb. 1) für sogenannte „Pick and Place (PnP) Packaging“-Aufgaben. Sie sind in der Lage, Produkte wie verschiedene Kekse oder Pralines auf einem Fließband visuell zu lokalisieren, zu identifizieren, zu greifen und in das dafür vorgesehenen Fach der Schachtel zu platzieren. Allerdings ist die Erstellung zuverlässiger und effizienter Programme für derartige Anlagen [1, 6] immer noch ein hochkomplexes, sehr technisches Unterfangen, welches immer und immer wieder erfolgen muss, wenn sich das Verpackungsszenario ändert. Um diesem Flaschenhals in der Produktion entgegenzuwirken wurden verschiedene visuelle Simulationsumgebungen entwickelt, die die Roboterprogrammierung im Stil von „what you see is what you get“ (WYSIWYG) ermöglichen. Allerdings sind diese Produkte häufig sehr generisch, was dazu führt dass ihre Ergebnisse nicht ansatzweise an die Leistungsfähigkeit von manuell geschriebenen Programmen herankommen.

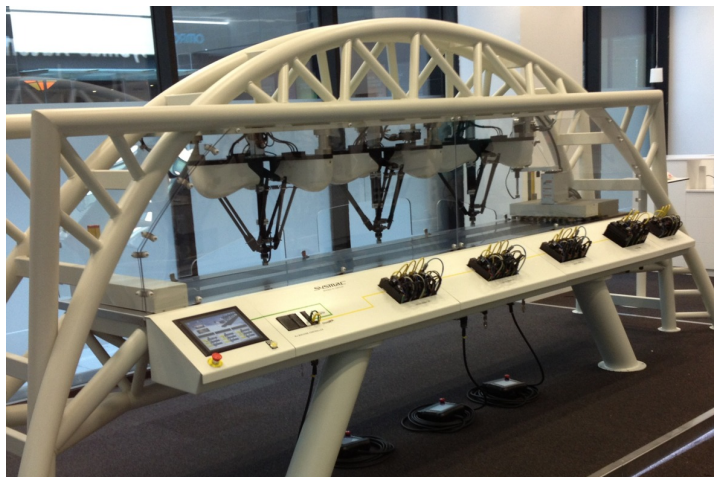


Abbildung 1: Omron „Pick and Place“-Anlage mit drei Deltarobotern; siehe auch [YouTube/watch?v=D45nkeKOb5o](https://www.youtube.com/watch?v=D45nkeKOb5o)

Ziel der Projektgruppe ist es, einen für PnP-Aufgaben spezialisierten Ansatz zu entwickeln und umzusetzen, in dem die PnP-Anlage nicht programmiert wird, sondern das von ihr erwartete Verhalten über Beispielausführungen und „Trial and Error“ *lernt*. Konkreter soll dieser lern-orientierte Ansatz des „Programming by Example“ [8] folgende Schritte unterstützen:

- In einem grafischen Simulator können Anforderungen an eine PnP-Aufgabe manuell „durchgespielt“ werden.
- Mehrere dieser aufgezeichneten Szenarien werden zu einem ausführbaren Steuerungsprogramm generalisiert.
- Im Simulator wird das inferierte Programm vom Entwickler validiert und ggf. korrigiert.
- Optimierter, effizienter Code wird für das reale System erzeugt.

