INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Context-Aware System Applied in Industrial Assembly Environment

Regular Paper

Tomislav Stipancic[1,*], Bojan Jerbic[1] and Petar Curkovic[1]

1 Department of Robotics and Production System Automation, Faculty of Mechanical Engineering and Naval Architecture, Zagreb, Croatia
* Corresponding author E-mail: tomislav.stipancic@fsb.hr

Received 25 Jun 2012; Accepted 10 Sep 2012

DOI: 10.5772/53235

Abstract The objective of this paper is to present an ongoing development of a context-aware system used within industrial environments. The core of the system is so-called Cognitive Model for Robot Group Control. This model is based on well-known concepts of Ubiquitous Computing, and is used to control robot behaviours in specially designed industrial environments. By using sensors integrated within the environment, the system is able to track and analyse changes, and update its informational buffer appropriately. Based on freshly collected information, the Model is able to provide a transformation of high-level contextual information to lower-level information that is much more suitable and understandable for technical systems. The Model uses semantically defined knowledge to define domain of interest, and Bayesian Network reasoning to deal with the uncertain events and ambiguity scenarios that characterize our naturally unstructured world.

Keywords Context-aware, Ubiquitous Computing, Ontology, Probabilistic Robotics, Design

1. Introduction

In an automatic assembly, the control of a system is usually connected to the control of the working environment. An uncontrolled situation is any situation where any object or subject is not completely defined in terms of position, orientation, action and/or process. Every environment is naturally unstructured, which can be revealed if it is observed on a sufficiently sensitive scale. In other words, if we neglect a sub-molecular level, it is not possible to completely determine any environment, no matter how tight the applied tolerance ranges may be. This is connected with issues of sensitivity and instability and may result in malfunctioning, even if only small environmental changes occur. Systems therefore have to be programmed for a limited range of actions foreseen in advance by the system developer. By default, these systems cannot act in any unpredicted situations.

The transition from free to controlled/known spatial state is one of the most demanding tasks in automatic processes. It is very hard to model a complex system to predict all the possible outcomes that the environment might produce, due to its nondeterministic nature. A programmer who tries to control the system has to cope with many certain and uncertain situations. A system designed in such a manner can be called reactive, because it reacts only based on environmental stimulus. The reactive system can be very fragile if something unexpected occurs because it usually does not have any self-recovery capabilities that would, by default, prevent errors arising from unexpected situations. On the other hand, a system that is able to partially recognize context can act depending on contextual information. Such a system can potentially both comprehend the present and predict results of its future actions. Contextual perception implies understanding of a problem domain much broader than a single agent could attain, and it can be accomplished through interaction between the agent and the environment, together with related objects, other agents, processes and events. Dynamic information control based on the contextual perception needs less predetermined operational and structural knowledge. That requires a certain system adaptation skill and some level of decision-making capability to be applied.

2. Cognitive Model

The context of space and time becomes an important task in autonomous system development. Each object, process or condition is unique by its very nature. If an agent is able to make decisions about actions that are not completely explained in the working space, using certain perceptions, knowledge and/or intelligence, we can say that the agent possess a certain level of control [1]. In order to make an environment controlled, the corresponding knowledge about all relevant components and processes should be available. For this reason, the research into new methodologies and paradigms is directed toward the development of adaptive, anthropometric and cognitive agent capabilities [2].

Fig. 1 shows the Cognitive Model for Robot Group Control [3], which can be integrated within a context-aware system to enable a partial perception of the environment. As can be concluded, the Model tends to be used in industrial environments for robot assembly operations. The Model includes the following components: information-gathering by means of sensors integrated into the environment, domain definition through ontology, and Bayesian Network (BN) reasoning mechanisms used to provide a single solution. The synergy of all model components can ensure an adaptive behaviour of the robot with regard to both current environmental conditions and predefined knowledge about a particular domain of interest.

Another challenge was to develop collaborative robot group-work applied in real life scenarios. By means of probabilities, it is possible to give to an agent the capability to behave in seemingly non-predictable scenarios.

First of all, the agent (robot) has to observe its environment to identify the most appropriate actions with respect to the current conditions. These actions are called Behavioural Patterns (BPs), and they are generated by the ontology.

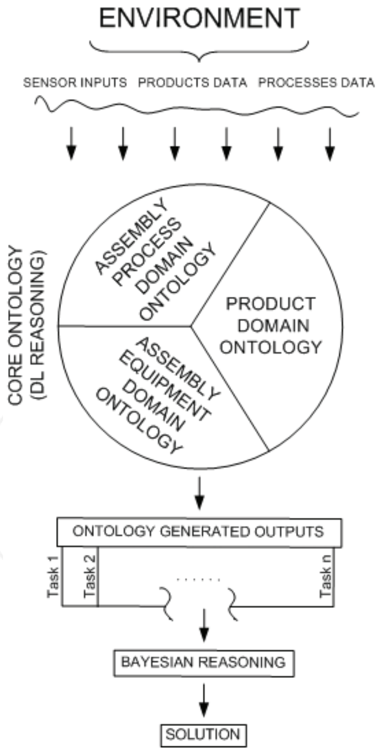


Figure 1. Cognitive Model for Robot Group Control

The last part of the Model uses predefined Bayesian Network (BN) to ensure unambiguous system decisions in response to given environmental conditions.

By utilizing the proposed model, a robot that belongs to the group would become capable of converting its ordinary environment to a ubiquitous one.

2.1 Environment

How and when to collect data, and which data to collect, are important questions for all further steps leading to contextual perception. The answers depend mainly on application goals and require a thorough analysis to be made. This analysis should reveal spatial, temporal and mutual dependencies and characteristics of processes, equipment and objects. Information obtained by analysis could be used during a process of development of so-called smart environments. The term “industrial environmental design” implies determination of all relevant system components, such as sensors, executive units and product carriers.

