Lappeenranta-Lahti University of Technology LUT
School of Energy Systems
Master's Programme in Electrical Engineering

Maksim Kuzmin

THE TITLE OF THE WORK - MULTI-AGENT SYSTEM FOR CONTROL OF A GROUP OF REACTIVE POWER COMPENSATORS.

Examiners: Professor Jarmo Partanen

Supervisors: Professor Jarmo Partanen
PhD Aleksandr Voloshin

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Energy Systems
Master's Programme in Electrical Engineering

Maksim Kuzmin

Title of the work - Multi-agent system for control of a group of reactive power compensators.
Master's Thesis

90 pages, 43 figures, 2 tables, 5 appendices

Examiners: Professor Jarmo Partanen

Keywords: Multi-Agent System, reactive power compensator, voltage, agent.

Nowadays there are different ways of controlling a group of reactive power compensators at substation but most of them tend to be deterministic and rigid. In other words, they do not take into account the possibility of a change in the number of compensators operating at a given moment due to maintenance or switching of circuit breakers which alters configuration of the substation thus affecting the number of compensators electrically connected to the busbar. Also, the current control systems of the group of compensators do not take into account the current state of each individual device with regards to the circuit breaker's on-off cycles which may lead to more frequent maintenance services. Furthermore, a decrease in the power losses in the compensators and minimization of harmonic input into the system from compensators such as variable shunt reactors are desirable as well.

These flaws of the regular control systems can be eliminated by using a multi-agent system technology. Therefore, in this thesis there will be developed a control system of a group of

compensators using a multi-agent system that is adaptive to any changes in the substation and also deployment of which leads to an increase in the lifetime of the circuit breakers, and decrease in the power losses in the compensators, and minimization of harmonic input into the system by the compensators.

ACKNOWLEDGEMENTS

Throughout the writing of this dissertation I have received a great deal of support and assistance. I would first like to thank my supervisors, Professor Jarmo Partanen, LUT, Lappeenranta, Finland and Aleksandr Voloshin, MPEI, Moscow, Russia whose expertise was invaluable in the formulating of the research topic and methodology in particular and helping me throughout my entire work. I would also like to thank my parents for their immense support throughout my studying.

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

CB	Circuit Breaker
FC	Fixed capacitor
FSM	Finite state machine
JADE	Java Agent Development framework
MAS	Multi-Agent System
MCSR	Magnetically controlled shunt reactor
MSC	Mechanically switched capacitor
RTDS	Real-Time Digital Simulator
SR	Shunt Reactor
SVC	Static VAR Compensator
TCR	Thyristor Controlled Reactor
THD	Total harmonic distortion
VSR	Variable Shunt Reactor

INTRODUCTION

1.1 Voltage control

Voltage regulation is a set of technical measures for limiting voltage deviations from their nominal values at network nodes within acceptable limits.

There are three reasons why it is necessary to manage reactive power and control voltage.

First, both customer and power-system equipment are designed to operate within a range of voltages. Usually it is within ±5% of the nominal voltage. At low voltages, many types of equipment perform poorly. For instance, light bulbs provide less illumination, induction motors can overheat and be damaged, and some electronic equipment will not operate at all while high voltages can damage equipment and shorten their lifetimes.

Second, to maximize the amount of active power that can be transferred across a congested transmission line, reactive-power flows should be minimized. Thus, to make it happen the voltages must be controlled. Similarly, reactive-power production can limit a generator's active-power capability.

Third, transferring reactive power through the transmission system incurs active-power losses, which is why it is economical to supply this reactive power closer to the load in the distribution system. Thus, to minimize active power losses, voltage needs to be kept within the limits.

1.2 Means of voltage control

1.2.1 Shunt Reactor

The shunt reactor (SR) is one of the most economical ways to compensate reactive power generated by long high-voltage power transmission lines in the transmission system due to the capacitance between the lines and earth.

The reactor consumes reactive power, which is proportional to the square of voltage U:

$$Q_R = b_R \cdot U^2 , \tag{1}$$

where b_R - inductive conductance, U - voltage.

The shunt reactor can be directly connected to the power line or to a tertiary winding of a three-winding transformer. The connection is usually carried out through circuit breakers. In this case, shunt reactors considered as passive as they either absorb reactive power when the breaker is closed or not when the breaker is opened and it does not offer any variable control.

1.2.2 Variable Shunt Reactor

Unlike the shunt reactors discussed above, variable shunt reactors (VSR) allow adjustment of the amount of reactive power being absorbed, which in turn allows more effective voltage control. Furthermore, it increases transmission line use efficiency for the power transmitted over the line in the range from no load to a natural loading of the transmission line. Shunt reactors allow only 40-50 % of a natural loading to be transmitted over the line because of unacceptable voltage decrease. At the same time, with daily variations in transmitted power, frequent closing and opening of conventional SRs is problematic due to the limited number of on-off cycles of high voltage circuit breakers.

There are several different implementation designs of VSR some of which are discussed below.

Thyristor Controlled Reactor

Variable shunt reactor based on thyristor controlled reactors is shown in the Fig. 1. It includes a group of single-phase transformers or a three-phase transformer T, the secondary windings of which with a voltage of 10, 20, 35 kV are connected in a delta arrangement via a permanently switched-on circuit breaker Q_g, and several parallel sections (modules) of single TCRs consisting of bidirectional thyristor valves VS1 and reactors LR1.

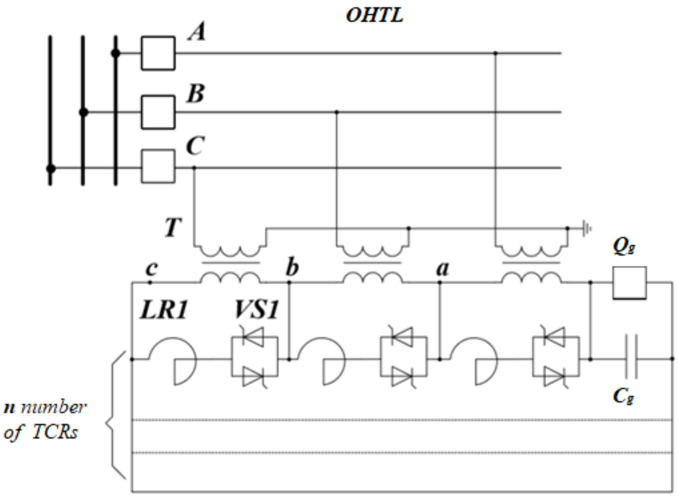


Fig. 1. Thyristor Controlled VSR

By changing the firing delay angle β of thyristors valves LR1 the value of a current flowing through the reactor LR1 is changed from maximum to zero. Firing angle β varies from 90 degrees to 180. At the angle β=90° the current is fully sinusoidal with no harmonics, while at β=180° the current reduces to zero. At the angles between 90 and 180 degrees there are current waveform distortion which include odd-order harmonic. However, tripled harmonics are trapped inside delta thus they are not injected into the power system. The 5th and 7th harmonics and to a lesser extent 11th, 13th, 17th etc. should be filtered in order to reduce total harmonic distortion.

Vacuum circuit breaker based VSR

This type of a VSR system is a simplified version of the one that was describe above. The simplification is done by replacing thyristor valves with vacuum circuit breakers. The circuit scheme of the VSR is show in the Fig. 2.

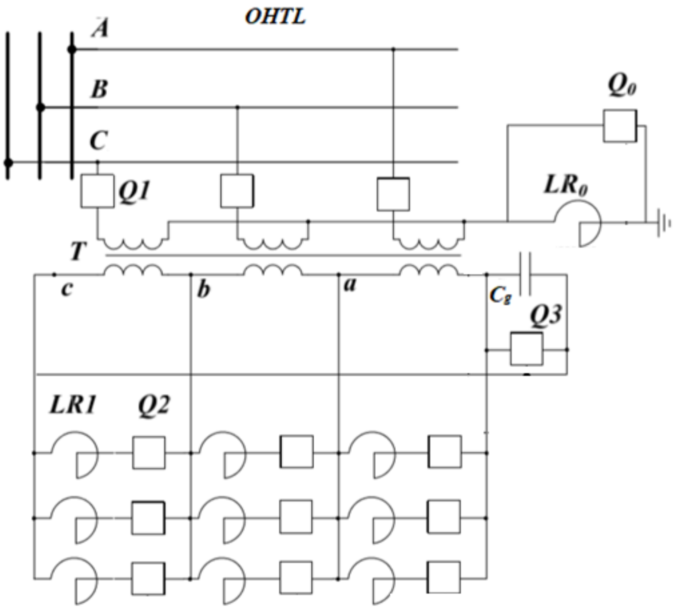


Fig. 2. Vacuum circuit breaker based VSR

There may be several parallel sections (modules) of reactors LR1 along with its circuit breaker Q2.

The amount of reactive power being absorbed is determined by the amount of sections that are currently switched on. Circuit breakers used for these purposes have a switching resource in terms of on-off cycles of more than 20,000 operations.

Magnetically Controlled Shunt Reactor

Circuit diagram of magnetically controlled shunt reactor is shown in Fig. 3.