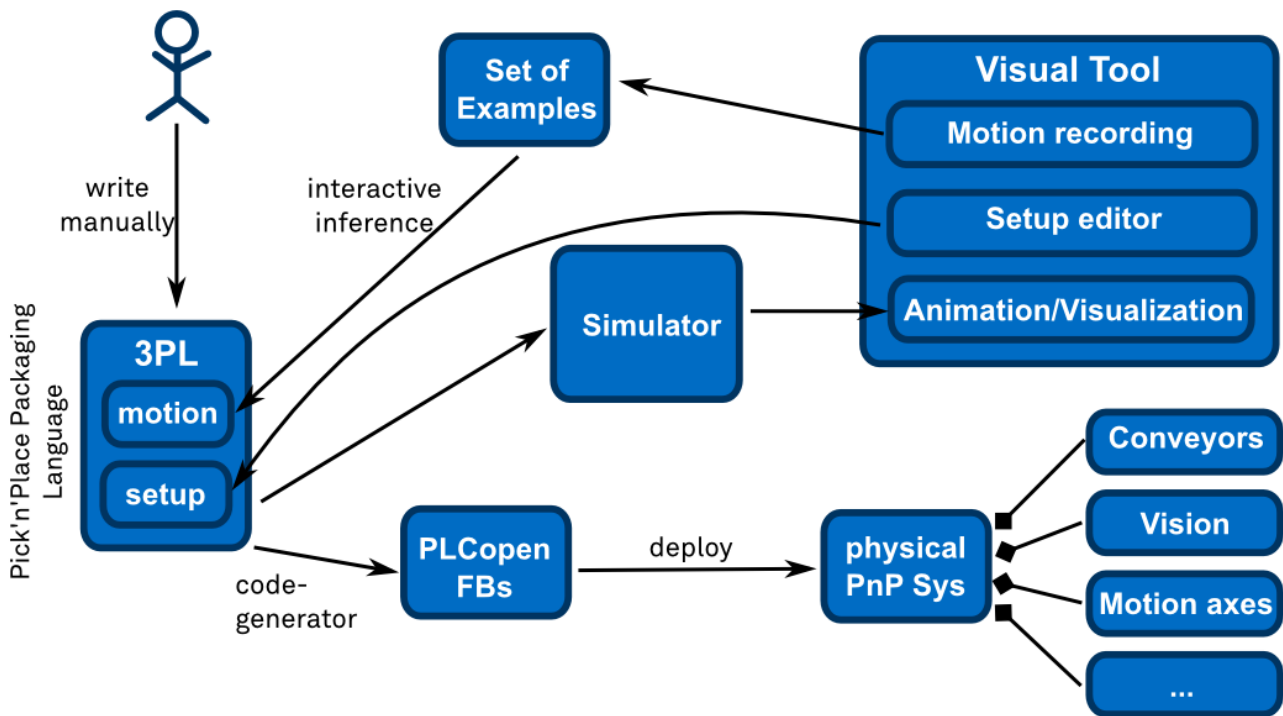


Abbildung 2: Übersicht der verschiedenen Aspekte des Projekts.

Die Arbeit der Projektgruppe wird von den Spezialisten für Verpackungsautomation der Firma *Omron Europe B.V.* (Standorte Barcelona und Düsseldorf) unterstützt, die den aktuellen „State of the Art“ bezüglich PnP-Programmierung, grafischer Umgebungen etc., deren Einschränkungen sowie ihre Visionen präsentieren. Sie verfolgen den Fortschritt der PG und kommentieren die Anwendbarkeit der Ergebnisse in der Praxis.

Abbildung 2 illustriert die verschiedenen Aspekte, die in der Projektgruppe beachtet werden müssen. Eine domänenspezifische Sprache (domain-specific language, DSL), die auf die Beschreibung von PnP-Anlagen und -Aufgaben spezialisiert ist muss von der PG entwickelt und umgesetzt werden. Diese „Pick and Place Packaging Language“ (3PL), die in Kooperation mit Omron basierend auf deren bisherigen Lösungen entwickelt wird, bildet die Basis auf der die weitere Implementierung der Projektgruppe aufbaut. Die tatsächliche Umsetzung (d.h. ihre konkrete Syntax) kann textuell wie eine Programmiersprache oder grafisch, mit Knoten und Kanten in einem Kontrollflussmodell, sein; oder eine Mischung aus beidem. Programme in dieser Sprache können entweder manuell geschrieben bzw. modelliert werden, oder sie werden anhand von Beispielen inferiert, d.h. aus einer Menge von im Simulator aufgezeichneten Sortiervorgängen generalisiert.

Die Entwicklung von Konzepten, wie aus einer Menge von Aufzeichnungen ein geeignetes 3PL-Programm erzeugt werden kann, bildet einen Hauptteil der PG. Sie sollen auf Basis verschiedener formaler Methoden, insbesondere Maschinenlernen [5] und Synthese-basierter Vervollständigung von Spezifikationen [7] realisiert werden.

