Project Proposal

Master's Thesis (1st year)

Software and System Engineering Programme

School of Software Engineering / FCS

Full Name: Anton Rigin

Student ID – M191МСПИН019

Title: Reference and Data Semantic-Based Simulator of Petri Nets Extension with the Use of Renew Tool

NATIONAL RESEARCH UNIVERSITY

Supervisor: Sergey Shershakov

Senior Lecturer at School of Software Engineering,

Research Fellow at PAIS Lab,

HSE University's Faculty of Computer Science

Date of Proposal's Submission: 30 January, 2020

Moscow, 2020

Abstract

The complexity of the software and other systems is constantly growing, which is even more enhanced by concurrency of processes in the system, so modelling and validating of such systems is necessary in order to eliminate the failures. Because of the concurrency, simple finite state machines (FSMs) are not sufficient for solving this problem even if the FSM is non-deterministic. One of the most well-known formalisms for solving it is the Petri net and its modifications such as the coloured Petri net, the reference net and others. However, the Petri net and its listed modifications do not support modelling of working with the persistent data in the system. The solution is the db-net which is the formalism consisting of the three layers: the control layer which is represented by the modified coloured Petri net, the data logic layer which is represented by queries for retrieving the persistent data and actions for updating this data and the persistence layer which is represented by the relational database, its schema and constraints for keeping the persistent data in the consistent condition. However, no software implementations of the db-net formalism are found in the open sources. The purpose of the proposed project is to develop the db-nets software simulator. This simulator is planned to be developed as the plugin for the Renew (the Reference Net Workshop) software tool which is built as a collection of plugins written in Java. The SQLite embeddable relational DBMS is planned to be used for implementing the db-net's persistence layer. The project is expected to have elements of the incremental software development model and the prototyping is planned to be used, as well. The results can be used in the research projects where modelling the complex software system, which uses the persistent data, is necessary regardless whether the project has an academic or industry-oriented purpose. This includes the research projects of the Laboratory of Process-Aware Information Systems at the HSE University's Faculty of Computer Science (the PAIS Lab). Moreover, the results are expected to be used in the master thesis which should be based on the proposed term project.

Key Words: db-net; Petri net; Renew; relational DBMS; persistent data; software modelling; software validation.

Table of Contents

Introduction

In the current days, the complexity of the software and other systems is constantly growing. One of the most important constituents of such complexity is concurrency of the processes performed by the system since it brings a lot of uncertainty and non-deterministic behaviour.

Because of such behaviour the deterministic finite state machines (FSMs) cannot support modelling and validation of these concurrent systems at all. The non-deterministic FSMs are suitable in this case, but their amount of the possible states is growing significantly, and system modelling is becoming impossible for being interpretable by human or impossible at all as well as system validation.

One of the most popular modelling formalisms which solves this problem is the Petri net which was invented by Carl Adam Petri in 1939 [1]. It allows to represent the concurrent system as the directed bipartite graph which consists of such vertices as places and transitions and such nodes as arcs. Tokens are used to represent the system's resources and their distribution across the net's places called marking is used to represent the current state of the system [1] [2].

Although the Petri nets are useful technique to model the concurrent software systems and the problem of significant growth of the non-deterministic FSM graph's nodes with the system complexity increase is solved in them, there are still problem that the Petri net may become too large for being understandable or even infinite if the data types with the very large or infinite ranges of possible values are used in the software system. The solution is the coloured Petri net which supports data types called "colours", arc expressions, guard expressions and other useful tools. They firstly were described by Jensen Kurt in his article in 1997 [3] and then in the textbook by Kurt Jensen and Lars M. Kristensen in 2009 [4].

The coloured Petri net can be used for modelling the concurrent software systems with the data types which can contain any number of possible values. Also there exists yet another formalism based on the Petri net which is called reference Petri net or simply reference net. The reference nets allow to use references to objects as the tokens, including even references to other nets. One of the well-known software tools implementing the reference nets is the Renew (the Reference Net Workshop) [5].

