Technische Umsetzung des Anwendungsszenarios „Interaktion im vernetzten Heim“ im Projekt AutoPnP

Autoren: Marcel Patzlaff (DAI-Labor)
Erdene-Ochir Tuguldur (DAI-Labor)

Inhalt

1. Anwendungsszenario und benötigte Plug&Play Fähigkeiten

Das Anwendungszenario „Interaktion im vernetzten Heim“ befasst sich mit der Integration von Servicerobotern in eine vernetzte Heimumgebung. Die Roboter werden dadurch in die Lage versetzt werden, ihre Dienstabläufe zu optimieren und auf die verfügbare Sensorik und Aktuatorik direkt oder über die von der Umgebung angeboten Softwarekomponenten zuzugreifen und eine Steigerung der Dienstperformanz zu erzielen. Weiterhin wird dem Nutzer durch diese Integration ein mobiler Akteur der Umgebung zur Verfügung gestellt mittels dessen eine natürlichere Interaktion zwischen Mensch und Wohnumfeld realisiert werden kann. Um diese Möglichkeiten aufzuzeigen werden basierend auf den Ergebnissen des Projektes entsprechende Anwendungen realisiert.

Die Applikation für intelligente **Hol- & Bringdienste** lässt den dazu befähigten Roboter die geforderten Objekte suchen und dem Nutzer bringen. Ohne die Interaktion mit der Umgebung ist dies ein Standardvorgehen: der Roboter sucht die Räume der Wohnung ab bis das gewünschte Objekt gefunden wird. Wird allerdings die Vernetzung mit der Umgebung genutzt, so kann auf deren Sensorik, zum Beispiel die Deckenkameras, und die dazugehörige Objekterkennung zugegriffen werden. Dadurch ist der Roboter in der Lage das Anfahren von Bereichen zur Objektsuche zu vermeiden und stattdessen über diese zusätzlichen Suchmöglichkeiten die Objektdetektion durchzuführen. Dies erlaubt eine wesentliche schnellere und zielgerichtete Bearbeitung der Bring-Aufgabe.

Zur Darstellung der **Interaktionsmöglichkeiten** zwischen Nutzer und Serviceroboter werden zudem Anwendungen realisiert, die sich über ansprechende Benutzerschnittstellen anstoßen lassen. Dazu zählen Assistenzanwendungen für Hol- & Bringdienste, Bodenbegutachtung und -reinigung sowie für die Sicherung des Wohnumfeldes. Letztere Anwendung verwendet die Kontaktsensorik (an Türen und Fenster) sowie vorhandene Umweltsensorik um den Zustand zu evaluieren und an den Nutzer weiterzugeben.

In den Anwendungen können auch mehrere Roboter zum Einsatz kommen, wenn sie sich beispielsweise in ihren Fähigkeiten ergänzen. Dies wird im Szenario entsprechend auch gezeigt.

1.1. Hardware Plug&Play

Das Hardware Plug&Play kann in verschiedenen Ausprägungen dargestellt werden. Einerseits ist es darüber möglich, direkt am Serviceroboter Hardware anzuschließen und sie automatisch ins System einbinden zu lassen und damit den Anwendungen zur Verfügung zu stellen. Andererseits ist es auch möglich, Geräte in der Heimumgebung anzuschließen. Auch hier werden sie automatisch ins System eingebunden und den Anwendungen zur Verfügung gestellt. Die internen Abläufe unterscheiden sich allerdings erheblich, worauf später näher eingegangen wird.

1.2. Software Plug&Play

Das Aufbringen neuer Softwarefunktionalitäten, beispielsweise Fähigkeitskomponenten oder Applikationen in Form von Assistenten, ist Kern des Software Plug&Play. Dabei können diese auf einfache und werkzeuggestützte ausgewählt und installiert werden. Diese Art der Systemerweiterung schlägt sich direkt im Anwendungsverhalten nieder und gibt Nutzern die Möglichkeit in die Hand, ihr Heimautomatisierungssystem auf die eigenen Bedürfnisse anzupassen.

