

18. **Rachel Hinman**, Emergent Computing Paradigms, 2011, HTML.URL:<http://rosenfeldmedia.com/blogs/the-mobile-frontier/emergent-computing-paradigms/> 12
19. **Weiser Mark D.** , The Computer for the 21st Century - Scientific American Special Issue on Communications, Computers, and Networks, September, 1991 17
20. 2013 edition of Gartner's long-running "Hype Cycle for Emerging Technologies", 2013 14

ОНТОЛОГИЯ НА ПРЕДМЕТНА ОБЛАСТ ЗА РОБОТ LEGO MINDSTORM EV3*

НАЙДЕН В. НЕНКОВ, ЕЛИЦА З. СПАСОВА

ONTOLGY OF SUBJECT AREA FOR ROBOT LEGO MINDSTORM EV3

NAYDEN V. NENKOV, ELITSA Z. SPASOVA

ABSTRACT: *The paper describes creation an ontology for subject area of mobile robot Lego Mindstorms Ev3. The goal is a structural specification the area, terms formalization and logical expressions which described their interaction, and the construction of a specialized vocabulary. It's been used software platform Protégé.*

KEYWORDS: *Robot, Lego Mindstorms EV3, Ontology, Artificial intelligence, Ontology framework, Protégé, Knowledge Base, Robot's Command, Environment.*

УВОД

Докладът описва създаването на онтология за предметна област: програмиране на мобилна конфигурация на робот Lego Mindstorms Ev3 [6]. Целта и е структурна спецификация на класовете и обектите в областта, формализиране на термините и логически изрази, описващи тяхното взаимодействие и изграждане на специализиран речник. Резултатите се използват при разработване на база знания за реализиране на алгоритъм и програма изпълнявани от робота за учебни и изследователски цели. Онтологията е разработена чрез open source платформата Protégé [4, 5].

ОСНОВНА ЧАСТ

Роботът *Lego Mindstorms Ev3* дава възможност за сглобяване на различни конфигурации: мобилни и неподвижни с различен комплект от датчици. За учебни и изследователски цели е удобна мобилната платформа. Хардуерната и софтуерната среда дават възможност за моделиране на сцена и алгоритъм, при който се открива обект и се генерира ответна реакция описана по-долу.

Изградената онтология спомага за ясно дефиниране на понятията при изграждането на един алгоритъм за мобилната реализация на робота, както и за обобщение на връзките между

* Докладът е финансиран от фонд „Научни изследвания“ на Шуменския университет по проект № РД- 10-599/04.04.2014 г.

тях. Онтологията съдържа както по–обща понятия, така и понятия, свързани с конкретната реализация на робота.

Онтологията според IEEE [1, 2] е цялостна интегрирана серия от техники и методи, създаващи обща систематизирана теория как би трябвало да бъде създаден един клас и как би трябвало да работи.

Изграждането на една онтология следва да спазва определена методология [2, 3].

Методологията би следвало да се дефинира като обектно (в идеалния случай количествено определени) множество от критерии за определяне на това дали резултатите от процедурите са с приемливо качество. От друга страна един метод е „подреден процес или процедура” използвана в инженеринга като продукт или представяне на услуга.

Техника е „техническа или управленска процедура, използвана за достигане до даден обект”.

Метод е „функция, която трябва да бъде извършвана в жизнения цикъл на софтуера”. Метода е композиция от дейности.

Една дейност е „задача, която е съставна част на процеса”. Задача е „добре дефинирана работа, задание за един или повече членове от проекта”. Свързаните задачи обикновено са групирани и формират дейности”.

За да се създаде методология за изграждане на онтология трябва да се разгледат следните три типа дейности:

1. Мениджмънт дейности по онтологията
2. Дейности по разработка на онтологията
3. Дейности по поддръжка на онтологията

Процедурите за управление на онтологията (мениджмънт дейностите) трябва да включват дефиниции и график за отделните задачи по изграждане на онтологията. Освен това е необходимо да се определи механизма за контрол и качество, както и етапите на осигуряване. При разработването на онтологията е важно процедурите да са определени в рамките на околната среда и предпроектните проучвания. След решението за изграждане на онтология инженера на онтологията се нуждае от процедури за специализация, концептуализация, формализация и имплементация на онтологията.