Due to requirements such as the high degree of automation and large series of products, contemporary robotic assembly systems cannot afford mistakes arising from the non-deterministic nature of the environment. Therefore, such industrial environments usually have to be determined in advance as far as is possible.

In robotic/automatic assembly, information about the position and orientation of work pieces and all other relevant in-process objects and processes represents essential data for control mechanism development and utilization. The change in information is a consequence of different environmental conditions. This can be called expected informational change, and it can be relatively easy to control. With regard to these facts, the Model described in this paper is used to control a robot's behaviour within the industrial environment modelled according to the principles of Ubiquitous Computing. Appropriate sensors are utilized to provide a constant flow of timely fresh information.

The environment is divided into two main parts: an Assembly Line (AL) and Working Places (WPs). The Assembly Line contains Part Carriers (PCs) and can contain some additional sensors according to the current assembly application. A single Working Place (Fig. 2) includes a Conveyer (C), a Pallet (P), a Stopping Place (SP), a Tool Changer (TC1 or TC2), a Robot (R), a Vision Sensor placed near the Stopping Place (VS1) and a Vision Sensor placed on a robotic arm (VS2).

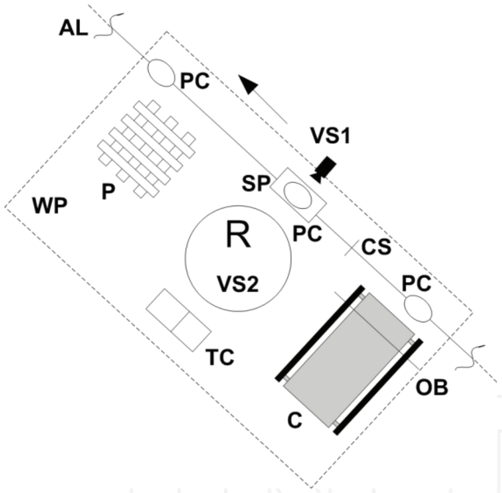


Figure 2. Single Working Place

The Assembly Line is used to transport Part Carriers between Working Places. The Stopping Place is used to retain or release the Part Carrier to the next Stopping Place. While the Part Carrier is located at the Stopping Place, the Robot uses ontology and Bayesian Network to decide about its further actions.

Ontology uses information from all sensors at the current Working Place to suggest the Robot a set of possible actions, called Behavioural Pattern (BP). If ontology suggests more than one Behavioural Pattern, theRobot will use the Bayesian Network to determine the correct unambiguous choice taking into account the conditions and equipment availability at other Working Places. For this application, the Robot can use two types of grippers for two types of parts. Combining the parts, the system can produce two types of products. The Working Place has two vision sensors. One (VS1) is placed above the Stopping Place and is used to determine if the Part Carrier is empty or not. The other (VS2) is placed at the Robot's arm and determines the type, position and orientation of the part at the Conveyor.

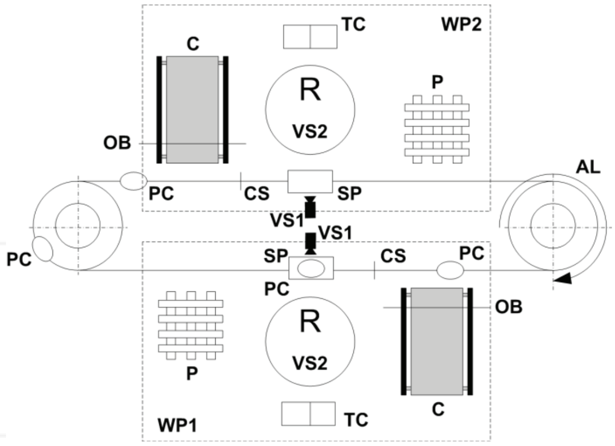


Figure 3. Two-robot assembly system configuration

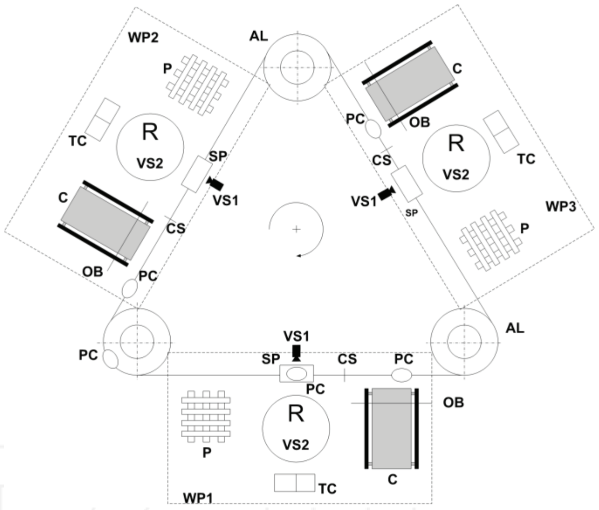


Figure 4. Three-robot assembly system configuration

The Conveyor is used to feed the system with new parts. When the part reaches the Optical Barrier, the Conveyor stops and the Robot can check the Optical Barrier to locate and identify this part by using the Vision Sensor (VS2). The Working Place contains the Pallet to store previously assembled products.

The whole assembly system is designed to follow principles of scalability, auto-recovery and context awareness. Figs. 3 and 4 depict different assembly system configurations. As can be seen, adding new assembly places can demonstrate a principle of scalability resulting in an increase in the overall productivity of the system. The principle of auto-recovery can be realized if one of the Working Places fails in performing its primary functions due to a defect or some other factor. By using the Bayesian Network, other Working Places can rearrange priorities and continue to work. The third principle can be realized by using sensors while collecting information. Sensors are placed to provide all relevant and timely synchronized information about spatial and temporal environmental conditions. By using this principle, a domain of interest becomes a space constantly analysed, just as it should be according to the definition of context-awareness.