1.3. Benötigte Plug&Play Funktionen

Im Szenario der „Interaktion im vernetzten Heim“ werden sowohl die Möglichkeiten der Softwarearchitektur bezüglich des Hardware Plug&Plays als auch des Software Plug&Plays vorgestellt:

1. Sensorikaustausch am Roboter (Plug&Play auf Modulebene)
   Der TurtleBot ist so gestaltet, dass man zwischen zwei Sensoren für die Navigation wählen kann: Asus Kamera und Laserdistanzmesser. Je nachdem, welche Hardwarekomponente angeschlossen ist, reagiert das System entsprechend und nimmt die Konfiguration für die Navigation vor. Sind beide Sensoren angeschlossen, werden nur die Daten vom Laserscanner verwendet, da diesem durch seine weit höhere Datengüte explizit Priorität gegeben wird. Der Austausch während des Betriebs des Roboters wird dargestellt.
2. Sensorik in der Heimumgebung (Plug&Play auf Knoten- und Modulebene)
   Es wird dargestellt, wie der Roboter seine Aufgabenbearbeitung effizienter gestaltet, wenn er Zugriff auf Sensorik und Aktuatorik des Heimumfeldes erhält. Diese Gegenüberstellung funktioniert u.A. durch die Freigabe bzw. Beschränkung des Zugriffs auf die Deckenkameras im Heimumfeld. Werden die Deckenkameras zugreifbar gemacht, werden die dazugehörigen Objekterkennungsroutinen von der Softwarearchitektur aktiviert. Diese kann der Roboter dann für die Objektsuche verwenden ohne erst die so abgesuchten Räume anfahren zu müssen. Dasselbe funktioniert auch für die Kontaktsensorik, die entsprechende Überwachungsfunktionalitäten des Roboters möglich machen. Dabei fährt der Roboter den Bereich an, in dem einer oder mehrere Kontaktsensoren, wie z.B. Sensoren zum Erkennen geschlossener Türen oder Fenster, ein Ereignis ausgelöst haben, und nimmt von dessen Umgebung Bilder auf, die er weiterleitet. Ohne diese Sensorinformationen, müsste der Roboter entweder permanent Bilder aufnehmen oder auf teure Bilderkennungsroutinen zurückgreifen, die den Zustand (offene Tür, offenes Fenster) analysieren können.
3. Assistenten und Fähigkeiten (Plug&Play auf Softwareebene)
   Zusätzlich lassen sich Softwarekomponenten auf die einzelnen Systeme installieren. Auch dies löst Plug-Ereignisse in der Softwarearchitektur aus, sodass die Intelligente Steuerung zeitnah auf die veränderte Systemkonfiguration reagieren kann.