Natürlich ist dieses gesamte Vorgehen nur dann sinnvoll, wenn letztendlich ausführbarer Code für die realen PnP-Anlagen generiert werden kann. Allerdings ist dies im Rahmen der Projektgruppe nur ein optionales Ziel und kann in folgenden Projekten und/oder Masterarbeiten noch vertieft werden.

Ziele

Zunächst soll die Projektgruppe die domänenspezifische Sprache 3PL entwickeln, die ausdrucksstark genug sein muss, um realistische „Pick and Place Packaging“-Aufgaben zu beschreiben, sowie basierend auf einer existierenden Roboter-Simulationsumgebungen (z.B. v-rep [3] oder ROS [2]) deren

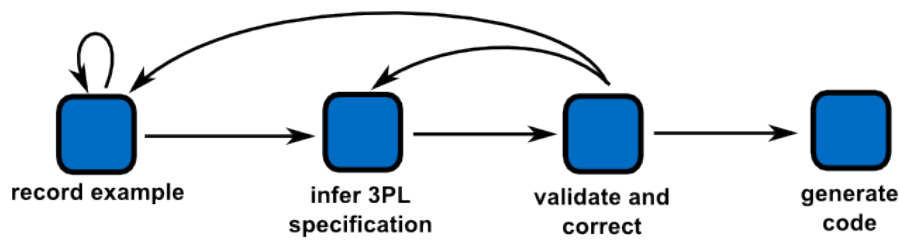


Abbildung 3: Geplanter Workflow für das interaktive 3PL-Inferenzsystem.

Ausführung visuell darstellen zu können. Für den gewählten Simulator soll zudem eine Aufzeichnungsfunktion manueller Eingaben implementiert werden. Die aufgezeichneten Beispiele müssen dann in 3PL transformiert werden. Geplant ist ein semi-automatisches System, bei dem Vorschläge für die Bewegungen automatisch erzeugt werden, so dass der Programmierer diese validieren und beispielhaft nicht beabsichtigtes Verhalten korrigieren kann. Basierend auf den ursprünglichen Beispielen sowie den zum aktuellen Vorschlag gemachten Korrekturen soll ein weiterentwickelter Vorschlag hergeleitet und dem Entwickler präsentiert werden. Dieses Vorgehen soll fortgeführt werden, bis der Programmierer mit dem Vorschlag einverstanden ist (siehe Abbildung 3).

Technologien

Zur Definition einer domänenspezifische Sprache wird ein Modell benötigt, das die Sprache hinreichend genau beschreibt. Metamodellierung soll hier eingesetzt werden, um die abstrakte Syntax der Sprache zu definieren. Ein Metamodell ist im Grunde genommen ein Modell, das die Beschaffenheit von Modellen beschreibt. Es wird also ein Metamodell benötigt, das die 3PL beschreiben kann. Als Framework, mit dem Modelle beschrieben werden können, soll das Eclipse Modeling Framework (EMF) [4] verwendet werden.

Als Bestandteil des Eclipse Modeling Projects wird EMF dazu genutzt, ein zentrales Modell zu definieren, das verschiedenen Zwecken wie beispielsweise der Datenserialisierung oder der Codegenerierung dient. Die wichtigste Komponente des EMF ist Ecore. Ecore ist das Metamodell, mit dessen Hilfe die abstrakte Syntax einer domänenspezifischen Sprache definiert werden kann. Im Verlauf der PG soll Ecore zur Definition der 3PL benutzt werden.

Innerhalb des Eclipse Modeling Projects existieren auch Werkzeuge, mit deren Hilfe sich die konkrete Syntax von 3PL definieren lässt. Beispielsweise kann eine textuelle Syntax mit Xtext definiert werden. Xtext verwendet das Ecore-Modell um Texteditoren mit Codevalidierung und Syntaxvervollständigung für die domänenspezifische Sprache zu generieren. Als Beispiel für Frameworks, die die Entwicklung von graphischen Sprachen unterstützen sei das Graphiti Projekt erwähnt.