Even though the Petri nets and their modifications can model the complex concurrent software systems' behaviour, they cannot easily model working with data in the persistent database. The solution called db-nets was proposed in 2017 by Marco Montali and Andrey Rivkin [6]. A db-net is a formal model which consists of three layers: control layer (the modified coloured Petri net), data logic layer (queries and actions which allow the control layer to retrieve and manipulate the data in the database) and persistence layer (the relational database with constraints which declare the data consistency rules). The last two layers and the modifications of the coloured Petri net used in the control layer allow to model working with the data in the persistent database while the control layer allows to model the control flow(-s) of the system as well as working with the local (non-persistent) data. Therefore, the db-net solve the problem of working with the persistent data in the model [6]. The schema of the db-net layers is shown on the Fig. 1.

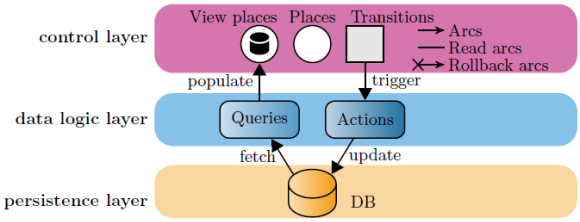

Fig. 1. The db-net structure [7].

Moreover, the db-nets can be used with the application of the reference semantics since the reference data type can be potentially considered as the ordinary data type (colour) in the coloured Petri net. It will allow to use complex data types with large values as tokens in the net.

Although the db-nets can improve the quality of modelling the concurrent complex software systems and their validation, especially those which use the persistent data, no their software implementations are found in the open sources. The Renew, which is mentioned above, is built as the collection of plugins written in Java and it is open-source, so extending this tool in order to support simulating the db-net's run seems to be an appropriate solution and it will form the main part of the proposed project.

The developed software simulator is expected to provide the ability to validate possible behavior of the designed complex concurrent software system even if we need to consider the persistent data used by the system. This can be used for modelling and validating the behavior of the real safety-critical software systems as well as for the further researches. The potential users of such simulator involve the research staff of the Laboratory of Process-Aware Information Systems at the HSE University's Faculty of Computer Science (the PAIS Lab) [8]. This simulator is also expected to be used in research applied to the real industrial software systems (for example, for the financial software systems) which will be conducted within my master thesis that should be completed and defended in 2021.

Thesis (Research) Statement

As it was stated above, the purpose of the project is to develop the program (the software simulator), which supports the db-net formalism, based on the Renew open-source software tool. This simulator should meet a certain number of functional requirements at least. These functional requirements are the following.

1) The program shall allow user to model the db-net's control layer using the Renew tool's graphical user interface (GUI) elements for net's interactive drawing.
2) The program shall allow user to create the database schema at the db-net's persistence layer using the Renew tool's GUI based dialog windows.
3) The program shall allow user to declare queries and actions for the db-net's data logic layer using the Renew tool's GUI based dialog windows.
4) The program shall allow user to simulate the modelled db-net's run using the Renew tool's GUI.
5) The program shall allow user to input the external data for the db-net's run during the db-net's run simulation where it is necessary according to the db-net model using the Renew tool's GUI.
6) The program shall allow user to save the modelled db-net using the Renew tool's GUI.
7) The program shall allow user to open the previously saved modelled db-net using the Renew tool's GUI.

The developed software product should consist of program (source code and executables) which meets the requirements listed above and developer's and user's documentation at least. The project should be completed by the first half of the June 2020.

Methodology (Approach / Methods)

In the proposed project the db-nets software simulator is planned to be developed as the plugin for the Renew software tool. The Renew tool consists of plugins which play a role of modules of this software tool. The basic way is to encapsulate the db-nets simulator in the separate plugin entirely. However, there is some possibility that some source code of the other plugins will also need to be modified.

The plugin is expected to use the Petri net model implemented in the Renew as a base for the db-net's control layer. The control layer also will have several modifications comparing to original net model implemented in the Renew. Firstly, it will have special places called "view places". These places are used for retrieving the data from the persistence layer through the data logic layer using the queries. Secondly, two new types of arcs are introduced (in addition to the original Petri net arcs): read arcs and rollback arcs. The former is used for connecting the view places with transitions and the latter is used for rollbacking the transition firing and database updates produced by this firing if the new database state violates one or more constraints declared at the persistence layer. The last modification is that transition firing can trigger actions from the data logic layer which leads to the persistent data update [6]. The example of the db-net's control layer is shown on the Fig. 2. The most left two places are the view places and the most right red arc with "x" symbol is the rollback arc. The view places are connected to the transitions using the read arcs.