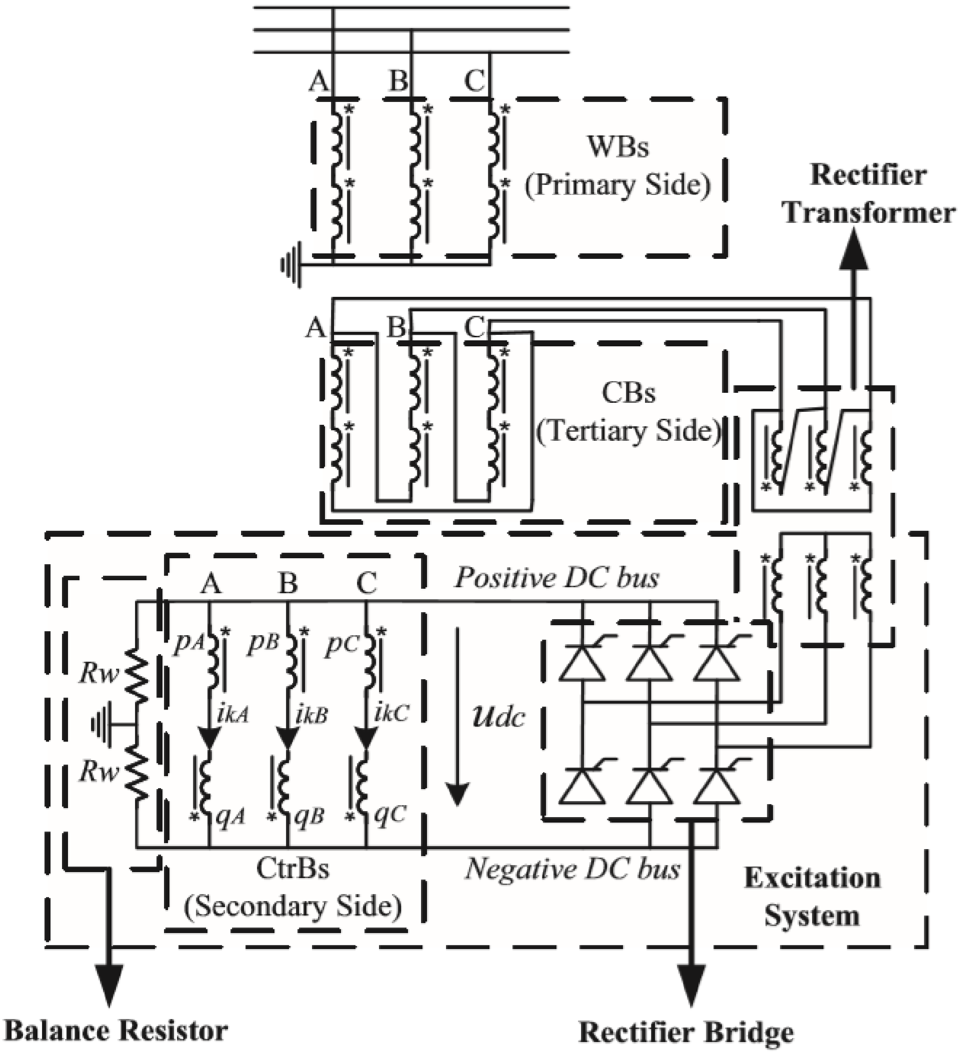


Fig. 3. Circuit diagram of magnetically controlled shunt reactor. [1]

A MCSR is a semiconductor switch transformer device, based on the principle of alternate high saturation of each magnetic core [2]. Magnetically controlled shunt reactor consists of a AC working windings and DC control windings that are placed on two solid cores. When a regulated DC voltage is applied a DC magnetic flux appears in the core which makes the resulting magnetic flux shift to the saturation domain of the MCSR core. This saturation leads to an inductive current in the working windings. When the DC control current increases, the working current also increases. Thus, by changing the magnitude of DC voltage applied to the control winding the reactive power consumed by the MCSR can be changed.

The DC voltage is fed from the compensating winding of the MCSR after rectifying AC voltage by a thyristor convertor.

1.2.3 Static VAR Compensators (SVC)

Static VAR Compensators are meant for smoothly varying generation or consumption of reactive power.

SVCs are usually deployed for the following reasons [3]:

- Voltage regulation.
- Increase power transfer capacity of transmission systems.
- Increase transient stability limits of a power system.
- Increase damping of power oscillations.
- Reduce overvoltages.
- Damp subsynchronous oscillations.

There are different versions of SVC depending on the combination of different compensating equipment. Some of the widely used are:

- TCR with fixed capacitors (FC).
- TCR with mechanically switched capacitors (MSC).
- TCR with thyristor switched capacitors (TSC).

Fig. 4-6 show TCR with FC, TCR with MSC, and TCR with TSC respectively.

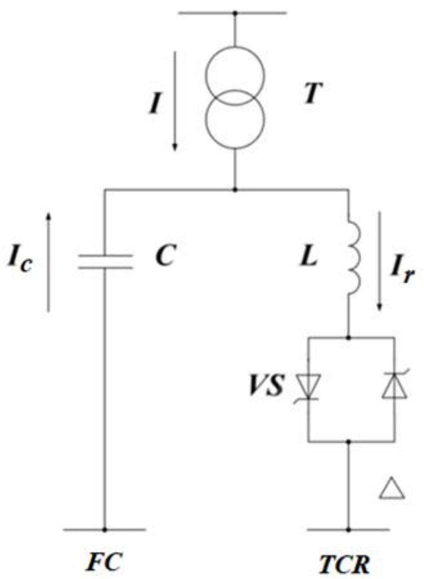


Fig. 4. TCR with FC.

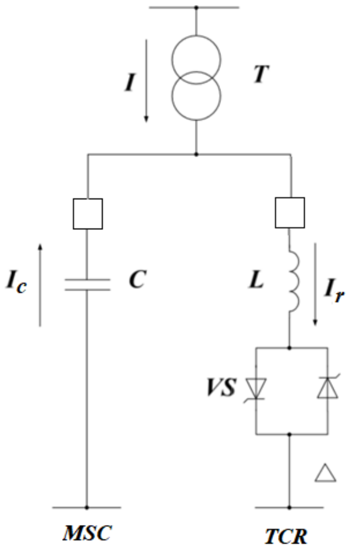


Fig. 5. TCR with MSC.

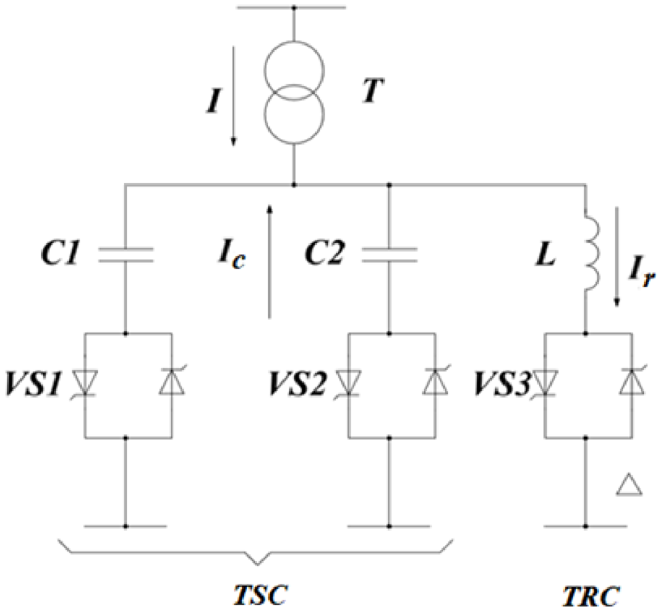


Fig. 6. TCR with TSC.

In the SVCs harmonic filters are used to suppress high harmonics so as to prevent adverse effects of them.

1.2.4 Static Compensator (STATCOM)

A STATCOM (Static Compensator) which is shown in Fig. 7 is a power electronic device to provide reactive power that uses insulated gate bipolar transistors (IGBT) as a voltage source converter (VSC) that coverts DC voltage from DC source which is a capacitor into AC.

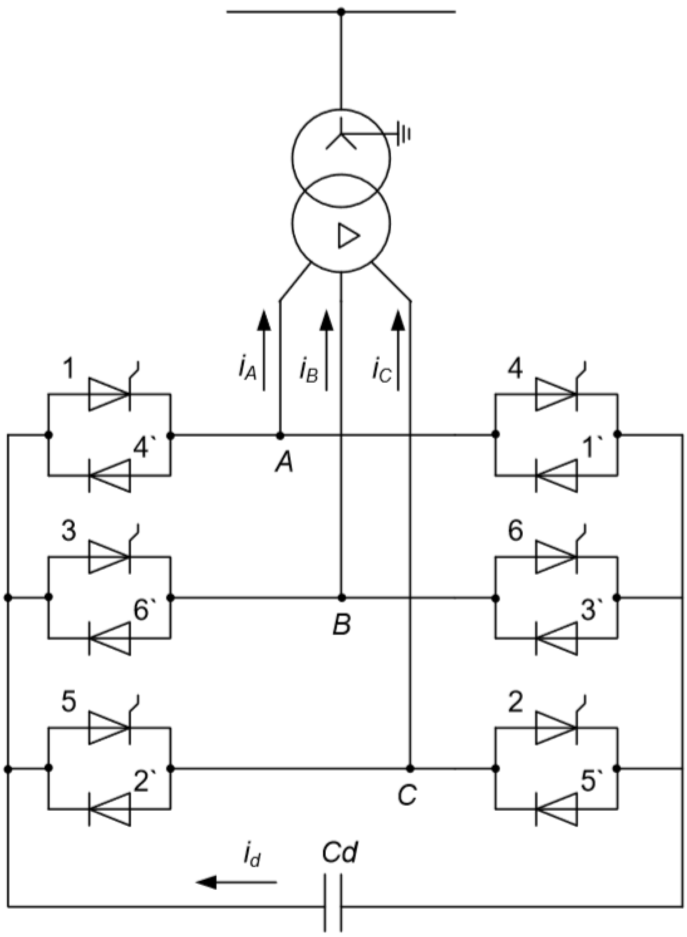


Fig. 7. Static Compensator.

IGBT transistors can be turned on and off to synthesize a voltage sine wave of any magnitude and phase. If the magnitude of synthesized voltage is greater than AC grid voltage, then reactive power is generated by the STATCOM while if the magnitude of synthesized voltage is less than that of a AC bus then reactive power is absorbed by the STATCOM.

The reactor consumes reactive power, which is proportional to the square of voltage U:

$$P = \frac{U_C \cdot U_S}{X} \cdot \sin\beta , \tag{2}$$

$$Q = \frac{U_S (U_S - U_C \cos\beta)}{X} , \tag{3}$$

where U_C - fundamental component of the voltage at the terminals of the VSC, U_S - AC grid voltage, β - phase difference between two voltages, X - coupling reactance.