Накрая потребителите и инженерите се нуждаят от насоки за поддръжка, използване и развитие на онтологията. За да се подпомогне развитието на онтологията, трябва да бъдат предприети поредица важни поддържащи дейности. Те включват придобиване на знание, оценка, интеграция, сливане, съответствие и управление на конфигурацията.

Важно е да се документират резултатите след всяка дейност. В по–късен етап от развитието на процеса това помага да се проследи защо са били взети някои решения по моделирането на системата. Документирането на резултатите може да бъде улеснено с използването на подходящ инструмент.

В зависимост от методологията документирането на системата може да бъде доста различно. Една методология може да наложи да се документират само резултатите от процеса на инженеринг, докато други дават самия процес на вземане на решения.

Дефиниции на понятия, участващи в онтологията

Понятия, включени в онтологията според Cambridge Dictionary [2, 3]

- **Робот** – Машина, контролирана от компютър, която се използва за извършване на определена човешка дейност автоматично
- **Хардуер** - Физическите или електронни компоненти на компютъра (без инструкциите, които ги управляват)

- **Софтуер** – Инструкциите, които контролират това, което прави един компютър; компютърна програма
- **Операционна система** – Множество от програми, които контролират начина, по който работи компютърната система, по – специално как се използва паметта и как различните програми работят заедно.
- **Компютърна програма** – Множество от инструкции, които карат компютъра да направи определено нещо.
- **Сензор** - Устройство, което се използва за дефиниране на това, че нещо е налично или, че има определени промени в нещо.

Преглед на същите понятия, спрямо нуждите на съответният анализ и нуждите на онтологията:

- **Роботът** е машина с автоматично управление, което може автономно да изпълнява определени задачи, почти винаги с помощта на електронен хардуер и програмирани инструкции. Чрез своя вид и движения роботите често създават усещането, че имат собствени намерения и самостоятелна способност за действие.

Роботите обикновено се използват за да изпълняват задачи, прекалено еднообразни, мръсни или опасни за човека. Индустриалните работи използвани при поточните линии са най-често използваната форма на работи, но скоро бяха заменени от потребителски работи, почистващи пода и косящи трева. Други приложения включват почистване на токсични разливания, изследване на океаните и космоса, хирургия, минно дело, спасителни операции, търсене на пехотни и танкови мини. Роботите си проправят път в развлекателната индустрия и домашното здравеопазване.

Думата робот се използва по отношение на голям набор от машини, общото между които е, че могат да извършват движение и физически операции. Роботите се проявяват в множество форми, вариращи от хуманоид, който имитира човешката форма и начин на придвижване, до индустриални, чиито вид се определя от функцията, която изпълняват.

Роботите могат да се групират на мобилни работи (в това число и автономни превозни средства), манипулаторни работи (индустриални работи), и реконфигурируеми работи, които се приспособяват към задачата, която изпълняват. Роботите могат да се контролират от човек - например управляваните от разстояние унищожавачи бомби работи, роботизирани ръце и др. или да действат в съответствие със заложената в тях способност за вземане на решения, предоставена от изкуствен интелект. Повечето съществуващи работи попадат между тези две крайни категории, като се управляват от програмирани компютри. Такива работи могат да съдържат обратна връзка, за да взаимодействат със средата, но не притежават действителна интелигентност.

Думата робот се използва също така в общ смисъл за всяка машина, която имитира действия на човек във физически или мисловен смисъл. Терминът робот произлиза от чешката дума *robot*, означаваща промишлен труд и за първи път е употребен от чешкия писател Карел Чапек през 1920 г. в пиесата му „R.U.R.“ (Росумски универсални работи). Предполага се, че е измислена от брата на писателя - художникът Джоузеф Чапек.