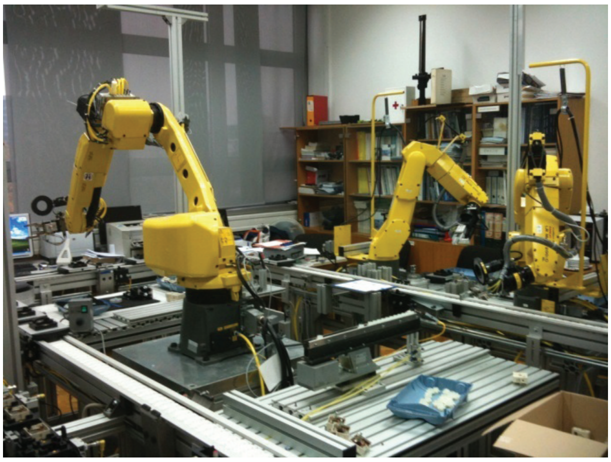


Figure 5. The environment used for application development

Fig. 5 shows the environment used for application development and testing. The environment shown is a part of the assembly system planning laboratory at the Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Croatia.

2.2 Ontology

Ontologies denote a formal representation of entities (classes) along with associated attributes (objects) and their mutual relations [4]. Since ontologies allow representation of an arbitrary domain and can simplify work for end users, they can be extremely convenient. With time, ontologies have also been proven to have certain disadvantages. By designing an arbitrary domain from his own field of expertise, an expert uses personal knowledge and impressions, which can differ when compared with other experts in the field. Such an approach can prevent other ontologies representing the same domain from exchanging information mutually. This means that gaps, overlaps, and inconsistencies will continue to exist when independently developed ontologies are used together [5].Projects MASON [6] and ONTOMAS [7] represent an endeavour by researchers to build up ontology for describing the field of production activities. Some authors are using the same ontologies while developing their own ontology-based applications [8]. Althoughit would have been possible to use parts of existing ontologies while developing the Core Ontology,this approach has proven to be insufficiently precise in cases of robot assembly operations. In order to retain compatibility with other similar ontologies, the Core Ontology design process was guided by a concern about possible future integration with other similar ontologies. Therefore, the ontology developed through the ONTOMAS project and the Core Ontology uses the same Integrated Assembly Model for domain knowledge description, originally proposed by Rampersad [9]. Fig. 6 shows the taxonomy of assembly operations as a part of the Core Ontology.

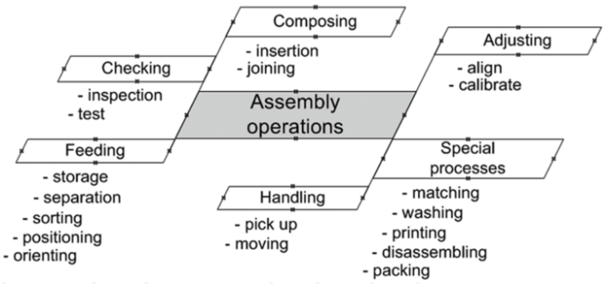


Figure 6. The assembly operations taxonomy

The other identified disadvantage is the fact that ontologies lack support when describing the states that are not naturally uniquely defined. In the world of the non-deterministic, deterministic ontological description is often not enough in itself. In efforts to neutralize these shortcomings, some researchers turned to the development of methods and applications based on probabilities. An approach based on Bayesian Networks (BN) seems to be an interesting solution, because BN can describe complex joint distributions by using sets of local distributions [10]. Larik and Haider [11] provide insight into the research and examples related to the integration of Bayesian Networks and ontologies. As can be seen, the authors have identified four recent directions of research: developing Bayesian Networks from ontology [12, 13], usage of Bayesian Networks for connecting different ontologies [14, 15], probabilistic extension of OWL (Ontology Web Language) [16, 17], and improvement of ontological reasoning mechanisms by means of Bayesian Networks [18]. The application described within this paper can be connected mostly with the fourth identified area of research.

To create the Core Ontology of the Cognitive Model, the knowledge meta process (Fig. 7), described in [19], is used. Before the process of ontology design starts, it is good practice to carry out thorough preparation. This preparation includes complete analysis of the domain of interest as well as all relevant processes and objectives that the ontology should represent. This can reveal possible weak points or misjudgements, and simplify the overall process. After the process of describing the ontology components is finished (definition of classes and related objects), the next step is to define relations among objects and classes. When the preparation stage ends, an ontology designer can define appropriate technology and start to design relevant classes, objects and corresponding relations. As can be seen from Fig. 7, from time to time it is necessary to return to ontology development and make an evaluation to identify and refine the created ontology.