Unlike the SVC, the reactive power that can be injected into the AC grid does not depend on the AC voltage, since the reactive power is not produced by means of capacitors and inductors, but by a synthetic voltage source defined by the DC bus [3]. That is why STATCOM can supply rated reactive power even at low AC grid voltages.

1.2.5 Synchronous Compensator

A synchronous compensator is a synchronous motor running without a mechanical load and, depending on the value of excitation, it can absorb or generate reactive power [4].

Reactive power is controlled by means of controlling DC excitation current. When synchronous compensator is run overexcited it generates reactive power and when it is run underexcited it absorbs reactive power. Usually due to its design synchronous compensator can not absorb more reactive power than it can generate.

Since all of the synchronous compensators are of a salient pole, reactive power output is determined by the following formula:

$$Q = \frac{U_B E_q}{X_d}\cos\delta + \frac{U_B^2}{2}\left(\frac{1}{X_q} - \frac{1}{X_d}\right)\cos 2\delta - \frac{U_B^2}{2}\left(\frac{1}{X_q} + \frac{1}{X_d}\right), \tag{4}$$

where U_B - terminal voltage, E_q - no load voltage, $\cos\delta$ - load factor, X_q - reactance in q axis, X_d - reactance in d axis.

The main purposes of these compensators are increasing power factors and controlling the voltage at substation busbar. Above all since synchronous compensator is a rotating machine it increases inertia of the power system and it makes riding through transients disturbances easier [4].

1.2.6 Tap-Changing Transformers

Some power transformers are equipped with tap changer which allows changing the ratio of between nominal voltages. These transformers have a number of taps on one of the windings switching and by switching between them the ration can be changed. Thus this way the voltage can be controlled.

The taps as shown in Fig. 8. are normally placed on the high-voltage side of the transformer since on the higher side there are lower currents.

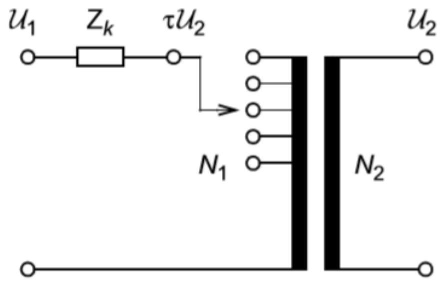


Fig. 8. Transformer with tap changer [5].

Tap changers can be either on-load or off-load. The latter type of tap changer does not allow switching taps while operation of the transformer while former does and can be used to control voltage during day voltage variations.

1.3 Reactive power compensators control at substations

At the moment, the voltage regulation at the substation is set up as follows. Control over shunt reactors is most often performed manually by the operational staff at the command of the system operator. But it is also possible to use relay characteristics when switching on or off occurs when the setpoint is exceeded or went below the setpoint, respectively.

As for compensators that can adjust their output based on the current voltage at the busbar like VSR, there are different principles of management of the group of such compensators installed at the substation. One of the possible control options, for instance, is when a group of VSR are loaded or unloaded at the same time in unison upon receiving a signal with a total amount if reactive power that needs to be absorbed which is then divided by the amount of VSRs and distributed across them. Another option is when each individual VSR has its own regulator that controls the output of VSR independently of other VSR. In addition, there is also a possibility of controlling a group of VSRs on the principle of master-slave.

There might be slightly different approaches of controlling a group of VSRs at substation but most of them tend to be deterministic and rigid. In other words, they do not take into account the possibility of a change in the number of VSRs operating. For instance, should one of the VSR be taken out for a maintenance the whole system needs to be stalled and reconfigured. Change in the number of available VSRs and other compensators can happen also down to the changes in the state of circuit breakers (CB) at the substation which can alter configuration of the substation thus affecting the number of compensators electrically connected to the busbar. There also could be new compensators installed later on which would lead to changes of the control system of the group of compensators again.

Also the current control systems of the group of compensators do not take into account the current state of each individual device. For instance, a circuit breaker that links shunt reactor to a bus or a tertiary winding of a transformer has its on-off cycle limits. Therefore, to keep the shunt reactor operable for longer it is recommendable to avoid unnecessary CB operation or prioritize operation of the shunt reactor, CB of which is less “worn down”.

What also is not taken into account is the power losses in the compensators. It would be better to distribute reactive power that needs to be compensated so there’s less active power losses.

The last but not the least, minimization of harmonic input into the system from VSRs which is also not taken into account.

1.4 Thesis objectives

The aim of the thesis is to develop a system of coordinated automatic control of reactive power compensators using multi-agent system (MAS) to maintain a given voltage level on the busbar of the substation that meets the requirements below:

- Reducing power losses and improving its quality.
- Even use of device’s resources with regards to a CB on-off cycles thus increase in their lifespans.
- Adaptability to changes in the substation configuration diagram and in the number operating compensators at any given time.

1.5 Control object description

As a control object for the system to be developed there will be group of reactive power compensators installed at the real substation “Novoanzherskaya 500/220/110/10 kV” which is a part of the Siberia Power Grid.

Siberia Power Grid map is illustrated in the Fig. 9.

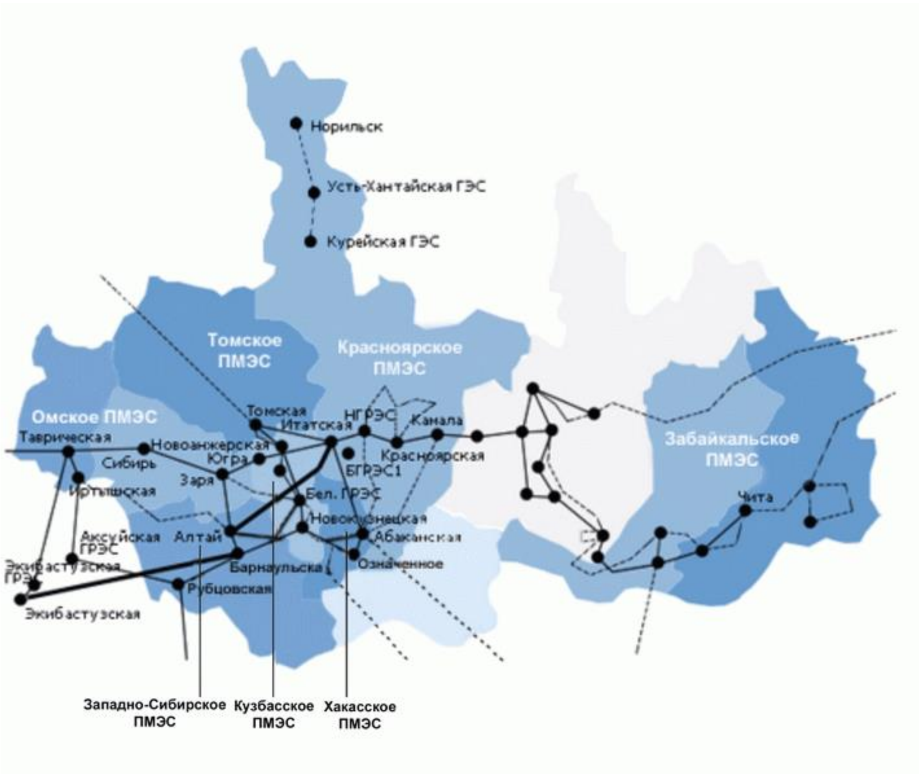


Fig. 9. Siberia Power Grid.

Substation’s single-line diagram is shown in the Fig. 10.

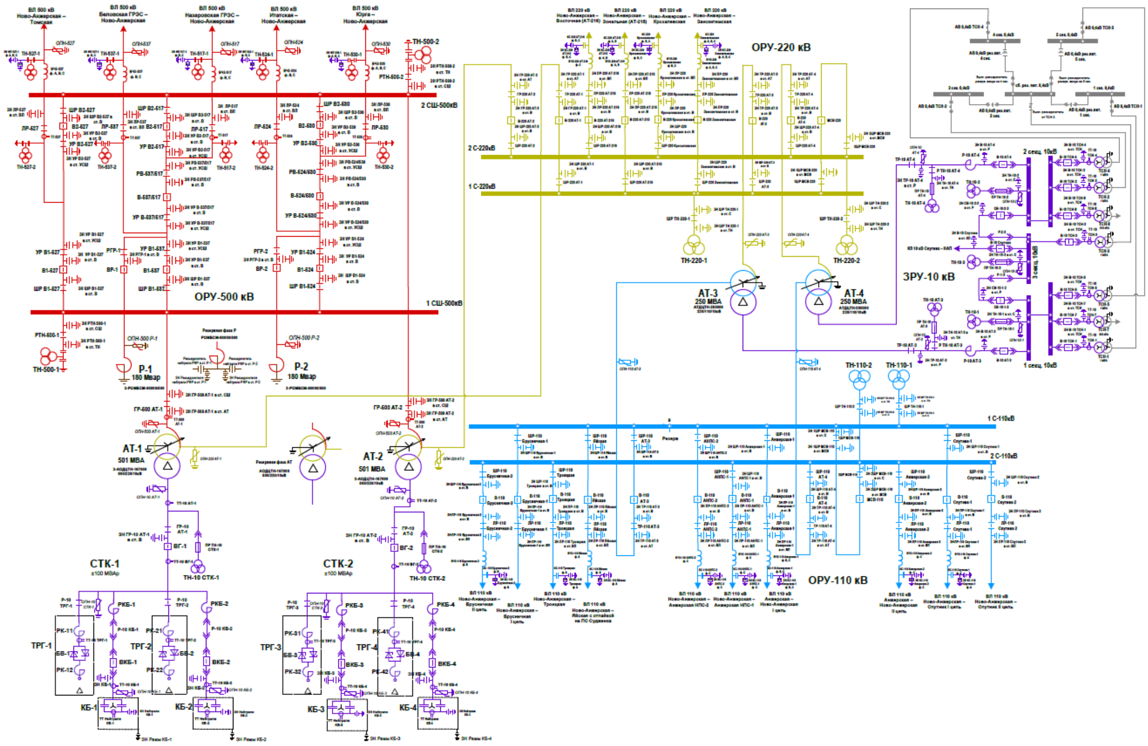


Fig. 10. “Novoanzherskaya 500/220/110/10 kV” substation single-line diagram.