Es sollen 3PL-Programme aus zuvor aufgezeichneten Bewegungsabläufen hergeleitet werden. Hierfür müssen sowohl positive als auch negative Beispiele vorhanden sein. Die PG-Teilnehmer sollen eine Methode entwickeln, Bewegungen innerhalb der Simulationsumgebung durchzuführen, aufzuzeichnen und diese in 3PL Beschreibungen zu übertragen. Existierende Simulationsumgebungen wie v-rep verfügen über Aufnahmekomponenten, deren Eignung evaluiert und gegebenenfalls angepasst werden soll. Mittels verschiedener Lernverfahren sollen aus diesen Aufzeichnungen Verhaltensweisen für zuvor bestimmte Aufgaben hergeleitet werden. Ein geeignetes Framework für die Umsetzung der Lernverfahren ist LearnLib [5], welche eine Schnittstelle zur Modellierung verschiedener Lernszenarien zur Verfügung stellt.

5 Teilnahmevoraussetzungen

Vorausgesetzt werden:

- Fundierte Kenntnisse in mindestens einer objektorientierten Programmiersprache, zum Beispiel Java oder C(++|#)
- Elementare Kenntnisse über Modellierungstechniken, wie sie zum Beispiel in den Vorlesungen „Formale Methoden des Systementwurfs“, „Virtualisierung und Compilation“, „Softwarekonstruktion“ und „Dienstleistungsinformatik“ vermittelt werden.

Wünschenswert sind zudem:

- Kenntnisse im Umgang mit Eclipse-Werkzeugen und -Programmierung
- Kenntnisse im Bereich der Steuerungs- und Regelungstechnik

6 Minimalziel

Die Minimalziele der PG sind:

- Entwicklung einer domänenspezifischen Sprache zur Beschreibung von „Pick and Place Packaging“-Aufgaben und Maschinenkonfigurationen (3PL).
- Realisierung einer visuellen Simulationsumgebung, die 3PL-Programme ausführen kann.
- Hinzufügen von Aufzeichnungsmöglichkeiten für Beispiele.
- Entwicklung eines Frameworks zur iterativen und semi-automatischen 3PL-Inferenz aus Beispielen und Gegenbeispielen.

7 Literatur

- [1] IEC Standard 61131-3 - Programmable Controllers - Part 3: Programming Languages. http://webstore.iec.ch/webstore/webstore.nsf/Artnum_PK/47556.
- [2] ROS: Robot Operating System. <http://www.ros.org/>.
- [3] v-rep: virtual robot experimentation platform. <http://www.coppeliarobotics.com/>.
- [4] Richard C. Gronback. *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*. Addison-Wesley, Boston, MA, USA, 2008.
- [5] Falk Howar, Malte Isberner, Maik Merten, and Bernhard Steffen. LearnLib Tutorial: From Finite Automata to Register Interface Programs. In *Proceedings of ISoLA (1)*, volume 7609 of *Lecture Notes in Computer Science*, pages 587–590. Springer Verlag, 2012.
- [6] Karl-Heinz John and Michael Tiegelkamp. *IEC 61131-3: Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids*. Springer, 2 edition, 2010.
- [7] Anna-Lena Lamprecht, Stefan Naujokat, Tiziana Margaria, and Bernhard Steffen. Synthesis-Based Loose Programming. In *Proceedings of the 7th International Conference on the Quality of Information and Communications Technology (QUATIC 2010)*, September 2010.
- [8] Henry Lieberman, editor. *Your Wish is My Command: Programming by Example*. Morgan Kaufmann, 2001.

8 Rechtlicher Hinweis

Die Ergebnisse der Projektarbeit und die dabei erstellte Software sollen der Fakultät für Informatik uneingeschränkt für Lehr- und Forschungszwecke zur freien Verfügung stehen. Darüber hinaus sind keine Einschränkungen der Verwertungsrechte an den Ergebnissen der Projektgruppe und keine Vertraulichkeitsvereinbarungen vorgesehen.