

Illustration der Vorgehensweise

Im den folgenden Sektionen werden die Anpassungen an die FTonto erläutert, die notwendig waren, um die Aufgabenstellung lösen zu können.

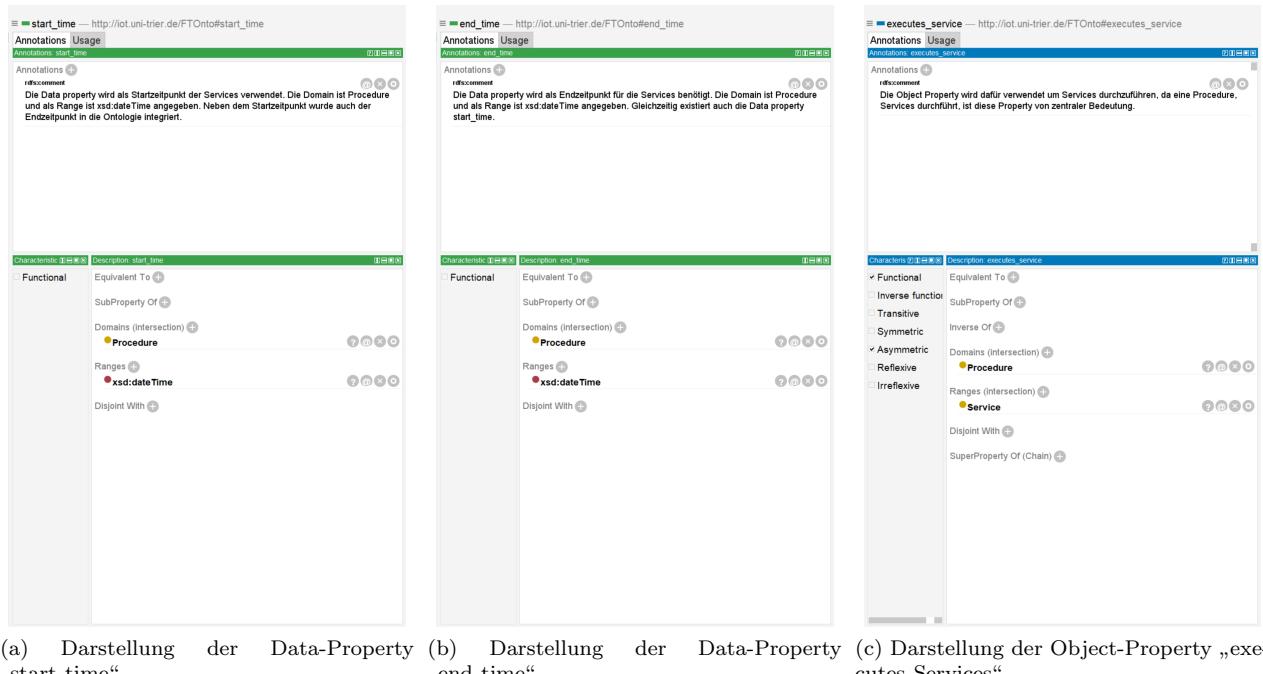
Modellierung der Service Executions

Als erstes wurden die Service Executions in die Ontologie integriert. Die einzelnen Service Executions des Vacuum Gripper Robot (VGR) wurden dabei als Individuals in der Ontologie angelegt. Die Individuals wurden der Klasse „Procedure“ zugeordnet, da diese aus unserer Perspektive, die am besten geeignete Klassifizierung darstellt. Auf die Erstellung einer neuen Klasse wurde verzichtet, um die Ontologie möglichst übersichtlich und verständlich zu gestalten. Die folgende Abbildung (Abbildung 1) zeigt einen Screenshot, auf dem deutlich wird, wie die Service Executions in die Ontologie integriert sind.

Abbildung 1: Modellierung der Service Executions als Individuals der Klasse Procedure

Integrierung weiterer Eigenschaften

Danach wurden die einzelnen Prozesszeiten aus der Tabelle der Aufgabenstellung in die Ontologie eingefügt. Hierfür wurden die Data Properties „start_time“ und „end_time“ erstellt und die Object Property „executes_Services“ angelegt. Die Object Property „executes_Service“ wird, dafür verwendet, den aus der Aufgabenstellung genannten Service auszuführen. Mit den folgenden Screenshots (Abbildung 2) wird die Modellierung der zuvorgenannten Eigenschaften dargestellt.