The list of compensators installed at the substation the control of which will be done by the multi-agent system under development is presented in the Table 1.

Table 1. Compesators at the “Novoanzherskaya 500/220/110/10 kV”.

Type of device	Device’s name
Shunt Reactor	R-1
Shunt Reactor	R-2
SVC (TCR with MSC)	STK-1
SVC (TCR with MSC)	STK-2

1.6 Definition of MAS and its use justification

An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives [6]. An environment in that sense is just everything that is external to an agent. By the agent’s

autonomy it is meant that an agent can initiate and schedule its own actions which is usually prompted by the changes in the environment.
A multi-agent system is a system consisting of two or more agents [7]. A system of agents does not have an overall goal. Instead, each individual agent has its own local goals. And a set of separate agent's goals constitute a MAS designer's overall intention.

MAS in general has three very important characteristics that makes it a very promising and beneficial approach to use when controlling a group of compensators. MAS offer flexibility, extensibility and fault tolerance [7].

Starting off with flexibility. Flexibility basically means the ability of a system to operate well under dynamically changing environment or conditions. That with respect to reactive power compensators control would mean that MAS would be able to withstand the changes in the single-line diagram after opening or closing a certain CB which could affect the compensators connected to the busbar. Also, should the number of compensators change due to taking any equipment out of the system for maintenance MAS having flexibility would be able to cope with that as well.

Extensibility connotes the possibility to change existing functionality of a system. Indeed, since each individual agent usually encapsulate a very limited number of goals and has its own set of actions that it takes as a response to changes in the environment or to some situations like receiving messages from other agents, these functions of an agent or the way an agent reasons about something or behaves in certain situations can be altered independently to other agents in the MAS. Thus, when there's a need to make some changes to an agent's behavior, only this agent needs to reconfigured. Or if there's a need to include a new functionality into the MAS there can always be added new agents that possess that functionality into the system while other agents are running uninterrupted

When it comes to a fault tolerance, again MAS benefits off the idea of an independent from each other agents i.e. should some agent lose the ability to perform its functions for some reason there will always be other agents capable to take over and help out.

For these reasons MAS technology would pose a very strong tool for controlling a group of compensators at substation.

Besides, MAS has already been successfully integrated into power engineering domain for monitoring and diagnostics, distributed control, modeling and simulation, and protection.

2 THESIS

2.1 Identification of a required set of agents for this project

To implement this system, which meets the requirements outlined earlier, the following types of agents are to be used:

- Bus Agent. Bus Agent is a type of agent that represents a busbur at the substation. Each busbar has its own Bus Agent. In this case there will be a Bus Agent for the busbar “1SSh-500kV” and for a busbar “2SSh-500kV”. The names are in accordance with the single-line diagram.
- Circuit Breaker Agent. Each CB is represented by its own circuit breaker agent.
- Shunt Reactor Agent that represents a shunt reactor. There will be two Shunt Reactor Agents for the shunt reactor “R-1” and “R-2”. The names are in accordance with the single-line diagram.
- TCR Agent + MSC Agent. In this work a SVC is of a type that consists of two TCRs and of two MSCs. Therefore, it has been decided that for the SVC “STK-1” there would be two TCR agents each representing TCR “TRG-1” and TCR “TRG-2” and also two MSC agents each representing MSC “KB-1” and MSC “KB-2”. The same is applied for SVC “STK-2” that has two TCR agents for “TRG-3” and “TRG-4” and two MSC agents for “KB-3” and “KB-4”.
- Power Transformer Agent. Power transformers “AT-1” and “AT-2” are represented by the agents of this type.
- Telemetry and Telecommands Agent. This agent is used for testing the system and is meant to transfer data and commands to the running model of a substation and from it. The essence of this agent will be explained in more details in further sections.

The TCR Agents, MSC Agents, and Shunt Rector agents from now on can be referred to with more generic term as Reactive Power Compensator Agents (RPC Agents) since they will all have pretty much the same functionality and behaviors.

The primary functions for each type of an agent is given in the Table 2.

Table 2. Each type of agent primary functions.

Agent type	Primary Functions

Bus Agent	1) Votage control on the bus represented by this agent. 2) Command distribution across reactive power compensators agents.
Circuit Breaker Agent	1) Data acquisition with respect to the CB state (whether it is closed or opened). 2) Control over its own CB.
Power Transformer Agent	In this work this type of an agent will not posses any distinctive functionalities apart from taking part in the all-round agents collaboration in the MAS.
RPC Agent	1) Data acquisition with respect to the current reactive power absorbed or generated depending on the type of a device and its operation mode. 2) Control of the reactive power output (compensated reactive power).
Telemetry and Telecommands Agent	1) Data acquisition from the model. 2) Commands transfer from MAS to the model.

2.2 Outline of the agents' behaviours

2.2.1 Bus Agent

The main goal of the Bus Agent is to maintain voltage on its bus by the distribution of the commands across all available reactive power compensators. Here commands mean reactive power that needs to be compensated by the compensator.

This agent has a main behavior that runs cyclically every 200-500 ms and controls voltage on its busbar. Block diagram of this behavior is show in the Fig. 11.

In the very first cycle, all RPC agents present on the substation should be determined by looking up in the register that is shared among the members of the MAS.

Next, the topology of the electrical connections of the substation is checked for its invariance with respect to the previous cycle. If there is a change in the topology of substation, for instance, a CB has been opened, a search is done for all RPC Agents that have an electrical connection to the busbar. This action is always performed on the very first cycle. In case of no change in the topology, reactive power that needs to compensated is calculated by the PID or PI regulator.

Then the difference between the reactive power that is currently compensated by all compensators connected to the busbar in total and the new value that has just been calculated is distributed across different compensators by means of bidding process made by all available RPC Agents. After bidding process was completed, Bus Agent has a list of RPC Agents that have won and that are going to compensate the difference in reactive power explained earlier.

After that, Bus Agent requests the winning RPC Agents to confirm the readiness to perform the actions declared in the bid made during the bidding process with relation to reactive power compensation. For example, if the from

For example, if two RPC agents (agent SR and agent TCR) were selected as winning based on the results of the bidding process, their declared actions could be: closing the CB of SR for the SR Agent and unloading the TCR by 30 % for TCR Agent.

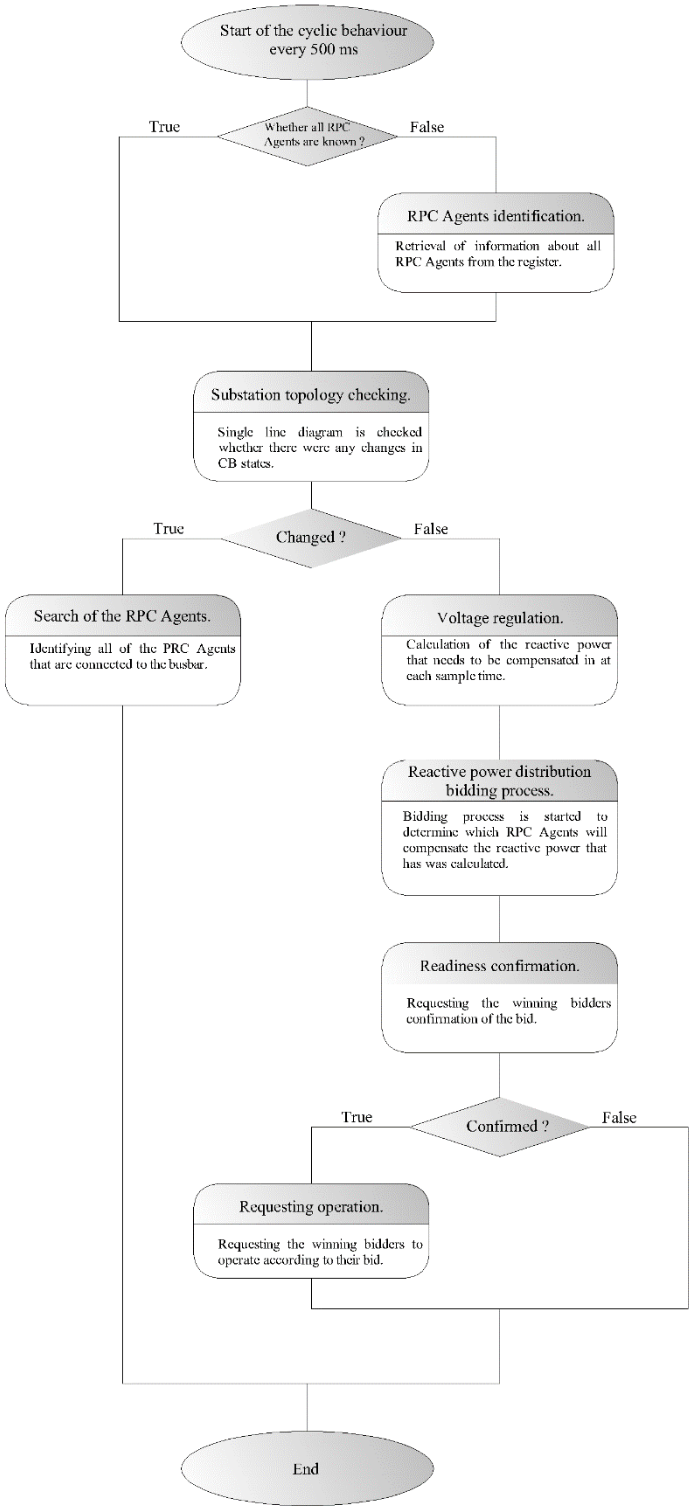


Fig. 11. Main cyclic behavior of the Bus Agent.

Next, in case of a successful confirmation, Bus Agent requests the winning RPC Agents to carry out declared in the bid actions. Bus Agent receives the results of their operation as to whether the action was completed or not for some reason.