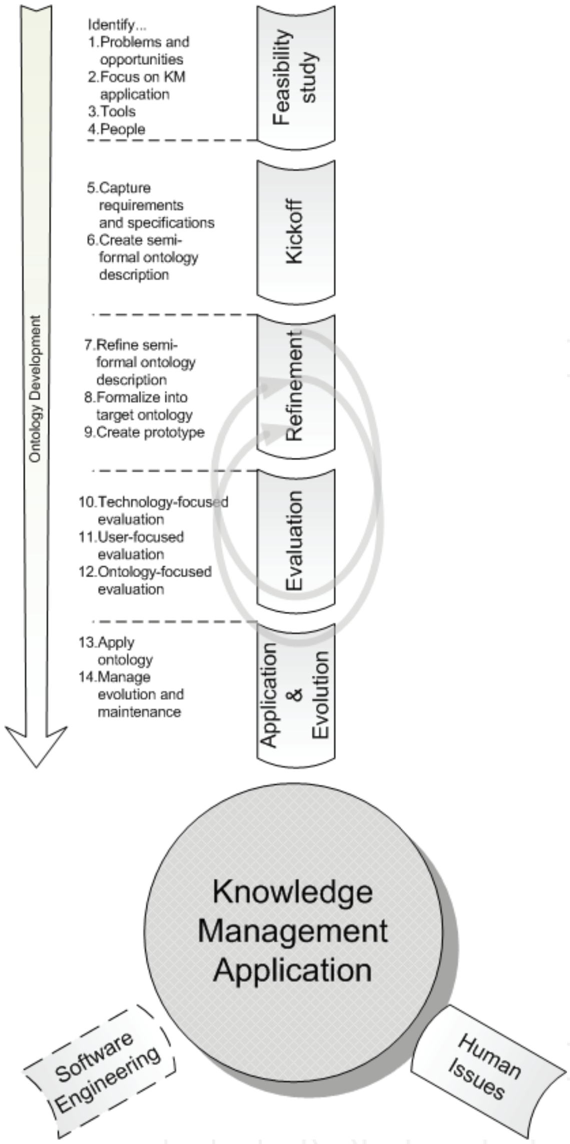
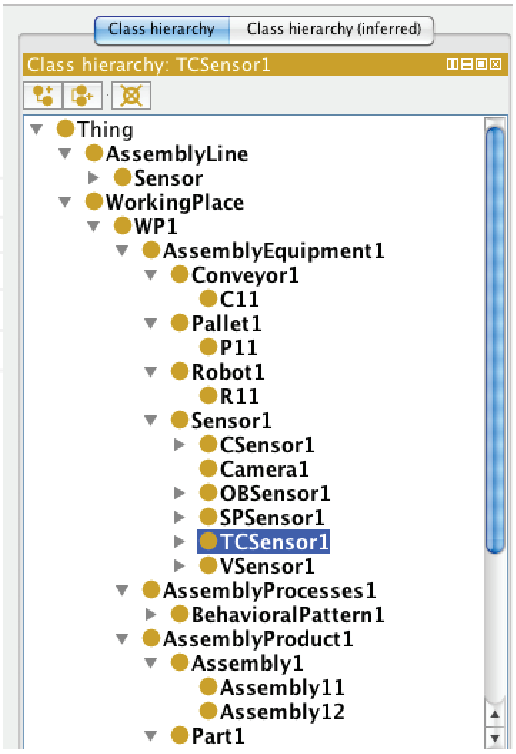

Ontology Development
Identify...
1.Problems and opportunities
2.Focus on KM application
3.Tools
4.People
Feasibility study
5.Capture requirements and specifications
6.Create semi-formal ontology description
Kickoff
7.Refine semi-formal ontology description
8.Formalize into target ontology
9.Create prototype
Refinement
10.Technology-focused evaluation
11.User-focused evaluation
12.Ontology-focused evaluation
Evaluation
13.Apply ontology
14.Manage evolution and maintenance
Application & Evolution
Knowledge Management Application
Software Engineering
Human Issues

Figure 7. The knowledge meta process

While building the Core Ontology, Protégé-OWL editor [20] is used. All classes, objects and corresponding relations are defined with the above-mentioned industrial environment and robot assembly system configurations in mind. Fig. 8 depicts the Protégé-OWL classes description view window, including some of the Core Ontology classes. As can be seen from Fig. 8, this part of the Model has two main classes: Assembly Line and Working Place. According to the proposed Cognitive Model, the class Working Place is divided into three main classes: Assembly Equipment, Assembly Operations and Assembly Product. The Assembly Equipment class contains the subclass Sensor,with all the sensors that are located at the Working Place. The Assembly Product class contains classes that describe all constituent parts: Part, Assembly and Product.

When it needs to react to a certain environmental stimulus, the Robot has to use so-called Behavioural Pattern (BP). BP is a program structure containing a set of elementary operations understandable to the Robot. For the purposes of this application, five Behavioural Patterns are developed, meaning that the Robot can make five decisions about its actions according to current environmental needs.



Class hierarchy: TCSensor1

```
▼ Thing
  ▼ AssemblyLine
    ▶ Sensor
  ▼ WorkingPlace
    ▼ WP1
      ▼ AssemblyEquipment1
        ▼ Conveyor1
            C11
        ▼ Pallet1
            P11
        ▼ Robot1
            R11
        ▼ Sensor1
          ▶ CSensor1
            Camera1
          ▶ OBSensor1
          ▶ SPSensor1
          ▶ TCSensor1
          ▶ VSensor1
      ▼ AssemblyProcesses1
        ▶ BehavioralPattern1
      ▼ AssemblyProduct1
        ▼ Assembly1
            Assembly11
            Assembly12
        ▼ Part1
```

Figure 8. The classes description view

These decisions include: retain or release the Part Carrier at the Stopping Place, wait for constant number of seconds, pick the proper part from Conveyor, and assemble the Product. Behavioural Patterns are stored in robots and they are written according to a robot's programming language. A corresponding class for BPs is placed under the Assembly Processes Class. An example of analgorithm that defines Behavioural Pattern 3 is shown in Fig. 9. The first part of the algorithm is the same as for the other algorithms, and represents current sensor states. This BP is used when the Part Carrier transports the assembled product. By executing this algorithm, the Robot will pick up the assembled product and place it onto the Pallet. This action is used to unload the Part Carrier. Similar Behavioural Patterns are predefined for all other robot actions, and ontology can suggest the right one(s) to the Robot with respect tothe current system needs. Generally, context-aware applications should enable a transition from high-level implicit contextual information to low-level explicit information. Explicit information is much more suitable for agents (robots, devices, etc.) because, by default, they do not share contextual knowledge like living beings [21]. To make this transition possible, ontology uses reasoning based on Description Logic. By containing all relations, objects, and classes, a well-defined ontology can represent knowledge that can be used for building expert systems for various applications. Within semantic web tools, DL-based reasoning is usually expressed using programming languages like OWL-DL [22]. OWL-DL is one of the OWL (Ontology Web Language) dialects that support knowledge sharing and reuse.