(a) Darstellung der Data-Property „`start_time`“ (b) Darstellung der Data-Property „`end_time`“ (c) Darstellung der Object-Property „`executes_Services`“

Abbildung 2: Darstellung verschiedener Properties

Ergänzungen an der Übersicht der SOSA-Klassen und -Rollen

Zum besseren Verständnis ist hier, die Abbildung 4 (Overview of SOSA classes and roles) vom Aufgabenblatt aufgeführt. Die Abbildung ist an drei Stellen erweitert worden. Als „Inverse Of“-Beziehung der „used Procedure“ ist in der Ontologie „has_observation“ eingefügt. „has_observation“ wird beim Import der Sensordaten verwendet. Die Eigenschaft dient dazu die Sensordaten mit den entsprechenden Services zu verknüpfen, als Domain wurde „Procedure“ und bei den Ranges „Observation“ ausgewählt. Darüber hinaus wurde „is_result_of“ und „has_result“ integriert. Auf den ersten Blick erscheint diese Maßnahme redundant, allerdings wird hierdurch eine effiziente Modellierung der Ontologie ermöglicht. Durch die bereits vorhandenen „has result-Object properties“ entsteht eine fehlerhafte Zuordnung innerhalb der Ontologie, welche durch die neu erstellten Object properties „has_result“ und „is_result_of“ vermieden wird. Ohne die hinzugefügten Object properties würden die Results der Observations, von den importierten Daten, auch den Services zugeordnet. Die Verwendung eines Unterstrichs im Namen der Object properties ermöglicht eine eindeutige Zuordnung durch den Reasoner. Durch die syntaktische Erweiterung im Namen der Object properties können falsche Assoziationen des Reasoners ausgeschlossen werden, wodurch eine deutlich verbesserte semantische Klarheit erreicht wird. Die Implementierung der Properties ist in der Abbildung (Abbildung 4) dargestellt.