The considered cyclic behavior will be executed during the entire life time of the Bus Agent.

2.2.2 CB Agent

The main behavior of CB Agents is to periodically check their “mailbox" for new messages from other agents.

Upon receiving a message, the CB Agent analyzes its content and based on the content of the message, either performs an action or responds to the queries. CB Agent can also send requests for some actions to other agents, but this can only happen as a reaction to the received request.

Also, the agent may not understand the content of the received message, then this message will not be processed and the message will remain in the “mailbox”. Especially for such cases, additional cyclic behavior is provided that deletes all this kind of messages.

2.2.3 RPC Agent

Like CB Agent, RPC Agent periodically checks its ‘mailbox” for new messages from other agents and responds to some queries or requests. This agent also has a behavior that removes the messages that the agent could not understand.

What makes RPC Agent different from CB Agent in terms of behavior is the ability to respond to specific requests that are not meant for CB and only make sense for a RPC Agents and vice versa. For example, you cannot request increase of the load by 70% from the CB, because a CB agent simply would not understand what is wanted from it. Therefore, such a message will remain unprocessed in the “mailbox”, but will be deleted after some time by the behavior described above.

2.2.4 Power Transformer Agent

Like CB Agent and RPC Agent, this agent periodically checks its ‘mailbox” for new messages from other agents and responds to some queries or requests. This agent also has a behavior that removes the messages that the agent could not understand.

2.2.5 Identification of all connected to the busbar RPC Agents

Identification of all RPC Agents whose RPCs are connected to the busbar at a given time is done by a sequential flow of requests from the Bus Agent to the RPC Agents via all agents whose equipment is physically in between.

A simplified diagram of this process is depicted in Fig. 12.

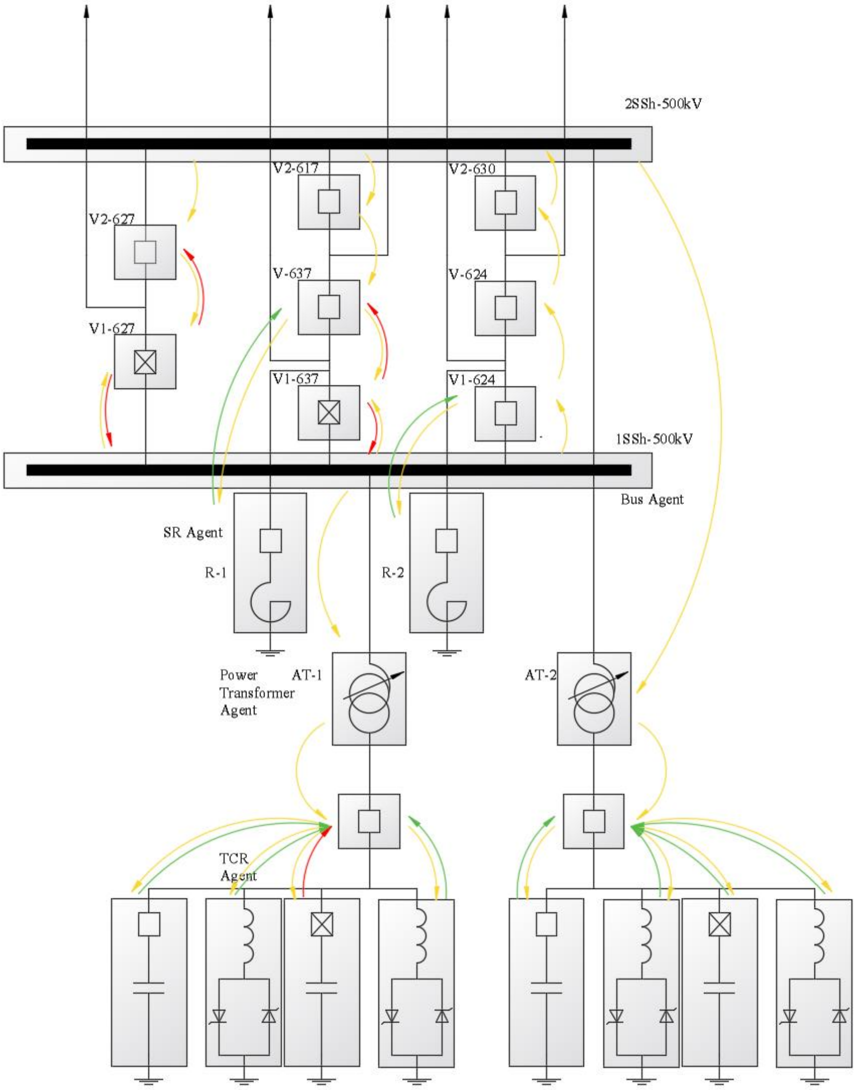


Fig. 12. Identification of all connected to the busbar RPC Agents.

After having carried out the identification of all RPC Agents at the substation regardless of it being connected or not to the bus which is done by looking up in the register where all of the RPC Agents are registered after initialization, Bus Agent knows which RPC Agents it should determine are connected to it. Bus Agent forms a list of these RPC Agents.

Bus Agent sends requests to adjacent agents to find all those RPC Agents. Agent's adjacent agents connotes agents whose equipment is adjacent to this agent's equipment. Referring to diagram in the Fig. 12, a Bus Agent of the bus "1SSh-500kV" will send requests to the CB Agents in "V1-627", "V1-637", "V1-624" and also to the Power Transformer Agent "AT-1" asking them to find RPC Agents from the list formed earlier.

When an agent receives this kind of a requests, it does the following:

1. Checks if it is the one being searched. If so, it forms a positive response saying that the agent is being searched has been found, which ultimately means it is connected to the busbar.
2. If the previous condition is not true, then it checks whether its equipment is in operation. In case of a CB it checks whether its CB is under maintenance or not. If it is then it sends a negative response meaning that agents that are being searched cannot be connected via this agent.
3. If the previous condition is true, then an agent would try and forward this request further to its adjacent agents to see if any of them are the ones being looked for and they are in operation. But before doing so it checks should it pass this request further would the message enter in a loop or not. In other words, would the message be forwarded to the agent that had already forwarded this very message at some point before. Every sent message includes a list of agents who this request has passed through. So an agent basically checks the presence of any of its adjacent agents in this list. If it detects a message is going to enter a loop, then it sends back a negative response. If not, it forwards this message further including itself in the list and waits for a response.
4. If an agent receives any positive response from any of it adjacent agents, then it responds to an agent that it received a request from in the first place with a positive message, which means that at least one of the agents from the searched ones has been found and it is connected to the busbar.

Referring to the CB Agent of a "V1-624" as an example, when it receives a request it sends it further to its adjacent agents (SH Agent "R-2" and CB Agent "V-624") since this agent is neither an agent being searched nor it is under maintenance. Then it waits for responses from all adjacent agents.

SH Agent “R-2” will send a positive response since it is a RPC Agents which is among those being searched. As for CB Agent “V-624”, it will forward this requests further.

The same request would be forwarded until it reaches an RPC Agent or it would be stopped due to some equipment being under maintenance or with respect to the CB it is open and cannot be closed, or should this message be forwarded any further it enters in a loop.

Should any agent receive several responses that have reached the same compensator, the one whose path is shorter is taken as final result and returned to the sender of the request for this agent as a response.

In this manner, Bus Agent gains a knowledge about which compensators are connected to its busbar.

2.2.6 Reactive power distribution bidding process

When a Bus Agent knows which compensators are connected to its busbar and the difference in the reactive power that is currently compensated needs to be made ΔQ, it now should decide how to distribute this ΔQ between those compensators that are connected to the busbar. This distribution is done by means of bidding process.

For explanation reasons of the devised bidding process system and the logic behind it there will be considered a bidding process for a group of compensators connected to the busbar. The following compensators and their respective agents are considered:

- SR Agent – “R-1”
- SR Agent – “R-2”
- SVC Agent – “SVC-1”
- SVC Agent – “SVC-2”

In this example, ΔQ has a negative value which means compensators should increase their reactive power consumption by ΔQ and this value is within the limits of the maximum possible consumption value of the reactive power for both SVCs.

Diagram of the bidding process for the example in question is depicted in the Fig. 13.

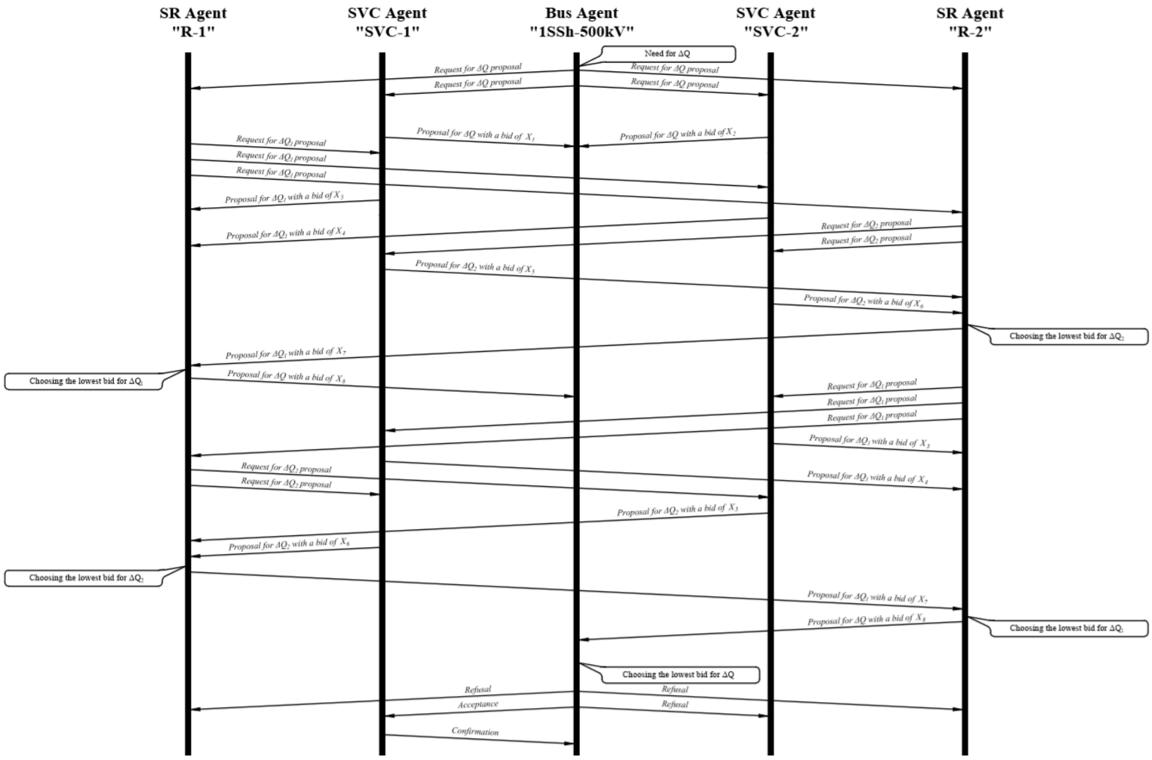


Fig. 13. Reactive power distribution bidding process.