2. Umsetzung der anwendungsspezifischen Systemarchitektur

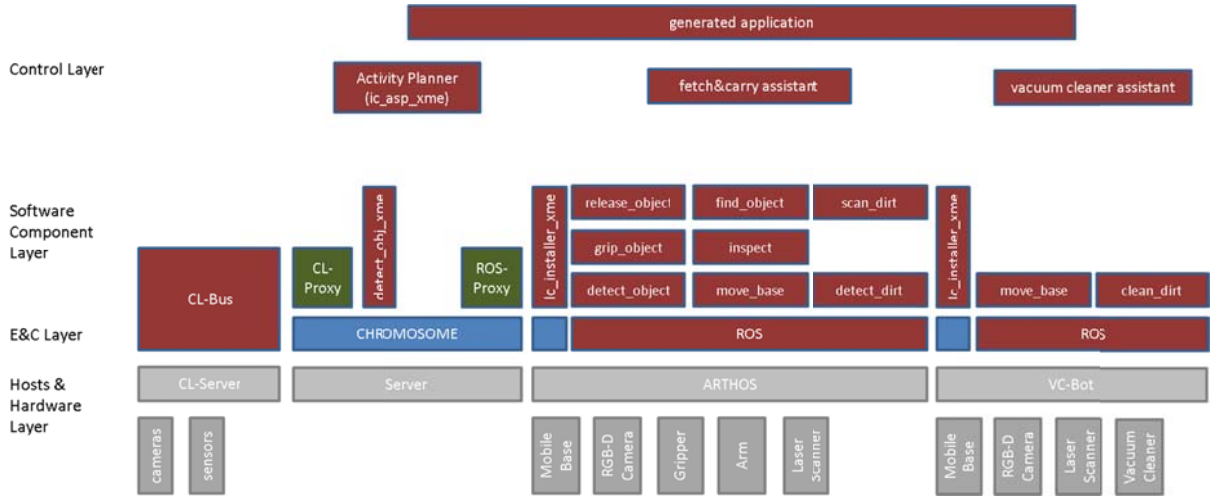

Abbildung 1: Systeme und Komponenten des Interaktionsszenarios

Abbildung 1 veranschaulicht den Aufbau des Szenarios und stellt zudem sämtliche, für die Plug&Play Architektur relevante, Komponenten dar. Aus Gründen der Übersichtlichkeit, werden die Kommunikationspfade später bei der Erläuterung der einzelnen Plug&Play Prozesse dargestellt, da dabei auch Treiberkomponenten mit eine Rolle spielen.

Im Szenario kommen zwei Robotersysteme mit unterschiedlichen Fähigkeiten zum Einsatz womit auch die Interaktion zwischen mobilen Systemen in der Heimumgebung veranschaulicht werden kann. Weiterhin gibt es einen CHROMOSOME Masterknoten auf einem separaten Server, der als Fixpunkt für die Vernetzung benötigt wird. Die vom Anwendungsumfeld „Ambient Assisted Living Testbed“ vorgegebene Infrastruktur des CL-Busses bleibt unangetastet und wird unverändert im Szenario verwendet. Die Verknüpfung der verschiedenen Bussysteme wird über Proxy-Komponenten in CHROMOSOME realisiert. Steuerungs- oder Kontrollkomponenten für Geräte, die über den CL-Bus angesprochen werden, werden direkt als CHROMOSOME-Komponenten realisiert (ein Beispiel ist die Objektdetektion). Weiterhin wird die Handlungsplanung der Intelligenten Steuerung auf CHROMOSOME erweitert, sodass Systemübergreifende (bzw. in diesem Fall ROS-übergreifende) Planung und Planausführung realisiert werden. Die durch die Planungsprozesse generierten Anwendungen werden als CHROMOSOME-Knoten aktiviert.

Es ist hervorzuheben, dass die Roboter im vorgestellten Szenario auf sehr unterschiedlichen Hardwareplattformen basieren. Einerseits wird ein Festo F5 Industrieroboter (siehe Abbildung 2) eingesetzt, der hauptsächlich aus einer mobilen Basis und einem darauf angebrachten Greifsystem besteht. Erweitert wurde er um ein Gestell welches die visuelle Sensorik (Microsoft Kinect) trägt. Auf dieser Grundlage konnten z.B. die Routinen wie Objekt- oder Farberkennung entwickelt und zudem das Greifen von Objekten realisiert werden.

Abbildung 2: Roboter im Interaktionsszenario (links Festo F5, rechts TurtleBot)

Der weitere Roboter, der im Szenario eingesetzt wird, ist der TurtleBot. Dieser verwendet als mobile Basis das „Roomba“-Saugsystem welches über mehrstufige Aufbauten durch einen Rechner, Laserdistanzmesser und visueller Sensorik (Asus Xtion Pro) erweitert wurde. Die komplette Liste der Hardware für dieses Szenario ist wie folgt:

(Teil-)System	Hardware
Festo F5 (ARTHOS)	Mobile Basis (Räder, Motoren, Laserdistanz-messer, Batteriefach + Batterien)
	Mitsubishi RV-1A Arm
	Kinect System + Halterung (Eigenmontage)
	EUROCOM Notebook (mit hoher grafischer Leistung)
TurtleBot (VC-Bot)	„Roomba“ Saugsystem - Mobile Basis
	Laserdistanzmesser Hokuyo URG-04LX-UG01
	Asus Xtion Pro Kamerasystem
	Netbook
	Aufbauten (Eigenmontage nach Anleitung)
CL-Bus	Server
	EnOcean Kontaktsensoren
	Deckenkameras
	(Touch-)Displays
	Blenden
	Schalter und schaltbare Steckdosen

3. Plug&Play Prozesse im Szenario

Wie eingangs erwähnt, gibt es im Szenario drei verschiedene Möglichkeiten die Fähigkeiten der Plug&Play-Architektur aufzuzeigen. Im Folgenden werden die Schritte und Prozesse, die den drei Möglichkeiten zugrunde liegen, detailliert dargestellt.

