Data-Driven Maritime Processes Management Using Executable Models

Tomáš Richta[1], Hao Wang[2], Ottar Osen[2], Arne Styve[2], and Vladimír Janoušek[1]

[1] Faculty of Information Technology,
Brno University of Technology,
IT4Innovations Centre of Excellence
612 66 Brno, The Czech Republic
{irichta,janousek}@fit.vutbr.cz

[2] Big Data Lab, Dept. of ICT & Natural Sci.
Norwegian Univ. of Sci. & Tech.
Aalesund, 6009 Norway
{hawa,ottar.osen,asty}@ntnu.no

Abstract. In this paper we describe a decision support system for maritime traffic and operations, based on formal models and driven by data from the environment. To handle the complexity of system description, we work with a decomposition of the system to set of abstraction levels. At each level, there are specific tools for system functionality specification, respecting particular domain point of view. From the business level point of view, the system consists of processes and vehicles and facilities over those the processes are performed. From the engineering point of view, each process consists of a set of devices, that should be controlled and maintained. Software engineering point of view operates on reading and converting bytes of data, storing them into variables, arrays, collections, databases, etc. For complex trading processes management purposes we need to cover all levels of abstraction by specific description, suitable to model and automate the operations on each particular level. As a case study we use salmon farming in Norway. The system implementation is based on *Reference Petri nets* and interpreted by the *Petri Nets Operating System* (PNOS) engine. This approach brings formal foundations to the system definition as well as dynamic reconfigurability to its runtime and operation.

1 Introduction and Motivation

In this paper we focus on describing the system for maritime traffic and operations support, based on formal methods and driven by the data from environment. Some of the work has already been done in this area. For example Ray et al. base their Decision Support System (DSS) on the idea that it needs to include mechanisms from which operators can define some contextual situations he wants to be detected as suspicious, dangerous or abnormal. They build this mechanism on a rule-based engine approach allowing to formalise rules ensuring the link between the conceptual specification of a situation and its implementation. The main aim of their DSS is a design, where the business logic might be re-configured by a surveillance operator [6].

Production rules are defined as fragments of knowledge, that can be expressed in the format: WHEN conditions are verified THEN perform some actions, where the WHEN part is referred to as the left-hand side, and the THEN part as the right-hand side. This format allows experts to express their knowledge in a straightforward way, without using any specific programming language and therefore removing the need for a computer programmer to assist the expert in encoding his knowledge [6]. Some of these ideas were already addressed e.g. by Ludwig Ostermayer and his colleagues [5].

We suggest a decomposition of the problem to a set of abstraction levels to reduce the complexity of a whole problem definition. This approach also allows for separating the concerns of different domains specialists as well as languages and tools they use for particular level specification. Simmilar ideas could be also found in some literature about expert systems like e.g. [4].

2 Maritime Logistics and Operations

We use salmon farming in Norway as a case study. Salmon production starts with hatching of eggs in freshwater tanks on land. After 1 - 1.5 years the juvenile salmon goes through a physical transformation process that is called smoltification that prepares the fish for life in seawater. The salmon is now called smolt and is ready to be transferred to the sea cages.

In the sea the salmon is fed pelleted feed for 1 to 2 years. Due to the high concentration of salmon it is common to add oxygen to the water and to remove CO_2. The salmon is harvested when it has reached optimum size. This is usually done by pumping the salmon into a well-boat and shipping the live salmon to the salmon processing plant.

Aquaculture is a profitable business dominated by big companies. In order to maximise the profit there are continuous efforts put on optimising the process. Optimisation of: time at sea (fast growth), fodder, produced biomass vs fodder volume, harvesting time, medicine, O_2 usage and fish quality. In later years sea-lice has been a problem for aquaculture companies, in addition to other pathogens such as toxic algae. In order to succeed a close control of biomass production at every step in the process is vital.

3 Rule-based Modelling

Each action in the system produces some data that are sent to the particular rule engine that decides, what action should be taken. Rules apply to much more higher number of situations, and they also must be applied first, before the action caused by the task occurrence within the process could take the place. Rules within the system trigger the task fulfilment, and therefore a start of following task.

The important problem is the language used for the rules definition. The main rule clause structure is when-eval-then. But the definition of all these three parts is not constrained at all, or the constraints depend on the environment used for rules execution, like Java in Drools or RPNs in our example.

4 Data-driven System

Technologies are being adopted for acquiring monitoring data about how the vehicle and different components are behaving. Recently, with the intention of remote ship monitoring for better services for shipping customers, vessel builders started to adopt new sensor technology by installing different sensors for different components on board a vehicle and transmit data using satellite communications to land-based service centres, e.g., *HEalth MOnitoring System* (HEMOS) by Rolls-Royce Marine AS.

