Automated Systems & Technologies
25-26 May 2015 • St. Petersburg, Russia

KNOWLEDGE BASED DECISION MAKING FRAMEWORK

Aleksandr Fedorov, Vyacheslav Shkodyrev

Department of CST
Peter the Great Saint-Petersburg Polytechnic University
29, Polytechnicheskaya st., St.Petersburg, 195251, Russia
e-mail: aleksandr.v.fedorov@gmail.com

Abstract

This paper describes the different features and theoretical advantages of knowledge based framework in smart manufacturing. The main purpose of developed framework is smart reorganization of automation line based on the knowledge entered to the system by the experts and integration of low-level decision-making system and high level planning. The output of system leads to minimizing costs and maximizing of the efficiency of production line by preventing faults, dangerous situations and optimizing plans of production. In paper defined the main technics and technologies that used in prototype of the system. Also provided the modelling results on prototype system for chemical automation line.

INTRODUCTION

Today's Intelligent or smart manufacturing is one of the promising approaches of automation and robotization of big industrial distributed systems. Modern trends of industrial systems development show that such a development is tightly connected with adoption of artificial intelligence approaches and attempts to increase performance, especially in case of complex manufacturing. Usually information systems on plants have large-scaled very complex structures with high number of unknown or badly estimated parameters. Leading vendors of automation equipment have surpassed all imaginable expectations about development of complex robotic systems, means of information perception, increase in number of control channels, intellectualization of the low level of industrial automation such as PLC, fieldbus systems, sensors and actuators.

Knowledge usage in smart manufacturing can be divided into two main sections: 1) in domain for planning and execution domain-related actions 2) internally in manufacturing systems for optimization of manufacturing systems work i.e. independent actual application.

This paper presents an intelligent knowledge-based framework, showing how it is possible to improve user experience of large interactive systems via control theory and neuro-fuzzy-based learning as mentioned in [1], where authors of the paper con-sider a problem of operational process organization and optimization in application to manufacturing plants and models as in Fig. 1, a highly modular and flexible manufacturing plant resigning in Festo laboratory in SPbSTU – Chemical station.

Abstract dataflow diagram of the final system is depicted in Fig. 1.

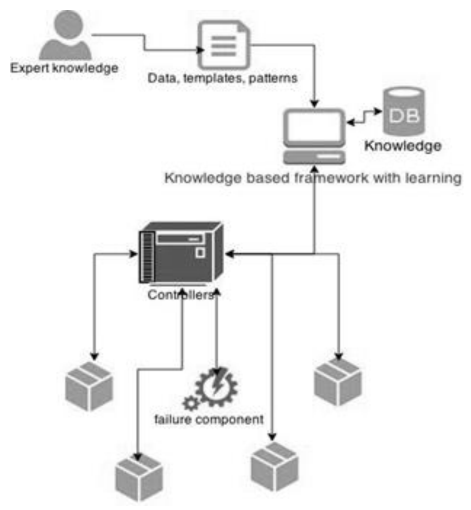
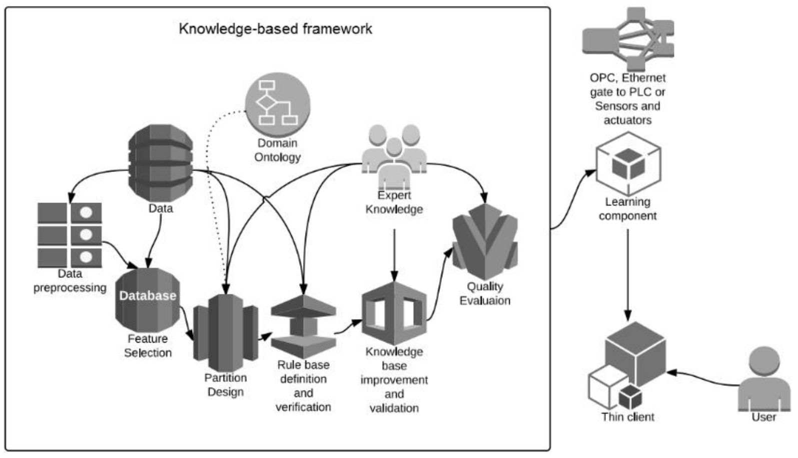

Expert knowledge
Data, templates, patterns
DB
Knowledge
Knowledge based framework with learning
Controllers
failure component

Fig. 1. Knowledge based system framework for smart manufacture

Domain experts import data in simple format such as Excel-file and system generates the representation of defined rules. A user then can connect each term with some sensor or actuator from PLC or any input/output device that can be reached via standard industrial acquisition technology (OPC). Next step for user to define boundaries for sensor information. Another case of program is that user can generate rules and launch system in learning phase where it can generate training data and find some abnormal behaviour if it will occur in production.