3.1. Sensoraustausch am Roboter (Modulebene)

Ziel ist es, die Navigation des TurtleBots von kamerabasiert auf laserbasiert umzustellen. Initial ist die Kamera angeschlossen und konfiguriert. Im betrachteten Prozess wird der Laserscanner an den Steuerungsrechner des TurtleBots angeschlossen und die internen Abläufe betrachtet (siehe Abbildung 3).

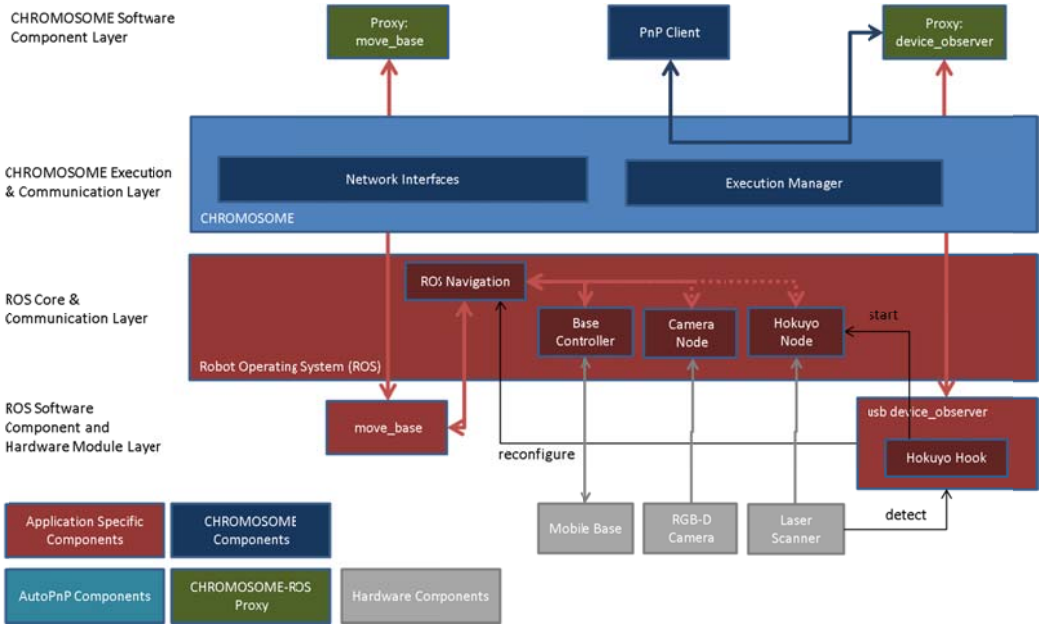

Abbildung 3: Relevante Komponenten für den Sensoraustausch

1. Schritt **Anschluss Laserscanner (via USB)**:
   Der Nutzer steckt das Kabel des Laserscanners an einen der freien USB-Ports des Steuerungsrechner des TurtleBots.
2. Schritt **Detektion durch udev Dämon**:
   Der Anschluss des neuen Geräts wird durch den Betriebssystem-Dämon erkannt. Er liest die vom Gerät gelieferten Informationen (z.B Vendor ID und Product ID) aus. Er schaut weiterhin in der Treiberliste nach dem richtigen Treiber und lässt ihn vom Kernel laden. Der Treiber baut den zur Kamera gehörenden Devicebaum im Systemverzeichnis /dev auf.
3. Schritt **Benachrichtigung Device Observer**:
   Das durch udev erzeugte Ereignis wird im Device Observer empfangen und verarbeitet.