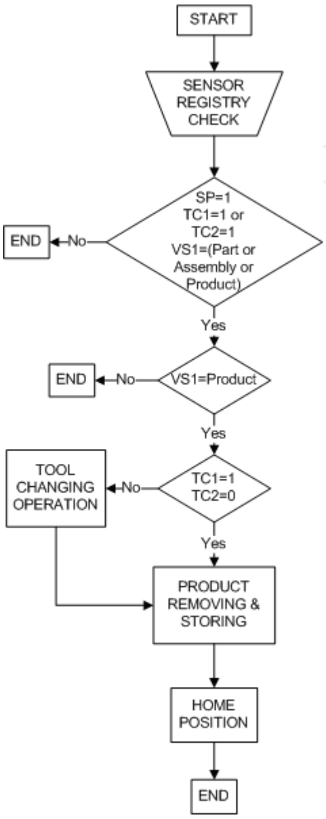


Figure 9. The algorithm of Behavioural Pattern 3

Table 1 shows how the logic is defined within the Core Ontology. The table contains 20statements that represent different sensor conditions, along with appropriate suggestions for the future robot actions (BPs). The logic stored within this table is used to control the robot behaviours at one Working Place. An agent, or let us say a robot, would start the program (BP) depending on the current contextual information derived from interactions between components ofthe observed Working Place. As can be seen in the table, almost all sensors use Boolean data (false or true, 0 or 1) to express changes detected within the environment. Only the Vision Sensor (VS1) uses a string of characters to identify the type of detected object. When analysing the table, an important observation should be made: in some cases it is possible to have more than one possible solution arising from the given state of sensors. This is possible because ontology by its nature follows the principles of Open World Assumptions (OWA). This is the reason why the table sometimes suggests more than one BP.

	TC2	TC1	CS	OB	VS1	BP
1	0	1	0	0	0	BP1, BP2
2	1	0	0	0	0	BP1, BP2
3	0	1	0	0	Part, Assembly	BP1, BP2
4	1	0	0	0	Part, Assembly	BP1, BP2
5	0	1	0	0	Product	BP3
6	1	0	0	0	Product	BP3
7	0	1	0	1	0	BP4
8	1	0	0	1	0	BP4
9	0	1	0	1	Part, Assembly	BP5
10	1	0	0	1	Part, Assembly	BP5
11	0	1	0	1	Product	BP3
12	1	0	0	1	Product	BP3
13	0	1	1	0	0	BP1, BP2, BP3
14	1	0	1	0	0	BP1, BP2, BP3
15	0	1	1	1	0	BP2, BP4
16	1	0	1	1	0	BP2, BP4
17	0	1	1	1	Part, Assembly	BP2, BP5
18	1	0	1	1	Part, Assembly	BP2, BP5
19	0	1	1	1	Product	BP2, BP3
20	1	0	1	1	Product	BP2, BP3

Table 1. The logic implemented within the Core Ontology

In such cases the Robot will become unsure about what to do, because each of the suggested BPs is equally important. To cope with ambiguities, the Cognitive Model uses an approach based on the BN. According to previously stored experience and current environmental conditions (one agent takes into consideration parameters such as sensor conditions or equipment availability, which resides on the group level), the Robot estimates the most appropriate BP.

2.3 The Bayesian Reasoning

Probabilistic-based methods are used in robotics in cases where uncertainties or system hesitations occur. Mostly, these are problems connected with robot planning and control, and problems of mobile robot localization [23]. Within the Cognitive Model, the hesitation is identified while the Robot in the group is planning its next actions. In accordance with Table 1, in some cases the Robot can receive a choice of Behavioural Patterns to be executed. While building the BN for the Cognitive Model, a hand-crafted approach is used that includes expert knowledge. This approach is usually time-consuming [24], and can be used to build small BNs. Information about conditional probabilities for each robot has to be defined in advance. By altering this information, the system designer has the opportunity to define system priorities and/or to achieve certain production goals.

Generally, "knowledge engineering" means converting expert knowledge into a computer model. While building the model by means of BNs, the first step is to decide what the variables of interest are. These variables represent nodes in the BN. Arcs connect the nodes with the direction, indicating a causal relationship. Relationships between connected nodes are quantified by CPTs (Condition Probability Tables), where each entry shows the probability of the child node having a particular value, given a combination of values of the parent nodes. Given the specification of the BN, we can compute posterior probability distributions for each of the nodes, so-called "beliefs".

In order to determine the right Behavioural Pattern, the BN takes into consideration particular information taken from the environment. This information should help the robot to reallocate resources according to the current availability. These kinds of decisions do not affect the quality of robot operation performances but rather the optimal group work in general. The methodology about how the similar cognitive models applied in mobile robots can be developed is described in the examples in [25]. Although the methodology is not directly applicable for industrial robot applications, these guidelines are considered when creating the BN for the purpose of this work.

Each Behavioural Pattern uses information collected by particular sensors while performing its actions. The same sensors are used for reasoning within the BN. For example, BP1 is used to set the Robot to wait for a constant number of seconds. The opposite Behavioural Pattern, BP2,is used to release the Part Carrier to the next Stopping Place. According to Table 1, these two can be suggested as solutions to the current environmental state. They both use CS and SP sensors to make reactive decisions. Therefore, the same pair of sensors belonging to the next Stopping Place is used to make a choice about the appropriate BP. The same principle is used to determine variables for other BNs.