- **Хардуерът** е вид техника. Изграден е от електрически вериги. Думата е чуждица в българският език. Употребява се главно (но не винаги) в сферата на информационните технологии. Произлиза от английската дума *hardware* (означава железария; техническа апаратура; изчислителна техника; устройства, изграждащи по-голям апарат или машина). Преди появата на тази дума в българския език се е използвал терминът **апаратно осигуряване**. Хардуерът, изграждащ компютъра се състои от процесор, дънна платка, видео

карта, RAM памет, хранващ блок, твърд диск, компютърна кутия и входно-изходни устройства

- **Софтуерът** (на английски: *software* — *програмно осигуряване*) е съвкупността от цялата информация от инструкции и данни, необходими за работата на всяка изчислителна машина. Употребява се главно в сферата на информационните технологии. Обикновено инструкциите се задават като съвкупност от алгоритми, групирани като програми с различно предназначение. Освен самите алгоритми, за изпълнението на програмите са необходими и начални данни. Резултатът от действието на даден алгоритъм може да служи като начални данни за стартирането на друг и т.н., обединявайки програмите в едно. В този смисъл все повече се налага и терминът **софтуер**, който исторически се е наложил като антоним на хардуер - физическата част на компютърните системи. Все пак границата между софтуер и хардуер се размива, когато се отчете, че програмното осигуряване има все пак някакви физически носители, от които изчислителната машина чете програмите. Тези носители, както и инструкциите за работа с програмите и тяхната поддръжка (документация), са спомагателни елементи от софтуера.

- **Операционната система (ОС)** е основна част от компютърния системен софтуер, която управлява и координира работата на процесора и устройствата в компютърната система. Тя обслужва работата на приложния софтуер, като заделя необходимите за това хардуерни ресурси и контролира достъпа на различните приложения до тях.

- **Компютърната програма** или още **софтуерна програма**, или накратко **програма** е съвкупност от инструкции към компютъра, които трябва да дадат някакъв определен изходен резултат. Компютърът изисква програмите да функционират, обикновено изпълнявайки програмните инструкции в централния процесор.^[2] Програмата има изпълнителна форма, която компютърът директно използва, за да изпълни инструкциите, зададени от програмата. Тази същата програма има и четивна за човека част, наречена сорс код, от който изпълнимите програми се компилират, като сорс кодът дава възможност на програмиста да изследва протичането на инструкциите или пък да разработва алгоритмите.

- **Сензор** (на английски: *sensor*) или **датчик** (на руски: *датчик*) е първичен преобразувател на физични или химични параметри в удобен за използване сигнал. Тези устройства представляват неизменна част от системите с автоматизирано управление. Разграничаването на термините сензор и измервателен елемент е проблематично, тъй като не съществуват единни дефиниции.

База знания - Робот Lego Mindstorms EV3

Предметната област на базата знания е робота Lego Mindstorms EV3. Базата знания е изградена, въз основа на конкретна програма, реализирана на робота, за която е използван езика Canvas и средата Lego Mindstorms EV3 Home Edition [6]. Програмата е свързана със засичане на обект от робота и съответна реакция от негова страна - удар с "чук".

Концепти:

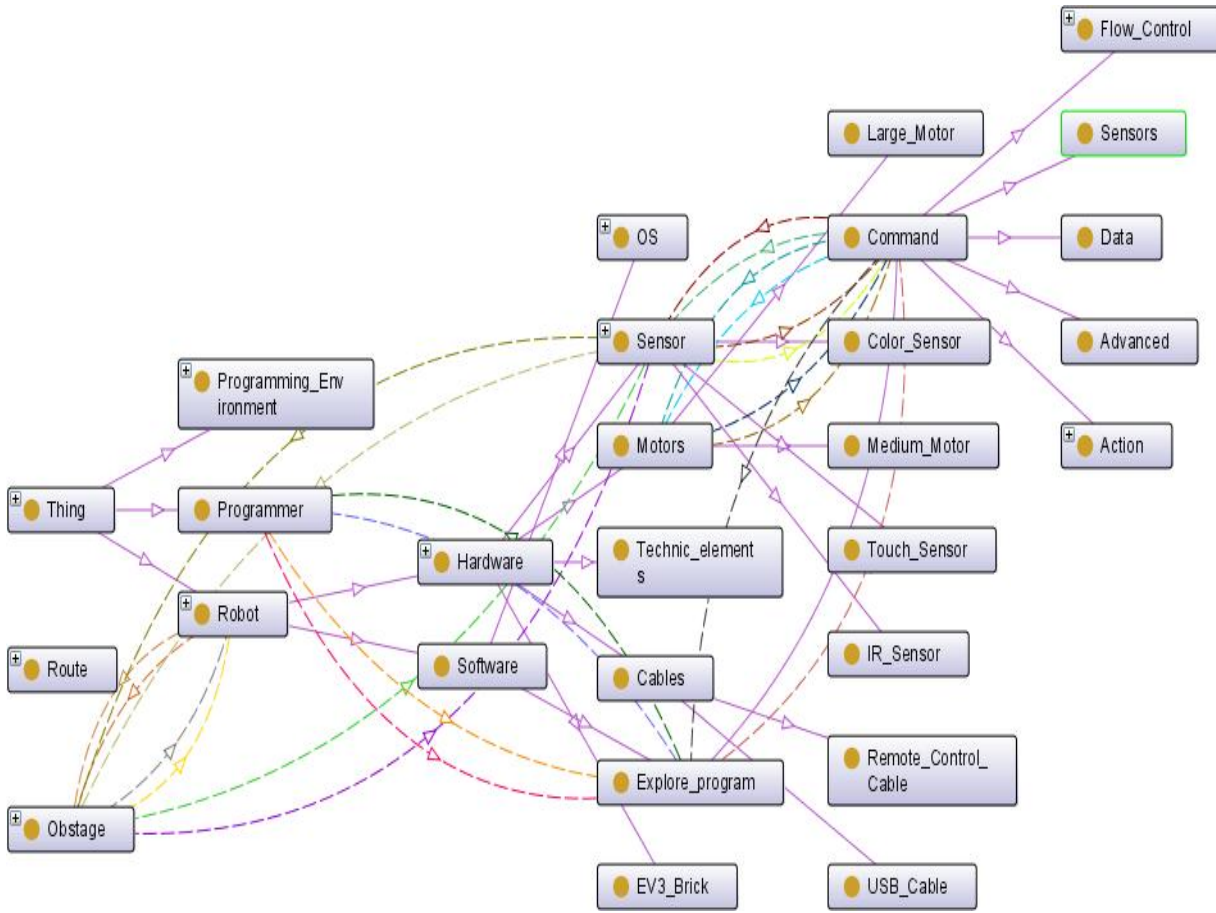
Атомарни и съставни концепти от базата знания

- Programmer (атомарен концепт)
- Programming environment (атомарен концепт)
- Environment - съставен концепт
 - Obstage - Route
- Robot - съставен концепт
 - Hardware - съставен концепт

- ✓ Cables - съставен концепт
 - ❖ Remote Control Cable
 - ❖ USB Cable
- ✓ EV3 Brick
- ✓ Motors - съставен концепт
 - ❖ Large Motor
 - ❖ Medium Motor
- ✓ Sensors - съставен концепт
 - ❖ Color Sensor
 - ❖ IR Sensor
 - ❖ Touch Sensor
- ✓ Technic elements
- Software - съставен концепт
 - ✓ Explore program - съставен концепт
 - ❖ Command - съставен концепт
- Action
- Advanced
- Data
- Flow Control
- Sensors
 - ✓ OS

Няколко примера за описание на класове:

- Command [Subclas Of: Explore Program
activated some Sensor;
power some Motors;
created_by some programmer; (наследено)
moved some Obstage (наследено)]
- Sensor [Subclas Of: Hardware
activated_by some Command
detects some Obstage
moves some Obsage (наследено)]
- Obstage [Subclas Of: Environment
detects_by some Sensor
moved_by some Robot
surrounds some Robot (наследено)]
- Programmer [Subclas of: Thing
creates some Explore Program]