These systems provide more accurate and timely operational data, but they also introduce new danger to the operations: *information overload problem* (IOP) [3, 9] – the crew members receive a large volume of monitoring information and alert messages that s/he can easily overlook important/vital ones. Therefore, it is urgently needed to develop and implement a new framework to integrate and visualise the monitoring data in an informative way. In this way, the crew members can examine the massive, multi-dimensional, multi-source, time-varying information streams to make effective decisions in time-critical situations.

Our system bases on data flows and their processing according to predefined rules similarly as Ray et al. defined in their system [6], where the AIS (Automatic Identification System) data are processed by the rules engine producing the specific information and warnings about vessels movement and behaviour.

5 Levels of Abstraction

To be able to define the whole system functionality while reducing the complexity of the problem, it is better to separate it by a set of levels of abstraction [4]. Each level could be seen as a sole system, consisting of nodes, communication means and dependencies checking. Each system operates on nodes specified in more detail within the level below it. From the level 3 to 5 the nodes of the system could be taken as actors (ref. Actor model), in levels below, they behave less independently.

5.1 Level 5 - Aquaculture Facilitation

This level represents a set of processes forming the maritime trade. When performing each task, the facilitation system uses services from Level 4. This level of abstraction is intended to be used by the maritime trading management people. The most appropriate way of modelling processes at this level seems to be the sequence of tasks with dependencies among them as well as participant involved. For example using BPMN notation. At this level, basic processes of the system are defined.

5.2 Level 4 - Vessels Chartering, Berthing Process, etc.

This level defines a set of nodes and communication means involved within trading processes that takes the place by serving as a platform for the Level 5 processes organisation. I.e. this level is an decomposition of participants from the level above. This level is intended for modelling vessels, ports, etc. relationships together with relevant communication channels.

5.3 Level 3 - Vessels, Ports, etc.

This level describes the functional nodes with independent behaviour that use services of modules from the Level 2 and serve as services for the level 4. This level is defined as Workflow System Specification and could be directly transformed to the interpretable Reference Nets structure for further process management purposes [7]. Typical example of system parts at this level of abstraction are independent units usable for the Level 4 purposes, like vessels, fish farms, or fish factories.

5.4 Level 2 - Modules

This level describes assembled components providing specific set of services within Level 3 models. Modules consist (physically or logically) of components from the Level 1 and are usually controlled by staff, or also using any kind of programming interface, or both. Components communicate among others using defined protocol. Modules could be represented by e.g. navigation module, dynamic positioning module, wellboat pumping and cleaning module.

5.5 Level 1 - Components

This level covers mountable devices with well-defined and encapsulated behaviour defined as a set of primitive operations defining the protocol of the component. The example of a device at this level is pumping component operating over one pipe within pumping facilities. Components operate on parts from the Level 0 and are accessible via programmable interface or some specific of bus. Here the appropriate examples of components belong to thrusters, engines, pumps, etc.

5.6 Level 0 - Sensors and Actuators

At this level, simple parts mounted within the environment take place. Sensors are able to read data from the environment and serve it as raw values, or digitised and calibrated. Actuators have direct effect on the environment, it means these are e.g. multiple types of switches, servos, motors, etc. In PNOS, sensors and actuators are triggered by invoking primitive operations bounded with Reference Petri Nets transitions. These operations produce or consume values in specific strings-based format, which are propagated through the system to particular

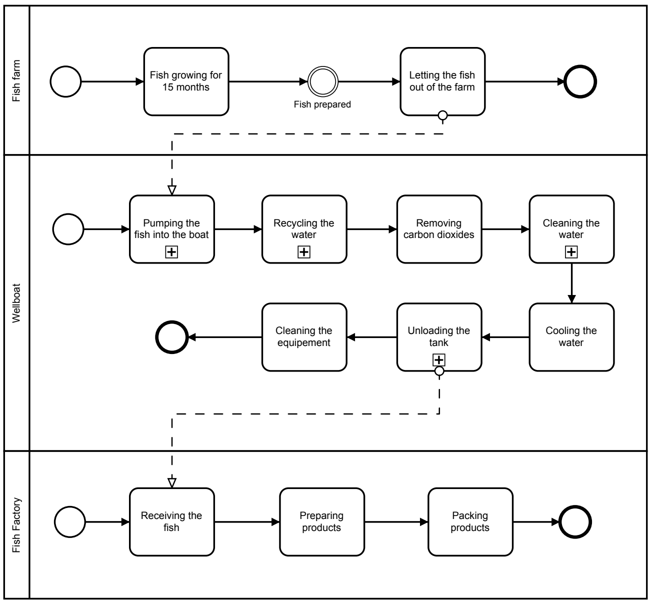

Fig. 1. Wellboat Process Description (Level 4)

node they are dedicated. The important part of each node is its ability to store rules for data filtering, before they are directly sent to upper levels of the system. Data could be also modified or combined by these rules.