Die Informationen im udev-Ereignis enthalten u.A. den Bus („USB“) sowie Vendor und Product ID. Anhand dieser Informationen wird der relevante Hook zur weiteren Bearbeitung ausgewählt. Im konkreten Fall ist dies der Hokuyo Hook.

4. Schritt **Abarbeitung Hokuyo Hook**:
   Der Hokuyo Hook sucht aus den Deviceinformationen den richtigen Port (im Linux /dev Baum) heraus und konfiguriert damit den Standard ROS *hokuyo_node* und startet diesen. Der Node greift dann in kurzen Zeitabständen die Daten vom Scanner ab und wandelt sie in die für die Navigation benötigten Punktwolken um. Sie werden auf ein neu eingerichtetes ROS-Topic publiziert.
5. Schritt **Rekonfiguration Navigation**:
   Entsprechend der vorgegebenen Prioritäten, dass der laserbasierten Navigation aufgrund der Datengüte Vorrang vor kamerabasierter Navigation gegeben wird, wird die ROS Navigation umkonfiguriert und auf das ROS-Topic des Laserscanners umgestellt.
6. Schritt **Information Plug&Play Client**:
   Der Plug&Play Client wird über die neuen Komponenten (Hardware und Software) informiert und die Komponentenliste entsprechend aktualisiert.

3.2. Sensorik in der Heimumgebung (Knoten- und Modulebene)

Der Zugriff auf die Sensorik in der Heimumgebung ist sowohl auf Modul- als auch auf Knotenebene möglich. In diesem Prozess wird auf die Knotenebene eingegangen, bei der der Zugriff auf den CL-Bus aktiviert wird (Abbildung 4).

1. Schritt **Aktivieren des CL-Bus Knotens**:
   Der Administrator fährt den CHROMOSOME Knoten hoch, der unter anderem die CL-Proxy Komponente mitbringt.
2. bis 8. Schritt siehe *„Entdeckung von Änderungen auf Knotenebene*“ im CHROMOSOME Dokument
9. Schritt **Registrierung CL-ContextModel**:
   Die CL-Proxy Komponente registriert sich über die JSON-Model API mit dem Context-Model um Änderungen bei Geräten mitzubekommen. Dabei wird der Proxy auch über hinzugefügte oder entfernte Geräte informiert. Diese Detektionsinformationen können dann entsprechend an den Plug&Play Client weitergeleitet werden.
10. Schritt **Information Plug&Play Client**:
    Der Plug&Play Client wird über die durch das ContextModel verwalteten Geräte und Sensoren informiert. Diese sind aus Sicht der Plug&Play-Architektur neu hinzugekommen. Die weiteren Schritte folgen dem Verlauf des Plug&Play auf Modulebene.