Фиг. 1. Граф на онтологията

Таблица 1. Роли

Domain	Name of Property	Range	Inverse
Command	<i>activated</i>	Sensor	activated_by
Sensor	<i>activated_by</i>	Command	activated
Command	<i>built</i>	Explore program	built_by
Explore program	<i>Built_by</i>	Command	built
Explore program	<i>created by</i>	Programmer	creates
Programmer	<i>creates</i>	Explore program	created_by
Sensor	<i>detects</i>	Obstacle	detects_by
Obstacle	<i>detects_by</i>	Sensor	detects
Robot	<i>moving_in</i>	Environment	surrounds
Obstacle	<i>moving_by</i>	Robot	moves
Robot	<i>moves</i>	Obstacle	Moved_by
Command	<i>power</i>	Motors	powered
Motors	<i>powered</i>	Command	power
Environment	<i>surrounds</i>	Robot	moving_in
	<i>distance</i>	int	
	<i>port</i>	literal	
	<i>Image file</i>	string	
	<i>sound</i>	string	

Константи:

В базата знания са включени 23 екземпляра на различни класове. Тук са включени конкретните команди на изследваната програма, както и конкретната реализация на хардуера на робота Lego Mindstorms EV3. Няколко примера:

- Hammer -> [Types : Hardware
activated_by Command_1
activated_by Command_6.2
activated_by Command_9.1
activated_by Command_10.1
activated_by Command_7.1]
- Wheels -> [Types: Hardware
activated_by Command_9.2
activated_by Command_5.1
activated_by Command_5.2]
- Command1 -> [Type: Action
power Hammer
port Literal]
- Linux 2.6.33 - rc ->[Type: OS]

Извод от вида $d \sqsubseteq e$

Нека $d \doteq$ Command [activated some Sensor

power some Motors
created_by someProgrammer
moves some Obstage]

Нека $e \doteq$ Explore program [created_by Programmer

moves some Obstage]

Да покажем, че втория се съдържа сруктурно в първия

- И в двата концепта се съдържа - created by
 - И в двата концепта се съдържа - moves
- Това е достатъчно.

Извод от вида KB $\models (c \rightarrow e)$

Нека c е Sensor, а e - Sensors

Sensor \rightarrow [activated_by Command_3
detects Book]

Sensors [activated_by some Command
detects some Obstage]

Проверяваме, дали концептите от първия списък покриват тези от втория:

- В първия имаме [activated_by Command_3], а във втория имаме [activated_by some Comand]
 - В първия имаме [detects Book], а във втория имаме [detects some Obstage]
- Това е достатъчно.

Пример за класификация:

Нека се опитаме да класифицираме концепта *SurroundingObjects*

SurroundingObjects [surrounds some Robot
detects_by some Sensor]

При класификация се определят S и G –най-специфичните класове, които съдържат класифицирания клас и най-общите класове, които се съдържат в класифицирания клас. В този случай резултатът ще бъде:

- S - Environment
- G - Obstage

Така в извлечената йерархия класът SurroundingObjects правилно ще бъде поставен като дете на Environment и като родител на Obstage.

ЗАКЛЮЧЕНИЕ

Разработената онтология позволява експлицитно да се определят понятията, определенията, атрибутите, аксиомите и правилата за извод в предметната област: програмиране на робота Lego Mindstorms EV3. Създава се възможност за организирането им в таксономия и йерархията свързана със сценариите на действия, извършвани от робота. Това позволява да се използва онтологията при програмиране на описаните действия чрез Canvas и други софтуерни среди. Методологията за разработване на онтология може да се използва за учебни и научни цели и в други предметни области.

ЛИТЕРАТУРА

1. Gruber, Thomas R. (June 1993). "A translation approach to portable ontology specifications" (PDF). Knowledge Acquisition 5 (2): 199–220.doi:10.1006/knac.1993.1008.
2. Matthew Horridge, Sebastian Brandt. A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools, Edition 1.3. University of Manchester, 2011. Available at http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf.
3. <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>
4. <http://protege.stanford.edu/products.php>
5. http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html
6. <http://www.lego.com/en-us/mindstorms/products/ev3/31313-mindstorms-ev3/>