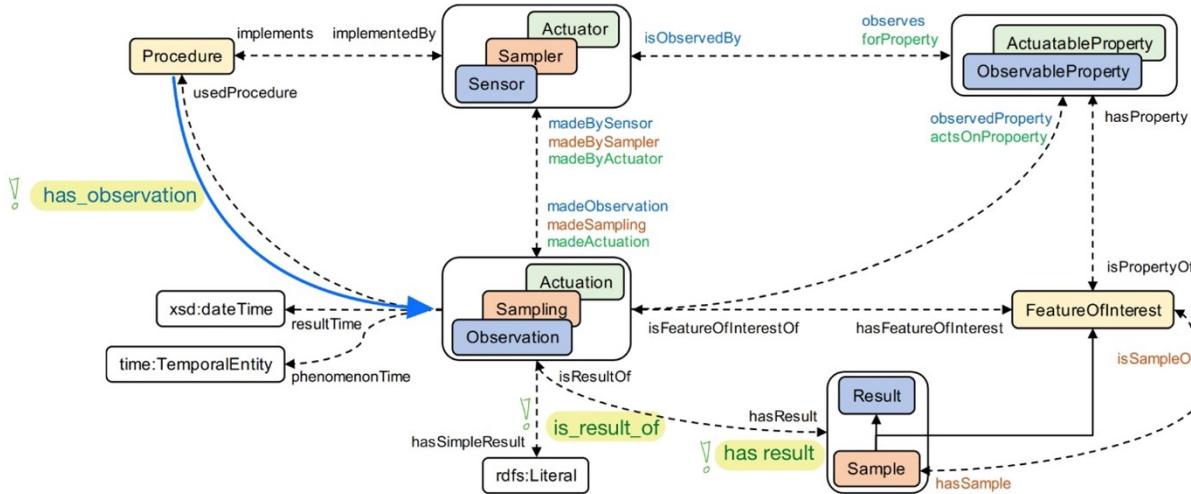


Abbildung 3: Erweiterung der SOSA-Ontologie

has_result — http://iot.uni-trier.de/FTOnto#has_result

Annotations Usage Annotations has_result

Annotations

rdft:comment

Die Object property ist für eine klare und übersichtliche Modellierung innerhalb der Ontologie angelegt worden. Mit den bereits existierenden has result-Eigenschaften gibt es Probleme mit dem Reasoner, da dieser durch die gleichen Ranges Observations falsch zuordnet. So wurden die Results der Observations auch den Services zugeordnet, was aus unserer Perspektive nicht korrekt ist. Durch den Unterstrich im Namen sind die Object Properties klar voneinander getrennt, wodurch der Zuordnungsfehler des Reasoners vermieden werden kann. Die Inverse Of-Eigenschaft lautet **is_result_of**, welche ebenfalls hinzugefügt wurde. Durch die beiden Eigenschaften kommt es zu keiner Verwechslung und die Ergebnisse können den entsprechenden Observations zugeordnet werden. Observation wurde als Domain und Result als range zugewiesen.

is_result_of — http://iot.uni-trier.de/FTOnto#is_result_of

Annotations Usage Annotations is_result_of

Annotations

rdft:comment

Die Eigenschaft ist die Inverse Of-Object property von **has_result**. Diese wurde aus dem gleichen Grund, wie **has_result** hinzugefügt. Damit alle Ergebnisse eindeutig zugeordnet werden können und keine Probleme mit dem Reasoner entstehen.

Characteristics Description **has_result**

Functional	Equivalent To +
Inverse function	SubProperty Of +
Transitive	
Symmetric	
Asymmetric	Inverse Of + is_result_of
Reflexive	Domains (Intersection) + Observation
Irreflexive	Ranges (Intersection) + Result

Characteristics Description **is_result_of**

Functional	Equivalent To +
Inverse function	SubProperty Of +
Transitive	
Symmetric	
Asymmetric	Inverse Of + has_result
Reflexive	Domains (Intersection) + Observation
Irreflexive	Ranges (Intersection) + Result

(a) Darstellung der Object-Property „has_result“
(b) Darstellung der Object-Property „is_result_of“

Abbildung 4: Darstellung weiterer Object-Properties, die notwendig waren um fehlerhaftes Reasoning zu verhindern wegen gleicher Benennung semantisch verschiedener Object-Properties

Zudem werden für Beschleunigungs- und Druckbeobachtungen in die bereits implementierten Klassen „Acceleration Observation“ und „Acceleration Result“ (bzw. „Pressure Observation“ und „Pressure Result“) unterteilt. So enthalten die Individuen der Observation-Klassen unter anderem Eigenschaften über den Sensor von dem diese ermittelt wurden. Die Individuen der Result-Klassen enthalten lediglich die Aufzeichnungen des Sensors. Bei einem Ergebnis der Beschleunigungsbeobachtung sind die Beschleunigungswerte der x-, y- sowie der z-Achse, mit den entsprechenden Data-properties zugeordnet. Dazu mehr im folgenden Abschnitt.

Import der gemessenden Daten am Beispiel der Beschleunigungsdaten

Beim Import der Daten vom „VGR1_BMX055_Acceleration.xlsx“ wurden insgesamt drei Regeln angewendet. Zum einen wird ein Acceleration Result erstellt, welches aus den drei Data Properties „has_x_acceleration“, „has_y_acceleration“ und „has_z_acceleration“ besteht. Dieses Ergebnis wird mithilfe der Object property „has_result“ der „Acceleration Observation“ zugeordnet. Abschließend wird durch „has_observation“ die Beobachtung der entsprechenden Service_Execution zugeordnet. Durch die Kapselung der X-, Y- und Z-Datenwerte in die bereits vorhandene „Acceleration Result“-Klasse wird versucht eine bestmögliche Modellierung der Ontologie zu erreichen. Gleichzeitig wird angestrebt, die Ontologie so präzise wie möglich abzubilden. Durch diese Vorgehensweise existieren in der Onotologie zum Beispiel die „AccObservation10“ und das „AccObservation10_Result“. Die folgende Abbildung (Abbildung 5) zeigt die Regeln, die notwendig sind um die Daten aus der Excel-Datei in die Ontologie zu laden. Mittels dieser Regeln werden die unter anderem die zuvorgeannten Individuen „AccObservation10“ und das „AccObservation10_Result“ erstellt. Die zuvorgeannten Individuen sind in der folgenden Abbildung (Abbildung 6) dargestellt.