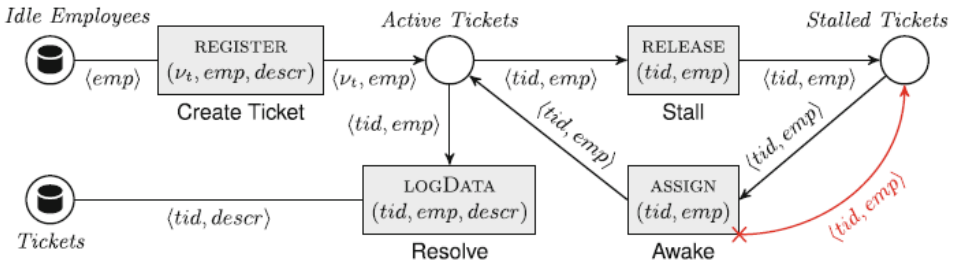

Fig. 2. The example of the db-nets control layer [6].

The plugin is expected to allow user to model the db-net's control layer using the Renew's GUI elements for the net's interactive drawing.

The data logic layer is the middle db-net's layer. It contains queries for retrieving the persistent data and actions for updating this data [6]. The plugin should provide user ability to create these queries and actions based on the persistence layer's database schema and to link the view places and transitions with the matching queries and actions respectively.

The persistence layer is the most inner db-net's layer. It is represented by the relational database, its schema and constraints. The data from the database is retrieved by the control layer's view places through the data logic layer's declared queries. It is updated by the control layer's transitions firing through the data logic layer's declared actions [6]. The plugin is planned to allow user to create the database schema and its constraints. The schema and constraints should be declared by user before start of the net's simulation and cannot be changed during the simulation running.

The main candidate for the relational database management system (RDMBS) used at the db-net's persistence layer by the plugin is the SQLite since it is the embeddable (not client-server), self-contained and relatively small database engine [9] which should be enough appropriate solution for our case.

Renew and all its plugins are written in Java, so the simulator will also be written in Java (Java 8 is considered to be used in the project). Also, the Java Database Connectivity (JDBC)

interface should be used to communicate with the SQLite engine. The Apache Ant is used as the build tool in the Renew, so the developed simulator should also use this.

The prototyping is used in the project. The simulator development starts with creating a prototype of the db-net's implementation where the real inner layers (the data logic layer and persistence layer) are replaced with stubs which are logging each query or action call and return the predefined response. This prototype is developed for the selected concrete case to be modelled in the db-net. After this prototype works correctly and it is approved by the main stakeholders of the project (by me and scientific supervisor at least), the further development for supporting the inner layers starts which may produce another intermediate prototypes. Each stage adds new functionality to the previous results, and this gradually leads to the software program which meets all the functional requirements through the kind of the incremental software development [10].

The developer's and user's documentation for the developed simulator should provide all necessary details in architecture of the developed plugin as well as user manual for the simulator. This completes the developed software product.

Preliminary (Expected) Results and Discussion

The main result is the software product which meets the requirements listed in the thesis statement section. This product is the db-nets software simulator based on the Renew software tool. This simulator should allow user to use the db-net formalism for modelling and validating the software systems in the Renew software tool environment.

To achieve this purpose, all the necessary literature sources must be studied carefully enough to implement the db-net formalism correctly, and the most of them are already studied. The Renew architecture research should be completed and the existing Renew modules with the source code that needs to be changed (if any) should be determined.

One of the most important constituents of the developed software product is the developer's and user's documentation. It will allow to learn all necessary details about the developed solution's architecture and source code as well as to read the user manual.

The developed software product will probably be used by the research staff of the Laboratory of Process-Aware Information Systems at the HSE University's Faculty of Computer Science (the PAIS Lab) [8] for the goals of modelling and validating the real safety-critical complex concurrent software systems considered in the PAIS Lab's current and future research projects including the financial software systems.