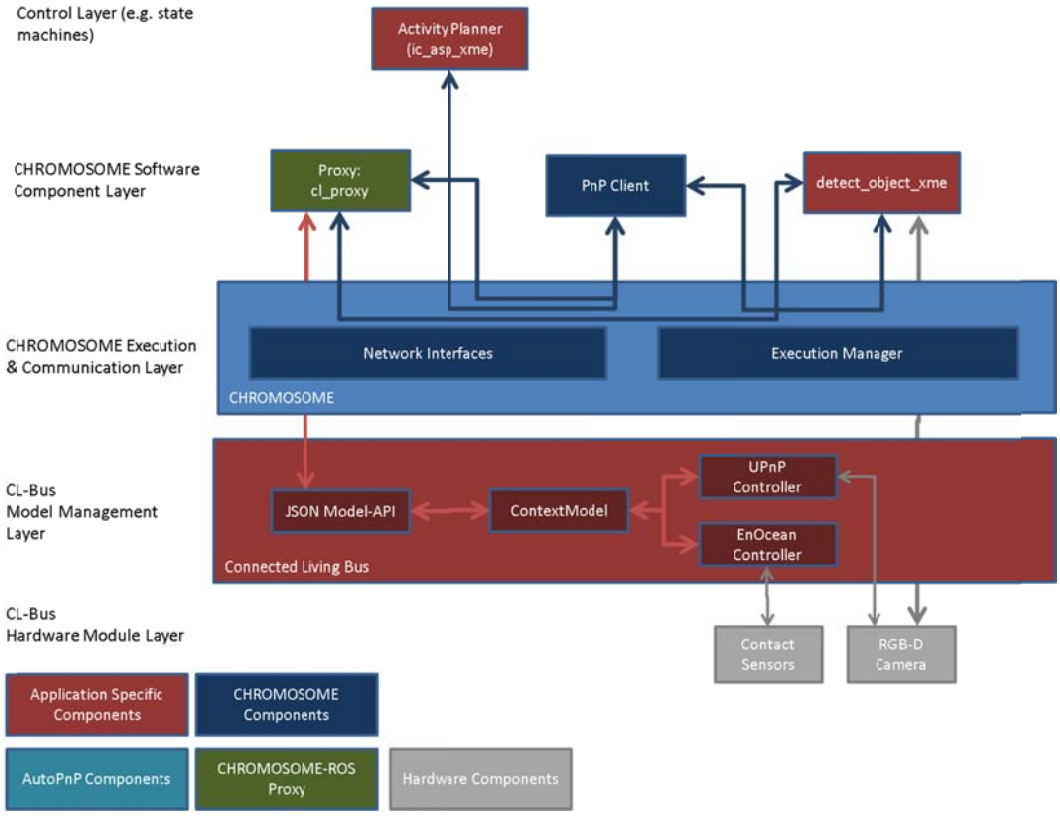

Abbildung 4: Relevante Komponenten für CL-Bus Integration

Ein im Szenario berücksichtigter Gerätetyp sind die Deckenkameras. Wird nun der Plug&Play Client vom Vorhandensein einer (oder mehrerer) Deckenkamera in Kenntnis gesetzt, wird der Prozess angestoßen wie in *„Entdeckung von Änderungen auf Modulebene“* im CHROMOSOME Dokument beschrieben. Die neue Komponente ist hierbei die Objektdetektionskomponente, die direkt auf den HTTP-Datenstrom der assoziierten Deckenkamera zugreifen und verarbeiten kann. Bei einer entsprechenden Zielvorgabe (*„Bring Objekt“*) werden dann diese erweiterten Detektionsmöglichkeit bei der Handlungsplanung mit berücksichtigt.

3.3. Assistenten und Fähigkeiten (Softwareebene)

Dieser Prozess zeigt die Möglichkeit der Architektur auf, zur Laufzeit neue Softwarekomponenten zu installieren und dadurch auch das Verhalten des Systems zu beeinflussen. Es wird auf die Installation einer neuen Fähigkeit *Schmutzkartierung* auf den *ARTHOS* eingegangen, die von einem Administrator oder auch vom Nutzer selbst vorgenommen werden kann (Abbildung 5).