Target Ontology: FTOnto (<http://iot.uni-trier.de/FTOnto>)

Workbook (C:\Users\Sascha\Desktop\SemanticTechnology\src\main\resources\A3\E6\VGR1_BMX055_Acceleration.xlsx)

Tabelle1								
	A	B	C	D	E	F	G	H
1	AccObservation	Measurement Point Sensor		AccRes	has x-axis value	has y-axis value	has z-axis value	Time of measurement
2	AccObservation1	VGR_1_Crane_Jib BMX055_Pl_1_AccSensor_1	AccRes1	0.13928993	0.838381142	0.17263394	2024-07-03T11:01:03	Service_Execution_VGR_1_1
3	AccObservation2	VGR_1_Crane_Jib BMX055_Pl_1_AccSensor_1	AccRes2	0.147395285	0.87278817	0.16768685	2024-07-03T11:01:04	Service_Execution_VGR_1_1
4	AccObservation3	VGR_1_Crane_Jib BMX055_Pl_1_AccSensor_1	AccRes3	0.158289929	0.82581643	0.166885551	2024-07-03T11:01:05	Service_Execution_VGR_1_1
5	AccObservation4	VGR_1_Crane_Jib BMX055_Pl_1_AccSensor_1	AccRes4	0.160547497	0.821121154	0.167580188	2024-07-03T11:01:06	Service_Execution_VGR_1_1
6	AccObservation5	VGR_1_Crane_Jib BMX055_Pl_1_AccSensor_1	AccRes5	0.1705477691	0.828022094	0.161856266	2024-07-03T11:01:07	Service_Execution_VGR_1_1

Transformation Rules (C:\Users\Sascha\Desktop\SemanticTechnology\src\main\resources\A3\E6\VGR1_BMX055_Acceleration_rules.json)

Add	Edit	Delete	Load Rules	Save Rules	Save As...	
<input checked="" type="checkbox"/>	Sheet Name	Start Column	End Column	Start Row	End Row	Rule
<input checked="" type="checkbox"/>	Tabelle1	A	I	2	+	Individual: @I* Facts: 'has_observation' @A*
<input checked="" type="checkbox"/>	Tabelle1	A	I	2	+	Individual: @A*(mm:append("_Result")) Types: 'Acceleration Result' Facts: 'has_x_acceleration' @E*(xsd:float), 'has_y_acceleration' @F*(xsd:float), 'has_z_acceleration' @G*(xsd:float) Annotations: rdfs:comment @A*(mm:append("_Result is an acceleration result measured at ",))
<input checked="" type="checkbox"/>	Tabelle1	A	I	2	+	Individual: @A* Types: 'Acceleration Observation' Facts: 'made by sensor' @C*, 'observed property' VGR_1_Crane_Jib_Acceleration, 'used procedure' @I*, 'result time' @H*(xsd:dateTime), 'has_result' @A*(mm:append("_Result")) Annotations: rdfs:comment @A*(mm:append(" is an acceleration observation from the sensor "))

Abbildung 5: Darstellungen der importierten Individuen zu den Daten des Beschleunigungssensors

Weiter wird das Individuum „AccObservation10“ durch die Data Property der entsprechenden Service Execution zugeordnet, in diesem Falls ist das die Service_Execution_1, dargestellt durch das gleichnamige Individuum. Dieser Zusammenhang ist in der Abbildung 7 dargestellt. Zudem ist in dieser Abbildung das zuvor erwähnte Individuum „AccObservation10“ hervorgehoben.

(a) Darstellung des Individuums „AccObservation10“

(b) Darstellung des Individuums „AccObservation10_Result“

Abbildung 6: Darstellung des Zusammenspiels zwischen den aus den EXCEL-Dateien importierten Individuen

Abbildung 7: Darstellungen der Zuordnung der Beobachtungen zu einer Service Execution