The Bus Agent sends simultaneously requests to all connected to the busbar RPC Agents. In this specific example, agents “R-1”, “R-2”, “SVC-1”, “SVC -2”. Furthermore, every request contains an information about all available RPC agents and about the value of ΔQ necessary for compensation.

After receiving requests for a ΔQ, SVC Agents such of “SVC-1” and “SVC -2” will immediately make a bid with its costs because they are able to provide ΔQ. These costs are calculated based on how the change in the load of the SVC would affect its power losses and a total harmonic input from this SVC.

As for the SR Agents, taking a SR Agent “R-1” as an example, since it is most likely that ΔQ would not be equal to the nominal reactive power compensation value for any of the shunt reactors, and provided that ΔQ<0, SR Agent will calculate how much would it need to be compensated if it is to be operated, the value of which of course would be now $\Delta Q_1>0$. Then it would solicit proposals from the other RPC Agents for compensating ΔQ_1. Again, SVC Agents would respond immediately with proposals if they can provide ΔQ_1 with their bids. However, every time an RPC Agent solicits proposals for ΔQ it removes itself from the list of available compensators so when an RPC Agent receives a request and decides to solicit its own proposals to be able to provide needed ΔQ, it does not send its own requests to the agent he received request from in the first place. A SR Agent “R-2” would reply with a proposal to a “R-1” after having solicited its own proposals for ΔQ_2 to be able to provide ΔQ_1.

Then SR Agent “R-1”, having calculated its own cost for operating its shunt reactor which will take into account power losses and a state of the CB, will choose the cheapest bid that it has received for ΔQ_1 and adding it to its own bid will send the overall bid to the Bus Agent.

After Bus Agent gets all of the proposals from the RPC Agents, it chooses the best in terms of the cost overall proposal that may include several RPC Agents.

In this example, the winning bid turned out to be from the SVC Agent “SVC-1”.

2.2.7 Cost functions

When a RPC Agent estimates a cost for providing reactive power that was requested in the call for proposal message an agent uses cost functions to determine what price it should put in the bid.

Using these cost functions should be aimed at:

- Minimization of harmonic input into the system from VSRs like TCRs.
- Minimization of active power losses.
- Increase in equipment’s lifespan with regards to a CB.

The idea is that, when calculating the cost based on the cost functions, a cost increases with the increase in active power losses, increase in total harmonic input, or opening/closing a CB. And a total cost (TC) would be an aggregate of three components whereby each of these components is related to the aims specified above.

$$TC = HC + PLC + EOC, \tag{5}$$

where HC - harmonics related cost, PLC - power losses related cost, EOC - equipment operation related cost with regards to operating (opening/closing) a CB.

Since all components should have the same unified units, HC, PLC, and EOC are to be evaluated in euros.

Therefore, the best bid or the lowest one for providing requested reactive power is the one that leads to the least power losses, harmonic input and does not involve opening or closing CBs.

Harmonics related cost.

Harmonic input into the system from TCRs depends on the control angle α which is typically between 90° and 180°. At the firing angle of 90°, thyristors leak the full sinusoidal current to the TCR, while at 180° the current is reduced to zero. Between these limits of the control angle, there are current waveform distortions with only odd-order harmonics. Moreover, multiple different harmonics are generated at same firing angle, and the typical amplitude values of a given harmonic in relation to the rated current depending on firing angle are shown in Fig. 14.

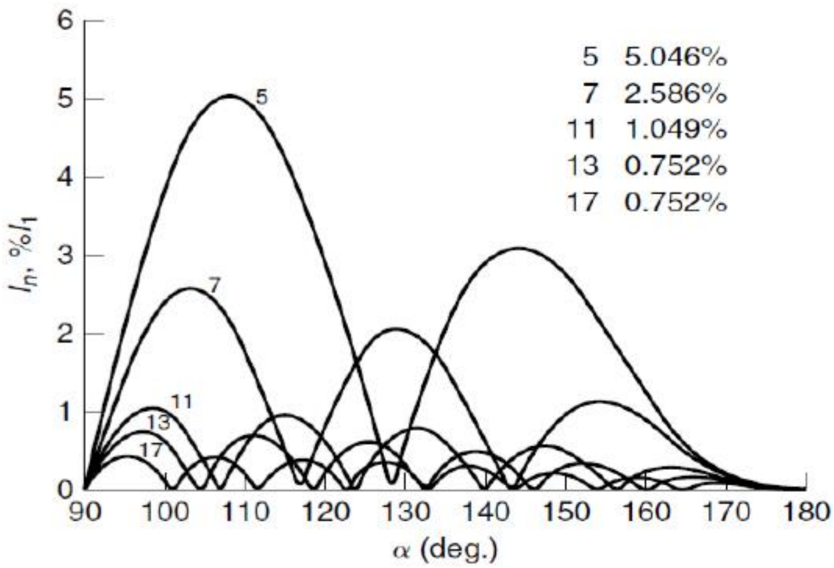


Fig. 14. Current harmonics of TCR depending on thyristor firing angle [8].

As for the cost related to harmonics HC, it would be proportional to the THD at a certain firing angle. That is why, HC would be higher in the areas where a total harmonic distortion is higher that is to say around 105 degrees and 145 degrees.

In order for HC to be in euros, there should be incorporated some base fines BF that are put in place for emanating harmonics in relation to THD. Thus cost related to harmonics can be calculated using the following formula:

$$HC = THD \cdot BF \,, \tag{6}$$

Power losses related cost.

Reactor total power losses (TPL) includes no-load losses (NLL) and load losses (LL).

NLL include dielectric loss, conductor loss in the winding due to exciting current, conductor loss due to circulating current in parallel windings, and core loss while LL are incident to the carrying of a specified load which include I^2R loss in the winding due to load and eddy currents, stray loss due to leakage fluxes in the windings, core clamps, and other parts, and the loss due to circulating currents in parallel windings or in parallel winding strands [9]. LL vary with the square of the load.

Thus, cost related to power losses is calculated using the following formula:

$$PLC = (NLL + LL) \cdot ET \cdot EP \,, \tag{7}$$

where ET - estimated time of operation with a certain load, EP - electricity price.

ET can be determined based on the statistical data.

Equipment operation related cost.

This cost is only applied to those reactive power compensators that require opening or closing of a CB like SR or MSC. Hence, for other compensators it would be 0.

Every CB has its own lifespan which is usually determined by the amount of on/off cycles a CB can withstand. And every certain number of on/off cycles a CB is put under maintenance which require a certain amount of money to perform.

Thus, cost related to power losses is calculated using the following formula:

$$EOC = MC \cdot \frac{(CNC - NNCBM)}{NNCBM} \,, \tag{8}$$

where MC - maintenance cost, $NNCBM$ - nominal number of cycles before maintenance, CNC - current number of carried out on/off cycles since the last maintenance.

As it can be seen from the formula EOC increases overtime with each operation of a CB. Thus if there is a choice whether to operate a SR whose CB is about to be put under maintenance or the one that is relatively further away from it, the latter would be operated since its cost is lower. That is why in the long run, a lifespan of a CB expressed in time would be increased.

Formula (8) might not be the most accurate interpretation in terms of real money cost but for the sake of prioritizing a certain reactive power compensator to another, it is pretty much sufficient.

2.2.8 Readiness confirmation

After the bidding process has been completed, Bus Agent knows which RPC Agents are going to provide ΔQ. A Bus Agent then requests their confirmation of readiness to provide ΔQ.

Upon receiving readiness confirmation request, RPC Agent verifies it can provide the value of reactive power put in the bid and if so, it then queries all agents whose equipment form a path from a busbar to its compensator about the state of their equipment. That is to say, whether any of the CB has been opened or not. This agents path is included in the initial request made by a Bus Agent so a RPC Agent knows whom to query.

The requests flow is shown in the Fig. 15.

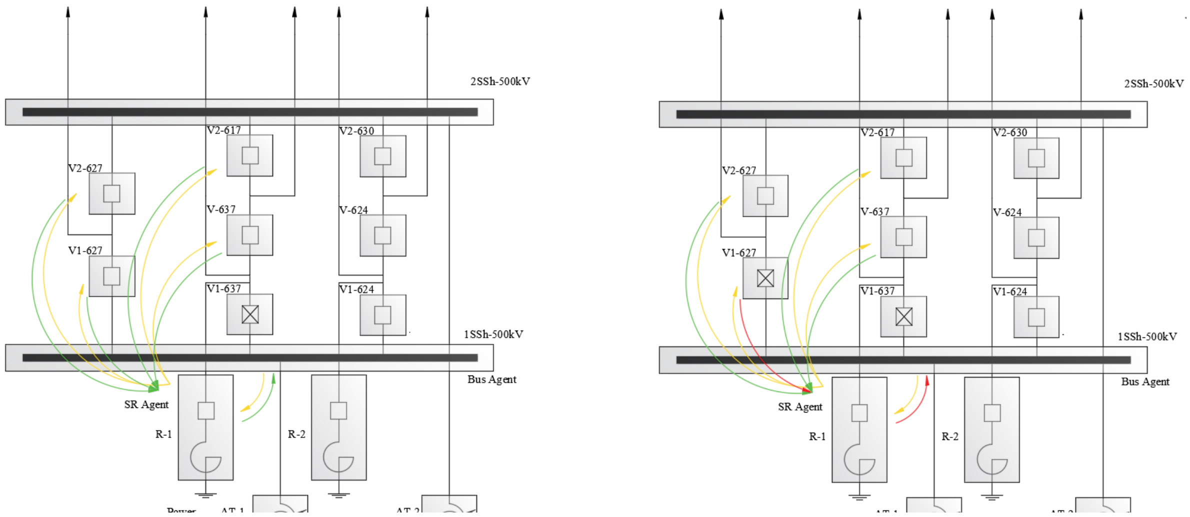


Fig. 15. Readiness confirmation.