6 Running Example

System construction process will be described on real-life scenario of wellboat operations and technology. Wellboats carry fish from fish farms to fish factories. Fish are pumped from the farm into the boat and then transported to the factory, where they are pumped back again. The water with fish is treated following some predefined rules to keep the fish in good conditions. While pumping the fish out of the boat, it is possible to separate them according to their size. An example of described process definition could be found in Fig. 1.

From the point of view of control system structure, there are three control sub-systems of fish farm, the wellboat itself, and the fish factory.

7 System Construction Process

Management of distributed trading processes must take into account many involved nodes and regarding the maritime processes, there is also necessary to take into account the conditions coming from the fact, that processes are undertaken on the sea. One of the main influencing condition is that ships and their

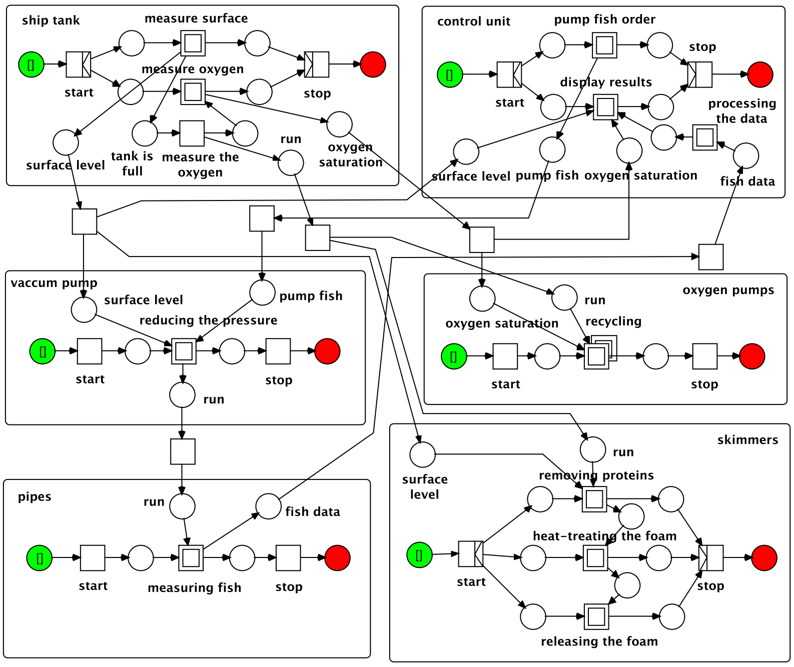

Fig. 2. Maritime System Example (Level 3)

crew could in some situations remain without the connection with the land. Therefore it is necessary to count on with adequate control system installation and communication ways.

Particularly it means that the system must be distributed and all the nodes must be able to behave independently on the connection to other nodes, as well as some particular sets of nodes that operate together should be able to act independently on the rest of the system. This leads to the isolation of particular sub-ecosystems, like the vessel control system, port control system, etc. that together form the process management platform. These ecosystems are defined at each level of abstraction and represented as a subset of PNOS installations.

7.1 Data Propagation and Analysis

There are two ways of data propagation - A) from the top to the bottom and B) from the bottom to the top. At each layer of the system decomposition there are PNOS nodes that allow to retrieve commands and Petri nets specifications from the above layers as well as the data produced by layers below. Each PNOS node hosts a set of Petri nets that perform commands. Other Petri nets are responsible for filtering data coming from lower layers.

Model-Driven Engineering moves the software engineering paradigm to the level, where the code itself does not play the central role of the application design and implementation, but more abstract model of the application logic takes the place as a first-class artifact within the development process [8]. This approach makes it possible to distinguish between modelling the application logic by the domain expert or specialist and the interpretation or transformation of the model into its executable form.

There are many papers describing model transformation into executable code, but all of these approaches lack the dynamic reconfigurability features as well as preserving the model during the runtime, therefore the model execution got more attention among researchers now [2]. Basic model transformations and target system construction process is documented in our previous papers [7].