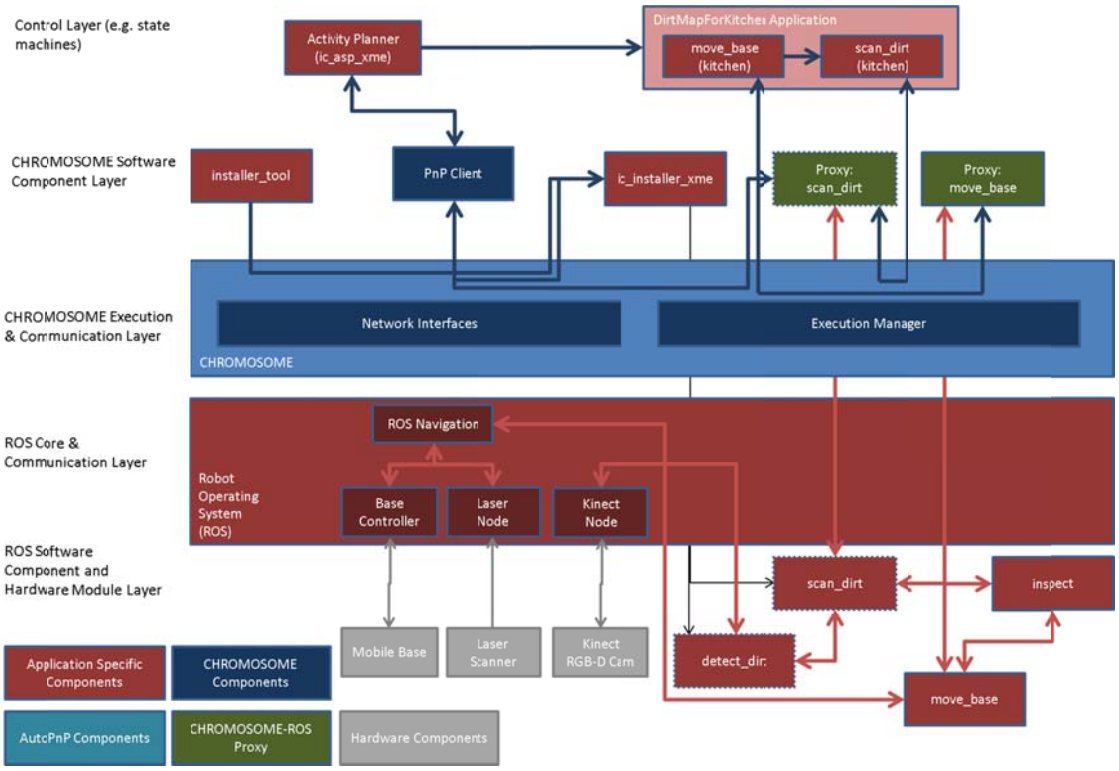

Abbildung 5: Relevante Komponenten des Fähigkeiten Plug&Play

1. Schritt **Auswahl Fähigkeitskomponente**:
   Mittels des Werkzeugs zur Komponenteninstallation, werden die für das Zielsystem relevanten Softwarekomponenten angezeigt. Dabei wird die aktuelle Systemkomposition berücksichtigt, die neben den Informationen zum Betriebssystem und den Bibliotheken auch Informationen zur angeschlossenen Hardware und den aktuell aktivierten Softwarekomponenten beinhaltet. Der Nutzer wählt die Fähigkeit aus (hier *Schmutzkartierung*). Das Vorschlagssystem löst transitiv die Abhängigkeiten auf und erkennt, dass auch die Fähigkeit *Schmutzdetektion* mit installiert werden muss.
2. Schritt **Information Installer**:
   Die Installationskomponente auf dem Zielsystem (*ARTHOS*) wird benachrichtigt und die Informationen zu den zu installierenden Komponenten mitgegeben.
3. Schritt **Installation**:
   Der Installer lädt die Komponenten, sofern sie nicht schon vorhanden sind, aus dem Komponentenrepository herunter und fügt sie dem lokalen Komponentenpool hinzu. Die Komponenten werden durch die mitgelieferten Startbinaries aktiviert und fahren im dargestellten Fall die benötigten ROS-Knoten hoch.
4. Schritt **Information Plug&Play Client**:
   Der Plug&Play Client wird vom Installer über die beiden neuen Softwarekomponenten in Kenntnis gesetzt.
5. Schritt **Information Activity Planner**:
   Der Activity Planner wird über die veränderte Systemkonfiguration in Kenntnis gesetzt und errechnet basierend auf dem aktuellen Ziel und dem Bearbeitungszustand einen

neuen Plan. Sollte dieser vom aktuell verfolgten Plan abweichen, wird diese Planausführung (generierte Applikation) terminiert. Im vorgestellten Fall, gab es zur Zielformulierung *„Erstelle Verschmutzungskarte für Küche“* mangels der benötigten Fähigkeiten noch keinen Plan.

6. Schritt **Bereitstellung Plan**:
   Der neue Plan des Activity Planners (= generierte Anwendung) wird bereitgestellt. Er gibt die Abarbeitungsreihenfolge der einzelnen Fähigkeiten vor.
7. Schritt **Abarbeitung Plan**:
   Die generierte Anwendung wird über CHROMOSOME ausgeführt.