Referring to a Fig. 14 as an example, where a SR Agent “R-1” has received a readiness confirmation request from a Bus Agent “1SSh-500kV”, after which SR Agent queries the state of the number of CB that form a path to the busbar. On the left side of a figure, there’s a case for when no changes were made to the state of a CBs during the bidding process. Thus, a SR Agent responds to a Bus Agent with a confirmation. While on the right side, there’s a case for when a CB “V1-637” gets opened and since the path to the busbar is now broken a SR Agent “R-1” would not confirm readiness to operate.

2.3 Implementation of MAS of reactive power compensators control

2.3.1 Configuration files

A configuration file is a set of initialization data for an agent of a particular substation element and includes information such as the name of the equipment, type of equipment, a list of neighboring equipment according to the single line diagram, and a number of other data depending on the specific type of equipment. Each file is individual for each particular equipment and is formed using XML language (eXtensible Markup Language), intended for structured storage and transmission of information.

Next, there will be viewed configuration files for all types of substation equipment used for a MAS, and that are represented by the respective types of agents described earlier.

Configuration file for a CB

An example of a configuration file for a CB “V-637” is shown below:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<agentSettings>
<name>V_637</name>
<equipmentType>CB</equipmentType>
<dataRetrievalTag>V_637</dataRetrievalTag>
<adjacentEquipmentLeft>
   <adjacentEquipment>
       <adjacentEquipmentName>V2_617</adjacentEquipmentName>
       <adjacentEquipmentType>CB</adjacentEquipmentType>
   </adjacentEquipment>
</adjacentEquipmentLeft>
<adjacentEquipmentRight>
   <adjacentEquipment>
       <adjacentEquipmentName>V1_637</adjacentEquipmentName>
       <adjacentEquipmentType>CB</adjacentEquipmentType>
   </adjacentEquipment>
   <adjacentEquipment>
       <adjacentEquipmentName>R_1</adjacentEquipmentName>
      <adjacentEquipmentType>ShuntReactor</adjacentEquipmentType>
   </adjacentEquipment>
</adjacentEquipmentRight>
</agentSettings>
```

This configuration file has the following elements:

- An element “agentSettings” – is a root element of the whole data structure.
- An element “name” – stores the name of the equipment according to the single line diagram of the substation.
- An element “equipmentType” – stores the name of equipment type for this particular equipment. For instance, in the configuration file shown above it is “CB”.
- An element “dataRetrievalTag” – stores a tag for retrieving the data regarding the state of CB (whether it is open or closed) from the model.
- An element “adjacentEquipment” – an element that stores information about adjacent to this CB equipment according to a single line diagram. By the adjacent it means that equipment is connected to this CB. This element also contains two child elements “adjacentEquipmentName” and “adjacentEquipmentType” that hold adjacent equipment’s name and its equipment type respectively.
- An element “adjacentEquipmetLeft” – this element stores a number of child elements “adjacentEquipment” from one side of this CB. For example, from the configuration file above it could be determined that this CB on side is connected to a CB “V2-617”.
- An element “adjacentEquipmetRight” – is the same as “adjacentEquipmetLeft” but stores child elements “adjacentEquipment” from the other side of this CB. Referring

to the same example, a CB “V-637” is connected to the CB “V1-637” and the SR “R-1”.

Configuration file for a compensator

The layout of the configuration file is the same for any compensator regardless of whether it is for a SR or a TCR.

A configuration file for a SR “R-1” is shown below as an example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RPCAgentSettings>
<name>R_1</name>
<equipmentType>ShuntReactor</equipmentType>
<cost>140</cost>
<minQload>-180</minQload>
<maxQload>0</maxQload>
<commandTag>0003004004</commandTag>
<dataRetrievalTag>QR_1</dataRetrievalTag>
<adjacentEquipmentLeft>
      <adjacentEquipment>
      <adjacentEquipmentName>V1_637</adjacentEquipmentName>
            <adjacentEquipmentType>CB</adjacentEquipmentType>
      </adjacentEquipment>
            <adjacentEquipment>
            <adjacentEquipmentName>V_637</adjacentEquipmentName>
            <adjacentEquipmentType>CB</adjacentEquipmentType>
      </adjacentEquipment>
</adjacentEquipmentLeft>
<adjacentEquipmentRight>
</adjacentEquipmentRight>
</RPCUAgentSettings>
```

This file is very similar to the one of a CB apart from a few additional elements:

- An element “RPCAgentSettings” is a root element of a file.
- An element “dateRetrievalTag” stores a tag that a RPC Agent uses to extract data from the simulation model such as current load of a TCR or whether SR is on or not.
- An element “commandTag” stores a tag that a RPC Agent uses to send commands to alter the load of its compensator. With respect to this SR, this tag is used to open or close a SR CB.
- An element “minQload” stores a minimum nominal value of a reactive power that could be compensated. From this example it can be seen that a SR’s nominal reactive power that it absorbs is equal to 180 Mvar.
- An element “maxQload” is the same but for a maximum nominal value. In this case it would be 0, since SR does not produce reactive power.

Configuration file for a busbar

A configuration file for a busbar “1SSh-500kV” is shown below as an example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<agentSettings>
<name>1SSh_500kV</name>
<equipmentType>Bus</equipmentType>
<cost>0</cost>
<dataRetrievalTag>1SShVrms</dataRetrievalTag>
<adjacentEquipmentLeft>
      <adjacentEquipment>
            <adjacentEquipmentName>V1_627</adjacentEquipmentName>
            <adjacentEquipmentType>CB</adjacentEquipmentType>
      </adjacentEquipment>
      <adjacentEquipment>
            <adjacentEquipmentName>V1_637</adjacentEquipmentName>
            <adjacentEquipmentType>CB</adjacentEquipmentType>
      </adjacentEquipment>
      <adjacentEquipment>
            <adjacentEquipmentName>AT_1</adjacentEquipmentName>
            <adjacentEquipmentType>AT</adjacentEquipmentType>
      </adjacentEquipment>
      <adjacentEquipment>
            <adjacentEquipmentName>V1_624</adjacentEquipmentName>
            <adjacentEquipmentType>CB</adjacentEquipmentType>
      </adjacentEquipment>
</adjacentEquipmentLeft>
<adjacentEquipmentRight>
</adjacentEquipmentRight>
</agentSettings>
```

This file is nearly identical to the configuration file of a CB but all adjacent equipment data is stored in only one element “adjacentEquipmentLeft” because there’s no such things as sides for a busbar as it is basically one node. And also an element “dataRetrievalTag” this time is used for monitoring voltage on the busbar.

Configuration file for a power transformer

A configuration file for a power transformer is identical to a that of a busbar and is shown below for a “AT-1”:

```
<agentSettings>
<name>AT_1</name>
<equipmentType>AT</equipmentType>
<cost>0</cost>
<adjacentEquipmentLeft>
    <adjacentEquipment>
      <adjacentEquipmentName>1SSh_500kV</adjacentEquipmentName>
            <adjacentEquipmentType>Bus</adjacentEquipmentType>
    </adjacentEquipment>
    <adjacentEquipment>
            <adjacentEquipmentName>VG_1</adjacentEquipmentName>
            <adjacentEquipmentType>CB</adjacentEquipmentType>
    </adjacentEquipment>
</adjacentEquipmentLeft>
```