Work Plan (as seen in December-January 2020)

The project's plan is presented in the form of the Gantt diagram (chart) on the Fig. 3 (the first part) and on the Fig. 4 (the second part) created in the Microsoft Project Professional 2019 [11].

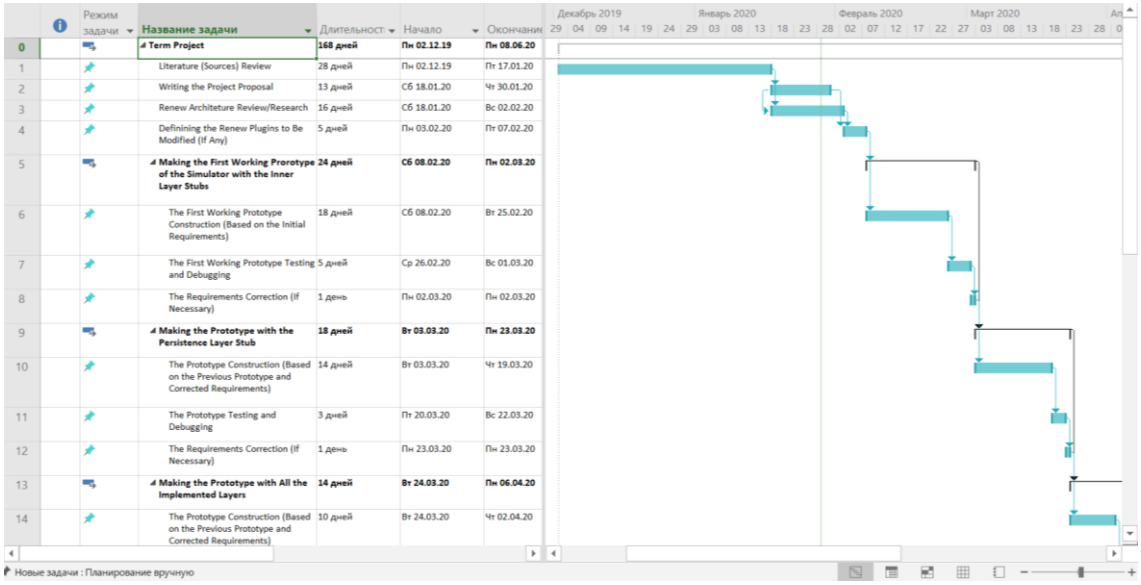

Fig. 3. The project's Gantt diagram (chart), part 1.