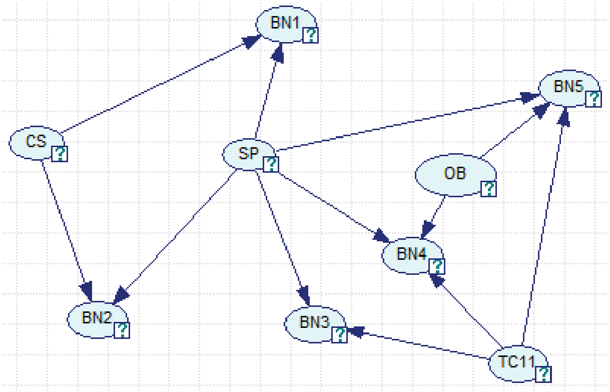


Figure 10. The Bayesian Network used to plan robot actions

Fig. 10 shows a BN developed to plan robot actions. As can be seen, so far there are five Behavioural Patterns and each of them has its own so-called parent node. These parent nodes are sensors used to determine states of particular sensors and availability of the equipment. When the Core Ontology suggests solution(s), for example BP4 and BP5, the BN should set all other BPs to zero (these are evidences) and update values (beliefs) of other nodes. The BP with higher probability value will be suggested to the Robot as a final solution.

While building the Bayesian Network, Genie & Smile [26] are used. Smile is a C++ library developed to build Bayesian Networks. Genie is an application that is based on Smile library, which provides efficient Graphical User Interface (GUI) for end users.

4. Application Implementation

The application is currently in an implementation phase. While the Model is developed, an environment is simulated locally using a dedicated PCServer. After the successful implementation, the whole set-up will be tested in the real environment with real robots applied on industrial assembly assignments (Fig. 5).

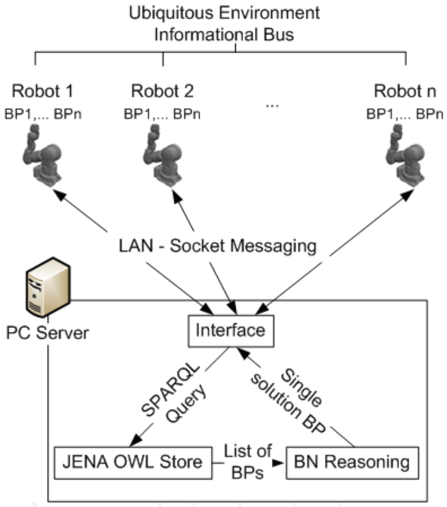


Figure 11. The Software Architecture

Fig. 11 shows the Software Architecture. The main part of the Architecture is the PC Server used as a host for the Cognitive Model. The group of robots and PC Server are mutually connected over the Local Area Network (LAN). They communicate to each other by exchanging messages using Socket Messaging network technology [27]. As can be seen, the Model has an interface used for this type of communication. The message that the Model receives contains an array of data representing information significant for the application (e.g. status of equipment, status of sensors, etc.). They can be found in a so-called Informational Bus. Each agent (a robot belonging to the group, an assembly line controller, etc.) has a data register used to collect information from its own sensors.

Since agents are mutually connected, the Informational Bus denotes a virtual space where all agents can share and exchange significant environmental information. Based on provided information, the Core Ontology can suggest an appropriate list of BPs.

The Core Ontology is stored using the Jena Application [28]. Jena is an open source Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS, OWL, and SPARQL, and includes a rule-based inference engine. After the data sent by the Robot are received, the Interface can build a query by using SPARQL Query Language, and ask the OWL Store to generate a list of appropriate BPs. This query contains a single row that can be found in Table 1, denoting the current environmental conditions that are currently valid for the considered Robot. This query is defined as a vector (1):

$$S_n = \left[TC2_n, TC1_n, CS_n, OB_n, VS1_n\right]. \qquad (1)$$

Row 16 in Table 1 can be analysed to demonstrate how the Model works. The status of the S_{16} vector is [1, 0, 1, 1, 0]. It can be concluded that the Robot, which sends the message containing this vector to the Model, carries tool no. 2 (TC2=1, TC1=0). Moreover, the Capacitive Sensor has its own value, set to 1 (CS=1), meaning that there are many Part Carriers in a queue behind the Stopping Place (Fig. 2). This assumption can be used as an indicator for possible production bottlenecks. Since the Optical Barrier has its sensor value set to 1 (OB=1), there is a new part that waits to enter the assembly system. According to the provided evidences, the Core Ontology suggests two possible robot actions, BP2 and BP4 (as can be seen in Table 1). The logical conclusion would be either to release the Part Carrier because of the queue behind the Stopping Place (Behavioural Pattern 2 represents a release signal for the Stopping Place), or to engage BP4 to check the Optical Barrier to determine the type of part. Fig. 12 depicts an algorithm defined within BP4.

If the Robot gets a response from the Cognitive Model (more precisely, from the Bayesian Reasoning part of the Model) suggesting that this algorithm should be engaged, a program named BP4 will be started. The algorithm (Fig. 11) has two main parts. An order of execution is defined within the algorithm. The first part would be engaged if Vision Sensor 2 (VS2), placed on the Robot's hand, detects Part1 while checking the Optical Barrier (VS2=Part1). In such cases, the Robot will simply pick up Part1 and place it into the Part Carrier located at the Stopping Place. On the other hand, if Part2 is detected, the Robot will remove it from its current position near the Optical Barrier and put it back at the beginning of the Conveyor. This operation could be engaged repeatedly as long as Part1 appears in front of the Optical Barrier. After that, the Robot can make the pick and place operation to placePart1 into the Part Carrier. When that happens, the Core Ontology should suggest a new list of BPs according to the new status of sensors (now VS1 can detect Part1 at the Stopping Place).A scan be seen, the Tool Changing Operation is used to determine if the Robot carries the right tool for the part according to the information provided by VS2.