7.2 Runtime and Reconfiguration

The core characteristics of resulting system - its formal basements and dynamic reconfigurability - is in our solution based on the ability of Reference Petri Nets interpretable representations to migrate among places of the system as tokens, similarly as in reference Nets. The new or modified Petri Net, that represents the system partial behaviour change is sent over other Petri Nets to its destination place to change the whole system functionality. In our solution, these Petri Nets parts are maintained by the Petri Nets Operating System (PNOS) and interpreted by the Petri Nets Virtual Machine (PNVM) engine[7]. System decomposition is inspired by MULAN architecture [1].

The installation of the system starts with placing proper nodes to the target environment. Each node should be installed with the PNOS, PNVM and basic platform layer. In our running example the scenario should start with installing the processes for each Workflow Specification and then sending particular subprocesses nets to relevant nodes. All processes of the node could be changed and then passed to its platform to change the behaviour of the node. Finally all the sub-processes nets could be modified and sent to particular nodes processes that reinstall them within the nets place.

8 Conclusion

In this paper we discussed the problem of data driven distributed control and decision support system. The system itself is decomposed into five levels of abstraction to reduce the complexity of its construction. At each level a specific notation or formalism for system functionality description and operation is used. The main idea of system design is to enable target users with the possibility to maintain the system during its runtime by introducing new functionality as well as new rules defining the expert knowledge. The implementation of the system is based on Petri Nets Operating System (PNOS) that is able to interpret textual representation of Reference Petri nets called Petri Nets Byte-Code (PNBC). System itself is constructed as a set of Reference Petri Nets installed within nodes of the system. Parts of the system specification at each node represents rules defining the data propagation from each node to the above layers of the system.

Acknowledgment

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602 and partially by the Norwegian Funds under the academic staff mobility programme (NF-CZ07-INP-5-337-2016).

References

1. Cabac, L., Duvigneau, M., Moldt, D., Rölke, H.: Applications and Theory of Petri Nets 2005: 26th International Conference, ICATPN 2005, Miami, USA, June 20-25, 2005. Proceedings, chap. Modeling Dynamic Architectures Using Nets-Within-Nets, pp. 148–167. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
2. Girault, C., Valk, R.: Petri Nets for System Engineering: A Guide to Modeling, Verification, and Applications. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2001)
3. Keim, D., Andrienko, G., Fekete, J.D., Görg, C., Kohlhammer, J., Melançon, G.: Visual analytics: Definition, Process, and Challenges. In: Information Visualization. LNCS, vol. 4950, pp. 154–175. Springer (2008)
4. Nikolopoulos, C.: Expert Systems: Introduction to First and Second Generation and Hybrid Knowledge Based Systems. Marcel Dekker, Inc., New York, NY, USA, 1st edn. (1997)
5. Ostermayer, L., Seipel, D.: A prolog framework for integrating business rules into java applications. In: Nalepa, G.J., Baumeister, J. (eds.) Proceedings of 9th Workshop on Knowledge Engineering and Software Engineering (KESE9) co-located with the 36th German Conference on Artificial Intelligence (KI2013), Koblenz, Germany, September 17, 2013. CEUR Workshop Proceedings, vol. 1070. CEUR-WS.org (2013)
6. Ray, C., Grancher, A., Thibaud, R., Etienne, L.: Spatio-temporal rule-based analysis of maritime traffic. In: Third Conference on Ocean & Coastal Observation: Sensors and Observing Systems, Numerical Models and Information (OCOSS). pp. 171–178 (2013)
7. Richta, T., Janousek, V., Kocí, R.: Dynamic software architecture for distributed embedded control systems. In: Moldt, D., Rölke, H., Störrle, H. (eds.) Proceedings of the International Workshop on Petri Nets and Software Engineering (PNSE'15), including the International Workshop on Petri Nets for Adaptive Discrete Event Control Systems (ADECS 2015) A satellite event of the conferences: 36th International Conference on Application and Theory of Petri Nets and Concurrency Petri Nets 2015 and 15th International Conference on Application of Concurrency to System Design ACSD 2015, Brussels, Belgium, June 22-23, 2015. CEUR Workshop Proceedings, vol. 1372, pp. 133–150. CEUR-WS.org (2015)
8. Rutle, A., MacCaull, W., Wang, H., Lamo, Y.: A metamodelling approach to behavioural modelling. In: Proceedings of the Fourth Workshop on Behaviour Modelling - Foundations and Applications. pp. 5:1–5:10. BM-FA '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2325276.2325281
9. Wang, H., Zhuge, X., Strazdins, G., Wei, Z., Li, G., Zhang, H.: Data Integration and Visualisation for Demanding Marine Operations. In: Oceans 2016: MTS/IEEE Oceans Conference (2016)