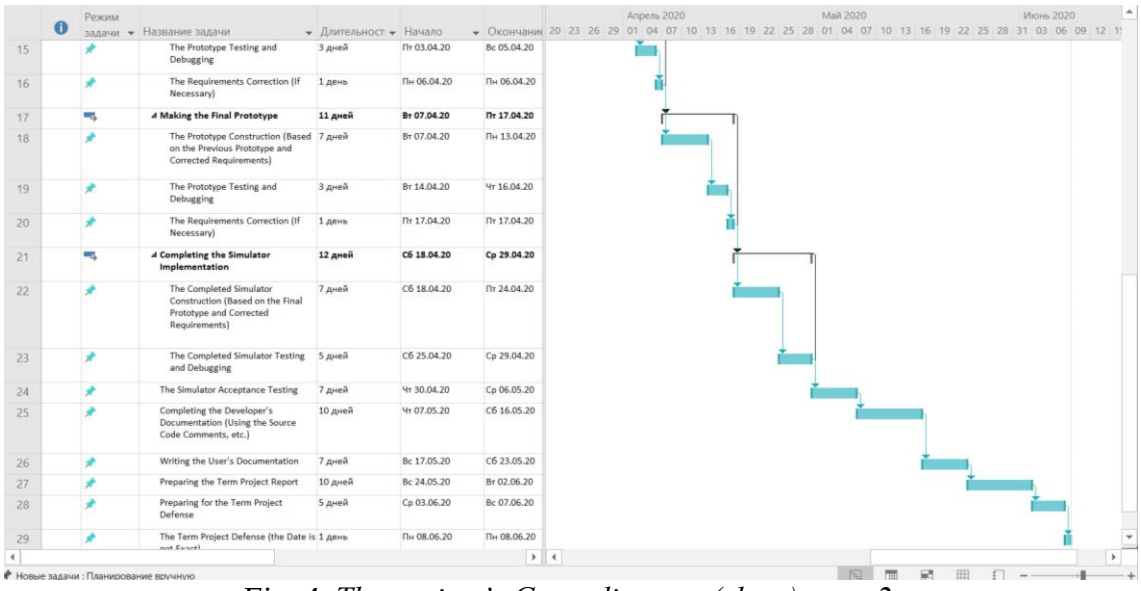

Fig. 4. The project's Gantt diagram (chart), part 2.

Since the project is performed not in a team, there are almost no parallel activities, and it seems that there is no sense to highlight the critical path of the project. The project uses the elements of incremental approach and prototyping.

Implications of the Project's Work

The db-net software simulator which is the main purpose of the project will provide the opportunity to use the db-net formalism for solving the problem of simulating the Petri net model which allows to model working with the persistent data, as well. Since now there are no db-net's implementations found in the open sources, this implementation will possibly be the first one. This simulator can be used in the research tasks as well as in the industrial tasks when the complex software system which uses the persistent data is under consideration.

This software tool can be used by researchers including the research staff of the Laboratory of Process-Aware Information Systems at the HSE University's Faculty of Computer Science (the PAIS Lab) [8] for conducting the laboratory's researches. Also, this simulator can be used for the further researches and modifying of the db-nets as well as for the developing new formalisms which use the persistent data.

Moreover, the simulator is planned to be used in my master thesis (in the 2020 – 2021 academic year) which will be the db-net based research of some real software system which is considered by the PAIS Lab. Probably it will be the complex concurrent financial software system since the financial software systems are often considered as safety-critical software systems because of possible large losses caused by software failures.

Bibliography (References)

[1] C.A. Petri and W. Reisig. “Petri net.” Scholarpedia. http://www.scholarpedia.org/article/Petri_net (accessed Jan. 25).
[2] W. Reisig. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Berlin, Heidelberg, Germany: Springer, 2013.
[3] K. Jensen. “A Brief Introduction to Coloured Petri Nets,” in *Third Int. Workshop on TACAS*, Enschede, the Netherlands, Apr. 2 – 4, 1997, pp. 203 – 208.
[4] K. Jensen and L.M. Kristensen. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Berlin, Heidelberg, Germany: Springer, 2009.
[5] “Renew – The Reference Net Workshop.” Renew.de. http://www.renew.de/ (accessed Jan. 25, 2020).
[6] M. Montali and A. Rivkin. “DB-Nets: On the Marriage of Colored Petri Nets and Relational Databases,” in *Transactions on Petri Nets and Other Models of Concurrency XII*, M. Koutny, J. Kleijn, W. Penczek, Eds., Berlin, Heidelberg, Germany: Springer, 2017, pp. 91 – 118.
[7] M. Montali and A. Rivkin. “DB-Nets: On the Marriage of Colored Petri Nets and Relational Databases.” ResearchGate. https://www.researchgate.net/publication/310122815_DB-Nets_on_The_Marriage_of_Colored_Petri_Nets_and_Relational_Databases (accessed Jan. 25, 2020).
[8] “Laboratory of Process-Aware Information Systems (PAIS Lab) — HSE University”. HSE University. https://pais.hse.ru/en/ (accessed Jan. 25).
[9] “SQLite Home Page.” SQLite.org. https://www.sqlite.org/ (accessed Jan. 25, 2020).
[10] P. Bourque and R.E. Fairley, Eds., “Software Life Cycle Models,” in *Guide to the Software Engineering Body of Knowledge, Version 3.0*, USA: IEEE Computer Society, 2014, ch. 8, sec. 2.2, pp. 8-5–8-6.
[11] “Project Management Software | Microsoft Project.” Microsoft Office | Productivity Tools for Home & Office. https://products.office.com/project (accessed Jan. 25, 2020).