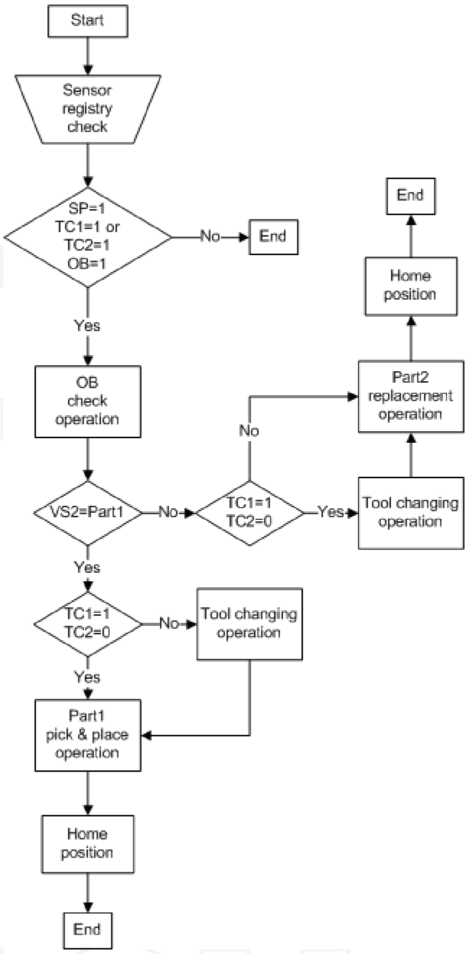


Figure 12. The algorithm of Behavioural Pattern 4

After the Core Ontology suggests the list of the most appropriate BPs, the Bayesian Network needs to provide an unambiguous solution in accordance with designer expectations (beliefs). Fig. 13 shows the Bayesian Network that has been developed and that contains designer expectations about targeted robot behaviours. These behaviours should provide optimal decisions at the group (global) level.

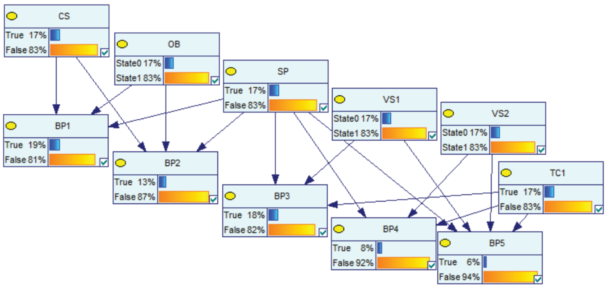


Figure 13. The Bayesian Reasoning part of the Cognitive Model

The edges within the Network represent casual connections between parent and child nodes. The parent node carries the information about the probability value representing a timely change of sensor statuses. Child nodes provide information about the strength of influences that the connected parent nodes have on them. These values denote beliefs of the system designer about related events that could lead to the achievement of certain production goals. Based on those beliefs, each robot gets its own probability estimation about suggested future actions as a response to significant environmental states, e.g. the availability of equipment and new parts, or sensor statuses.

At this point, it is possible to set evidences about certain events, e.g. a status of one sensor may be set to indicate that a certain event somewhere in the environment is detected. According to the provided evidences and a topology of the Network, probability values of connected nodes may be changed. By setting new evidences, it is possible to change beliefs about the most appropriate robot actions. Fig. 14 shows the same Bayesian Network after the evidences are provided.

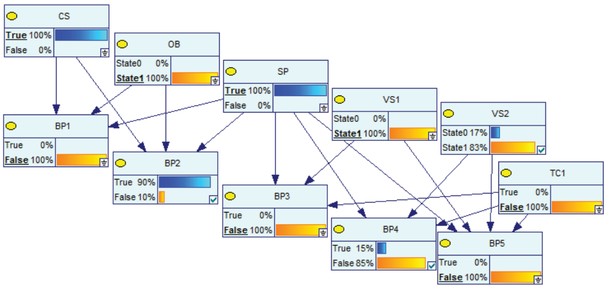


Figure 14. The Bayesian Reasoning part of the Cognitive Model

As can be seen, after providing the new evidences defined by the vector S_{16},the Bayesian Network believes that the most appropriate BP for the Robot to engage is the BP2, and the Robot will release the Part Carrier to the next Assembly Place.

The application is about to be tested on real equipment according to the previously explained system characteristics: scalability, context-awareness and self-healing. It is expected to have initial test results by the end of the year.

5. Conclusions

In this paper, the Cognitive Model for robot group control based on a semantic domain representation and a BN is discussed. A partial contextual cognition is achieved by means of ontology, which can be suitable for knowledge storing, sharing and reuse. In order to achieve a behavioural component, a probabilistic approach based on the BN is used. This approach enables a single robot to choose its actions.

Observing the proposed Model, a couple of conclusions should be made. Deterministic chaos inevitably obstructs absolute expectations, always producing slightly changed situations. Deterministic chaos could be accepted as a natural phenomenon. Respecting that conclusion, development philosophy should change toward the development of intelligent machines capable of adapting to their natural environment, where nothing is ideal or accurate. To alter uncertain situations, conventional automation methods tend to create technical systems as almost-perfect constructions. It seems that such efforts are hopeless and can result in expensive and inefficient systems. This approach can raise more issues that affect almost every contemporary industrial factory in the world: a lack of space and rigidity of production systems. These problems are particularly prominent in Europe. Making industrial systems adaptive, small, cheap and competitive with the rest of the world is an issue that appeared many years ago. The system based on contextual perception can be converted to work on other similar tasks relatively easily compared to classical industrial production lines. A robot with the proposed properties is able to partially interpret and understand the context in order to adapt its strategies to effectively work in a group on new assignments.