Knowledge based system considered in [2] takes some experts' data as an input, for example, plans for actions in emergency situations. Based on the rules and fuzzy logic algorithm the system then outputs data for intelligent management and supervision of system services. A Java-based framework will be used for generating understandable and accurate models as explained by authors in [3]. Dataflow of the joint approach are represented in Fig. 2.



Knowledge-based framework
Domain Ontology
Data
Expert Knowledge
Data preprocessing
Database
Feature Selection
Partition Design
Rule base definition and verification
Knowledge base improvement and validation
Quality Evaluaion
OPC, Ethernet gate to PLC or Sensors and actuators
Learning component
Thin client
User

Fig. 2. Scheme of the proposed knowledge-based environment.

INTEGRATED INTELLIGENCE FOR PLANNING AND FAULT-TOLERANCE

For systems build as a union of separate agents one may design a layer of integrated self-supervision and control powered by internal intelligence [4]. A view of decision-making process is presented for a single agent in the following paragraphs.

For making decisions to find optimal solutions were chosen partially observable Markov decision processes (POMDP) model. This model has proven to be an effective solution to the problem of finding an optimal sequence of actions that lead to the highest winnings.

A Partially Observable Markov Decision Process is defined as a tuple M = (S, A, O, T, Ω, R), where

- S is a set of states,
- A is a set of actions,
- O is a set of observations,
- T is a set of conditional transition probabilities,
- Ω is a set of conditional observation probabilities,
- $R: A \times S \rightarrow R$ is the reward function.

The example of system architecture with POMDP solver showed on Fig.3:

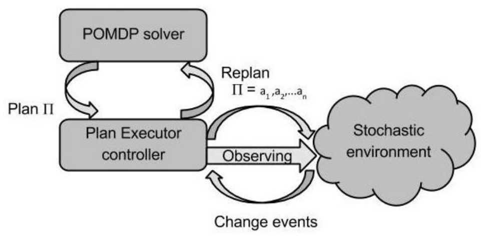


Fig. 3. Component architecture for planning and fault-tolerance

System actions in most basic view can be just leave component as it is, reload it, turn on or turn off. Systems states can be working or not. System read messages of type: "No logs – reload this module" and "No messages – reload this module".

Any component can ping another components and if it will not get any reply it will send messages like "Component 1 has crashed"
After getting the fault messages, some supervisor monitor will decide the appropriate action for that component.
System states can be divided into:

- Down – component can be reloaded
- Crushed – component cannot be reloaded without human interaction – for example after 20 attempts module will be marked as crushed and report will be sent to the responded people.
- Incapable – component is still alive but can't doing its job
- Working – component works properly

The following example shows how POMDP can be adopted for fault tolerant systems implementation.

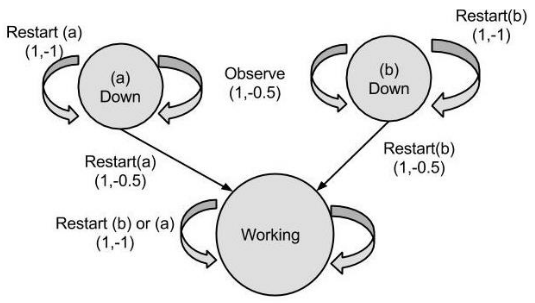


Fig. 4. Recovery model

Figure 4 shows a simple example of how an MDP might be used to model the recovery process of two components (a) and (b). In the figure, the different states represents is component works or it is down. Every component can observe its state with unavailability reward of 0.5 and if it down – restart it

with reward of 0.5. But if components works good and they tried to restart – cost will be more – unavailability reward 1. The same situation for restarting component in a case when it will lead to the down of component.
For reacting on any changed states the POMDP need a special policy, which can maximize reward after achieving of goal. The optimal policy, denoted by p*, yields the highest expected reward value for each belief state, compactly represented by the optimal value function V*. This value function is solution to the Bellman optimality equation:

$$V^*(b) = \max_{a \in A}[r(b,a) + \gamma \textstyle\sum_{o \in O} \Omega(o|b,a) V^*(\tau(b,a,o))], \qquad (1)$$

where γ is the discount factor, τ is the belief state transition function and b is a belief state probability. This value function can be find with a variety of algorithms (for references see [4, 5]).

