**Project Proposal**

Master’s Thesis (1st year)

Software and System Engineering Programme

School of Software Engineering / FCS

**Full Name: Anton Rigin**

**Student ID – М191МСПИН019**

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

Изображение выглядит как объект

Автоматически созданное описание

Supervisor: Sergey Shershakov  
Senior Lecturer at School of Software Engineering,  
Research Fellow at PAIS Lab,  
HSE Faculty of Computer Science

**Date of Proposal’s Submission: XX January, 2020**

**Moscow, 2020**

# Abstract

Mention all of main points you’ll be looking at; follow recommendations that were given in ‘Research seminar’ classes (essential parts of the Abstract must be present, viz. *Background*, *Purpose* (*Aim(s)*), *Approach (Methods)*, *Conclusion (Expected Findings/Results* can be mentioned aswell)) – Times New Roman or similar font is used, 12 points, single spaced.

Abstarct is formed as a single paragraph text.

**Key Words:** Give about 5-7 keywords that allow to characterize (position) your project (research) easily – try to choose those keywords that will make others interested in the contents of the study (proposal); do not write straight away the words (phrases) that suddenly pop up in your mind! Experiment, for example, with Keyword Tool / <https://keywordtool.io/> or similar services.

Separate your keywords (phrases) with a semicolon ;

Times New Roman or similar font is used, 12 points, single spacеd.

The length of the Abstract section together with a list of keywords (one empty line separates them as shown above) is appr. 250-350 words (0.6 – 0.9 page) – please, refer to services to convert specified number of words to appr. number of pages – e.g. <https://wordcounter.net/words-per-page>.

The text is aligned at both sides of the page (the same requirement is valid for all following sections of the document).

# Table of Contents

Abstract ……………………………………. page 2

Introduction ……………………………………. page 4

Thesis (Research) Statement ……………………………………. page 6

Methodology (Approach / Methods) ……………………………………. page 7

Preliminary (Expected) Results and Discussion ……………………. page 9

Work Plan (as seen in December-January 2020) ……………………. page 10

Implications of the Project’s Work ……………………. page 11

Bibliography (References) ……………………………………. page 12

# 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.

Изображение выглядит как снимок экрана, часы

Автоматически созданное описание

*Fig. 1. The db-net structure [6].*

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 Faculty of Computer Science (the PAIS Lab) [7]. 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 next year.

This section provides the context for the proposed project (research), explains the background of the research work (from a broad sight down to more narrow one; personal motivation must be mentioned as well). The length of the Introduction section is about 1 to 2 pages – if more, do not exceed 2.5 pages under any circumstances!).

The Introduction section must provide a solid base for a reader to undestand the background of the proposed project.

The indent of the first line of each paragraph is 4 blanks (consider ‘Tab’ to be 4 blanks).

Each section in the document starts with a new page!

The titles of document’s sections as they are shown now must not be altered!

# 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 June 2020.

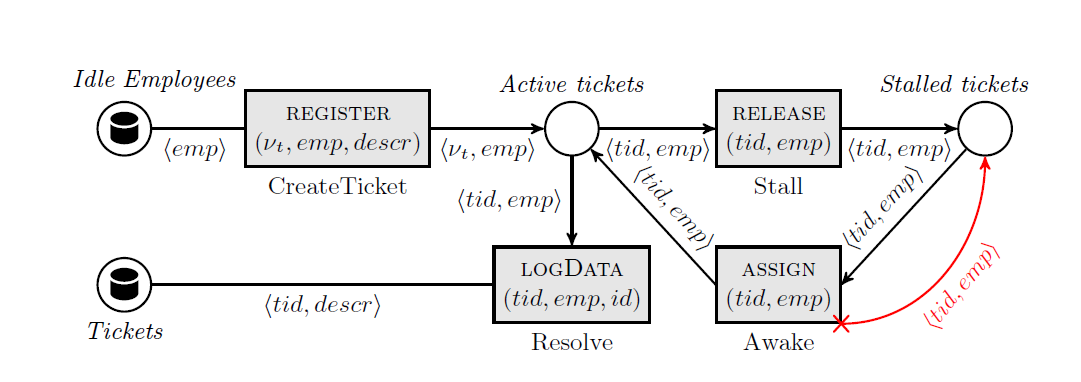
The material in (contents of) this section should flow naturally from what is described in the previous section. Be precise and convincing – the size of the section must be up to 0.5 page long; the statement must be absolutely coherent, all necessary details, despite the rather limited length of the section, should be presented in an unambiguous manner… Think about it carefully – now it is a right time to formulate your project’s statement.

Details were discussed during regular hours of the Research seminar.

# 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 [8] 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 [9].

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.

The material in (contents of) this section contains the overall description of approach(-es), methods, procedures, etc. that you are planning to use in the project (research study). The size of the section is between 1 and 2 pages long. All views (statements, explanations) should be clear and convincing.

# 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 goal, 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 Faculty of Computer Science (the PAIS Lab) [7] 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.

Details were discussed during regular hours of the Research seminar. The presentation of ideas must be concise. Thus, try to keep this section short (appr. 200-300 words) – in any case, if a detailed explanation is required (this is not prohibited, feel free to provide all relevant details, the ones “in essense”), you can't be beyond 1 (one) page.

# 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.

<…>

*Fig. 3. The project’s Gantt diagram (chart).*

Gantt diagram (chart) is the handiest form (<https://www.gantt.com/>) of presentation of the planned works’ stages in the project (research). In addition, it is possible to give a brief explanation (comments) of the individual aspects presented in this section.

# Implications of the Project’s Work

Say about new knowledge, what is worth knowing with regard to the work to be done... The size of the section is expected to be about 0.5 - 0.6 page long.

# Bibliography (References)

[1] Scholarpedia

[2] Reisig

[3] Jensen

[4] Jensen and Kristensen

[5] Renew

[6] DB-Nets

[7] PAIS Lab

[8] SQLite

[9] Something about the incremental software development

List all sources (papers, articles, books, journals, documents, reliable Internet-sources, professional resources, training materials (of conferences, master-classes, etc.), …) in the alphabetical order. Use either APA or IEEE citation styles (depends solely on your preferences).

Make sure that all references are properly formatted, and they are all used in the text (aforesaid sections) of the proposal (no "dangling" references)…

Indents of lines are not used.

Please, don't forget that everything highlighted within the template in yellow color should disappear in the final version of the document and be replaced with the real text.

The mandatory font color (of the text) is black.