The presented Cognitive Model enables some degree of contextual understanding and provides a means for the agent to plan its actions by observing its environment. Probabilistic reasoning based on the BN can be used to increase efficiency and overall system performance to produce optimal production goals. It is possible to use a similar approach to alter robot group behaviours on different kinds of industrial applications, e.g. robot joint assembly assignments or any other applications where robots have to make choices that will not affect the quality of its functional performances. Work on Cognitive Model implementation for diverse applications will therefore continue.

6. Acknowledgments

The authors would like to acknowledge the support of the Croatian Ministry of Science, Education and Sports, through projects No.: 01201201948-1941, Multi- agent Automated Assembly, and the joint technological project TP-E-46, with EGO-Elektrokontakt d.d.

7. References

[1] Stipancic T, Curkovic P, Jerbic B (2008) Robust autonomous assembly in environment with relatively high level of uncertainty. In: MITIP 2008, S.R.O. Press,Czech Republic Bily Slon. pp. 193-198.

[2] Group of authors (2009) Robotic Visions – The strategic research agenda for robotics in Europe. European Robotics Technology Platform, Brussels.

[3] Stipancic T, Jerbic B, Curkovic P (2011) Probabilistic Approach to Robot Group Control. Advanced Materials Research. 317-319: 742-749.

[4] Gruber T (1994) Toward Principles for the design of ontologies used for knowledge sharing. International Journal Human-Computer Studies.43: 907-928.

[5] Costa PCG, Laskey K.G, Alghamdi G (2006) Bayesian ontologies in AI Systems. In: Proceedings of the Fourth Bayesian Modelling Applications Workshop, Cambridge, MA, USA.pp. 52-59.

[6] Lemaignan S, Siadat A, Dantan J.Y, Semenenko A (2006) MASON: a proposal for an ontology of manufacturing domain. IEEE Workshop on Distributed Intelligent Systems, Prague, Czech Republic. pp. 195-200.

[7] Lohse N (2006) Towards an ontology framework for the integrated design of modular assembly systems, Ph.D. Thesis, University of Nottingham.

[8] Al-Safi Y, Vyatkin V (2007) An Ontology-Based Reconfiguration Agent for Intelligent Mechatronic Systems. Lecture Notes in Computer Science. 4659: 114-126.

[9] Rampersad H.K(1994) Integrated and Simultaneous Design for Robotic Assembly. Chichester: John Wiley & Sons Ltd.

[10] Mittal A (2007) Bayesian Network Technologies: Applications and Graphical Models. London: IGI Publishing.

[11] Larik A, Haider S (2010) Efforts to Blend Ontology with Bayesian Networks: An Overview. In:Proceedings of 3rd International Conference on Advanced Computer Theory and Engineering. Chengdu, China.2: 598-603.

[12] Peng Y (2005) A Bayesian Network Approach to Ontology Mapping. The Semantic Web - ISWC 2005. pp. 563-577.

[13] Andrea B, Franco T (2009) Extending ontology queries with Bayesian network reasoning.In: International Conference on Intelligent Engineering Systems. pp. 165-170.

[14] Devitt A, Matusikova K (2006) Constructing Bayesian Networks Automatically using Ontologies. Applied Ontology,lOS Press. 1:1-11.

[15] Zheng H, Kang B, Kim H (2008) An Ontology-Based Bayesian Network Approach for Representing Uncertainty in Clinical Practice Guidelines. ISWC International Workshops. Heidelberg: Springer-Verlag. pp. 161-173.

[16] Fenz S, Tjoa A, Hudec M (2009) Ontology-Based Generation of Bayesian Networks. In: Proceedings of the Third International Conference on Complex, Intelligent and Software Intensive Systems, Fukuoka, Japan. pp. 712-717.

[17] Costa P, Laskey K.B (2006) PR-OWL: A Framework for Probabilistic Ontologies. In: Proceedings of the Conference on Formal Ontologies and Information Systems, Amsterdam, Netherlands.pp. 237-249.

[18] Costa P, Laskey K.B, Laskey K.J (2008) PR-OWL: A Bayesian Ontology Language for the Semantic Web. Uncertainty Reasoning for the Semantic Web I: ISWC International Workshops. Heidelberg: Springer-Verlag. pp. 88-107.

[19] Laskey K.B (2008) MEBN: A language for first-order Bayesian knowledge bases. Artificial Intelligence. 172: 140-178.

[20] Sure Y, Staab S, Studer R (2009) Ontology Engineering Methodology. In: Staab S, Studer R(Eds.). International Handbooks on Information Systems. Berlin: Springer-Verlag. pp. 135-152.

[21] Dey A. K (2010) Context – Aware Computing. In: Krumm, J. (Ed.), Ubiquitous Computing Fundamentals, Washington: CRC PRESS.pp. 321-352.

[22] Knublauch H, Fergerson R.W, Noy N.F, Musen, M.A, (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. International Semantic Web Conference 2004, Hiroshima, Japan, pp. 229-243.

[23] Thurn S, Burgard W, Fox D (2006) Probabilistic Robotics. London: The MIT Press. pp. 4-8.

[24] Twardy C.R, Nicholson A. E, Korb K.B, John McNeil J (2006) Epidemiological data mining of cardiovascular Bayesian networks. Electronic Journal of Health Informatics.1: 1-13/www.ejhi.net.

[25] Go Y.C, Sohn, J.C, Cho Y.J (2006) Probabilistic approach to model intelligent robot services. In: Proceedings of the 8th International Conference on Advanced Communication Technology, Phoenix Park, Korea. pp. 1221-1224.

[26] Group of authors (2012)Genie & Smile. Available: http://genie.sis.pitt.edu. Accessed: 20 August 2012.

[27] Donahoo M.J, Calvert K.L (2009) TCP/IP Sockets in C: Practical Guide for Programmers. Burlington: Elsevier Inc.

[28] Group of authors (2012) Apache Jena Project. Available: http://jena.apache.org. Accessed: 20August 2012.