As one of the best fitted to the problems algorithm considered Fuzzy Sarsa algorithm. Sarsa is an on-policy temporal difference learning algorithm. The main principle of Sarsa is summarized by its name: State, Action, Reward, State, Action. In ongoing research, these two algorithms will be compared in a chosen problem and the best one will be chosen for a future development [6].
Another advantage of using POMDP in planning that it can optimize path of getting the final product on manufacture and can be scaled since it can be parallelized.

PROTOTYPE IMPLEMENTATION

Concepts described in previous sections are used as basic building blocks of automa-tion systems that are designed and built on the premises of Festo Laboratory in Saint Petersburg Polytecnical Univesity. The lab consists of two manufacturing models – a discrete and continuous manufacturing.

Chemical station manufacturing consist of four modules – mixing station, filter station, reactor station and bottling station. Full station can be represented as small plant with 20 actuators and 44 sensors. For simplicity in this paper only a subset of sensors is considered - Temperature mixture of liquids, Condition of three valves, Condition of mixer, Level of liquid, Time of mixing.

Depending on a combination of these sensors an adaptive neuro-fuzzy inference system can generate a target vector from which we can obtain information – whether the temperature is exceeded or liquid level was too high or low-level system can report alarm if expert introduced the boundaries of parameters.

The main problem of ANFIS toolbox is that for 7 terms and all combinations of rules (6804 rules) system trains model for a long period of time. For example for 6804 rules the time consumption of training the model takes 6 minutes – which is unacceptable for that kind of small problem. For making training faster considered to use java neuro-fuzzy framework, which can be easily parallelized on the same PC.

SUMMARY

Each day the number of unknowns in the production increases, as well as the number of controllable parameters. One way to effectively manage and optimize production is the proposed system knowledge bases with the possibility of self-study, which is part of the scheduler high and low level of production. Described techniques improve characteristics of resulting automated systems comparing to traditional approaches of design and implementation: fault-tolerance, scalability and latencies; that fact is proven by systems designed and implemented on the premises of Festo Laboratory in Saint Petersburg Polytecnic Univesity and in customer project. Further development of the project involves the implementation and integration of the components of the optimization and refinement.

REFERENCES

1. D.G. Arseniev, V.P. Shkodyrev, V.V. Potekhin, V.E. Kovalevsky, "Smart Manufacturing with Distributed Knowledge-Base Control Networks.", in Proceedings of Symposium on Automated Systems and Technologies, Hannover, PZH Verlag, 2014, pp. 5225-5230.
2. C. Legat, J. Neidig, M. Roshchin, "Model-based Knowledge Extraction for Automated Monitoring and Control", Preprints of the 18th IFAC World Congress Milano (Italy) August 28 - September 2, 2011, pp. 85-89.
3. J.M. Alonso, L. Magdalena, "Guaje – a java environment for generating understandable and accurate models", in: XV Spanish Conference for Fuzzy Logic and Technology, Universidad de Huelva, Spain, 2010, pp. 399–404.
4. A.V. Fedorov, S.S. Zobnin, V.V. Potekhin, "Prescriptive analytics in distributed automation systems", in Proceedings of Symposium on Automated Systems and Technologies, Hannover, PZH Verlag, 2014, pp. 43-49.
5. H. Milos, "Value-Function Approximations for Partially Observable Markov Decision Processes," Journal of Artifcial Intelligence Research 13, 2000.
6. L. Tokarchuk, J. Bigham, and L. Cuthbert1: "Fuzzy Sarsa: An approach to fuzzifying Sarsa Learning" International Conference on Computational Intelligence for Modelling Control and Automation, 2004.