```
<adjacentEquipmentRight>
</adjacentEquipmentRight>
</agentSettings>
```

To process the data from configuration files and initialize the initial data for each agent equipped with such a file, the built-in JAXB library (Java Architecture for XML Binding) is used.

JAXB is a fast and convenient way to convert Java objects to XML vice versa. JAXB maps XML elements and attributes to Java fields and properties using Java annotations. Reading data from an XML and mapping it into Java object is called un-marshalling and from Java object into a XML is called marshalling.

For retrieving data from XML there have been created two classes:

- AgentSettings is for a substation equipment such as busbar, power transformer, CB.
- RPCAgentSettings is for reactive power compensators.

2.3.2 Java classes and their relationship between each other

A set of classes used for this project and their inheritance relationships between each other is shown in the Fig. 16.

A main abstract class is a “PowerEquipmentAgent”, that has the most generic fields common to all equipment at the substation.

This class is extended by the following classes:

- “BusAgent” – is designed for a Bus Agent.
- “CBAgent” – is designed for a CB Agent.
- “PowerTransformerAgent” – is designed for a power transformer agent.
- “PRCUAgent” is an abstract class that has common for all RPC Agents fields and methods.

The following classes extend “PRCUAgent” class:

- “ShuntReactorAgent” – is designed for a SR Agent.
- “TCRAgent” – is designed for a TCR Agent.
- “MSCAgent” – is designed for a MSC Agent.

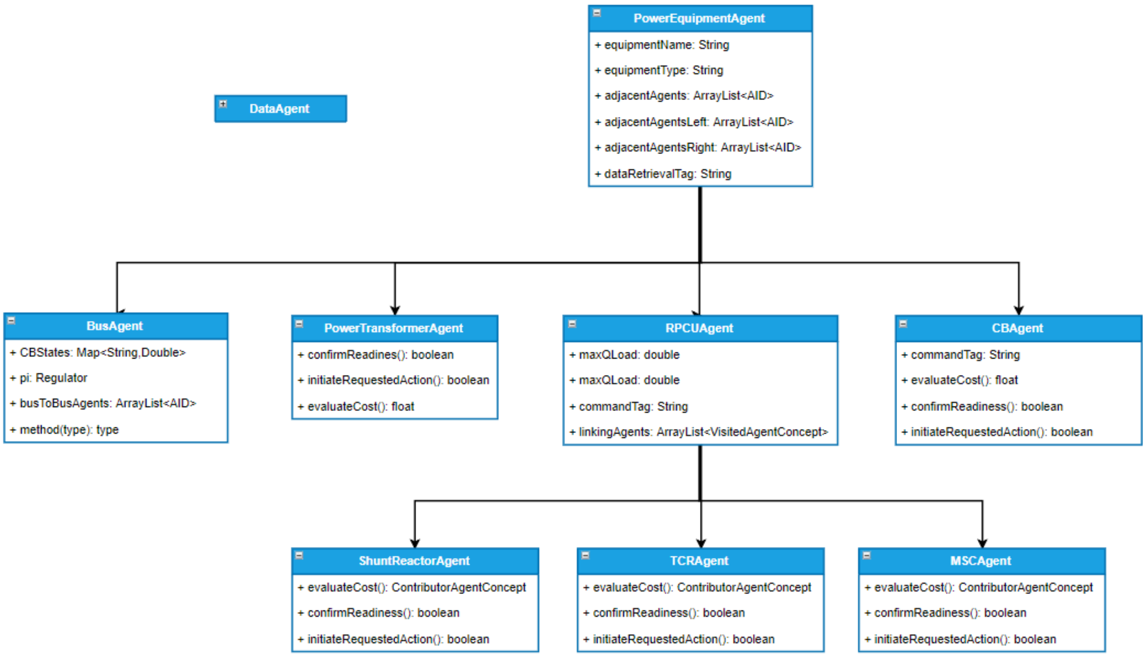


Fig. 16. Classes and their relationships.

2.3.3 Ontology and syntax of the messages.

When transferring messages between agents containing complex data, it is necessary to have a clearly defined syntax for all agents so that message recipients can parse and extract the information they need from them.

When implementing the agents' messaging process, the SL (Semantic Language) content lanuage is used which is recommended by the international organization that regulates the standards for MAS developers FIPA (Foundation for Intelligent Physical Agents).

An example of a message that was encoded with a SL syntax is given below:

```
(PowerEquipment :name "CB_500_1" :equipmentType "CircuitBreaker" :adjacentEquipment (sequence "Bus_500" "CB_500_2"))
```

In this message there was included an information about a CB with a name "CB_500_1" of a type "CurcuitBreaker" and its adjacent equipment in "Bus_500" and "CB_500_2".

Moreover, it is essential, that an agent that sends this kind of a message and an agent recipient have the same denotation or meanings about the terms or symbols used in the message. For instance, referring to the example in question, both agents should have the same understanding of the concepts such as “PowerEquipment”, “equipmentType”, “name”, and “adjacentEquipment”.

This set of concepts and symbols used to express them are known as an ontology [10]. Ontologies are usually very specific for every domain.

JADE offers ready-made libraries for using SL content language and creating any domain-specific ontology. Using these libraries will allow us have a content of a message be represented using Java objects which makes it relatively easy to handle. Apart from that these libraries do all of the message content conversion into an object and vice versa and also perform semantic checks of the ontology as shown in the Fig. 17.

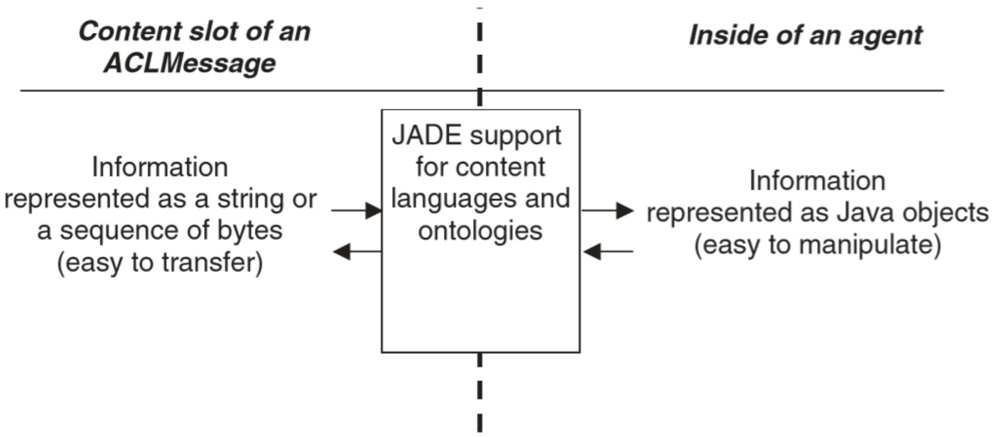


Fig. 17. The JADE support for content languages and ontologies [10].

To form a message by an agent and send to another agent, there can be used another approach which is Java serialization. Java serialization is a converting Java objects into sequences of bytes. However, for this work the use of content languages and ontology was justified for the following reasons:

- Java serialization is only applicable in a Java environment which means if at some point there will be a need to integrate some other agents into the MAS residing in a platform other than JADE, there is a chance that this agent would be able to understand the content of the message.

- Even though it is quite fast and simple approach, the messages if they were to be read by a human are non-readable whereas using ontologies and a content language would make a content of a message very easy to read and it is especially useful when debugging.

Before creating an ontology, there have to be created Java classes that specify the essence of the message and encapsulate some information within a message. These classes are also called schemas. In terms of semantics of the messages there are the following classes provided by the JADE library that need to be extended when making your own schemas:

- Predicate – class used when sending messages, the content of which say about a status of something which can be either true of false. The example of this message is shown below:

```
(Consumes (RPCU :name "TCR_1") (reactivePower: 35))
```

- AgentAction – class, needed when transmitting messages that contain a request to perform some action.
- Concept – class needed to group together some information. It is quite often used within a Predicate or AgentAction.
- And others.

2.3.4 Implementation of a Bus Agent's main cyclic behavior

As it has already been mentioned a behavior is something that an agent does. Any behavior represents a task that an agent can carry out and is implemented as an object of a class that extends a built-in class "Behavior".

To make an agent execute the task implemented by a behavior object, the behavior must be added to the agent's so-called pool of behaviors. When an agent completes execution of a certain behavior it removes this behavior from the pool.

Main cyclic behavior of a Bus Agent "mainCyclicBehaviour" is implemented as an object of a built-in class "TickerBehaviour".

This class has a method "onTick" that runs every specified amount of time. In case of this work this number will be 500 ms which is set during object initialization.

Method "onTick" on each tick adds another behavior of a class "PrimBus_HandleQAskedBehaviour" that extends built-in class "SequentialBehavior". "SequentialBehaviour" allows performing a sequence of other behaviors strictly in the order specified by the developer. Since normally if behaviors are added to pool of behaviors of the agent, there's no guarantee that they would be executed in the same order they were added. Therefore, there was used SequentialBehaviour to ensure that behaviors within a sequence are performed one by one in the right order.

Diagram of the "PrimBus_HandleQAskedBehaviour" behavior is presented in the Fig. 14. As it can be seen from the diagram it consists of seven subbehaviors that are executed one by one.

It should be mentioned that once all of the RPC Agents were identified via looking up in the register, it is no longer needs to be executed. Therefore, this subbehavior can be omitted after the very first cycle. And also, when there is no changes in the topology i.e. no CB changed its state, a repetitive search of the connected to a busbar RPC Agents would be redundant. That is why, the search is made only after a change in any of the CB's state and during the time when the topology remains the same this subbehavior also is omitted. This scenario is displayed in the bottom part of a diagram in the Fig. 18.

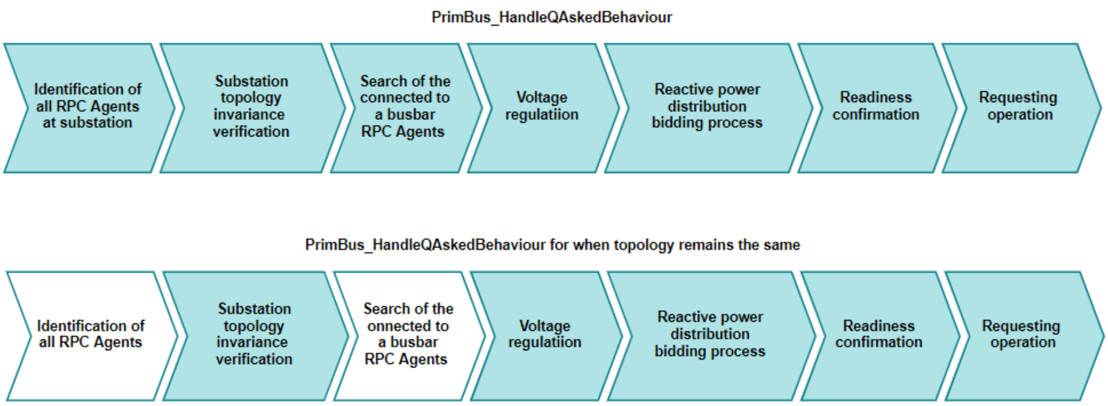


Fig. 18. "PrimBus_HandleQAskedBehaviour" behaviour.

The subbehavior are omitted by means of some boolean variables acting like triggers the value of which is checked just when a subbehavior begins execution. Should a subbehaviour needs to be omitted, its execution is finished right in the beginning when this variable is equal to false. Technically, they subbehaviors are not omitted completely, it is just their execution becomes very short.

Bus Agent's lifetime activity is shown in the diagram in the Fig. 19, where first is done an initialization and after that throughout its lifetime its executes every 500ms "mainCyclicBehaviour" which adds "PrimBus_HandleQAskedBehavior" to the pool of behaviors which in turns contain seven other subbehaviors.

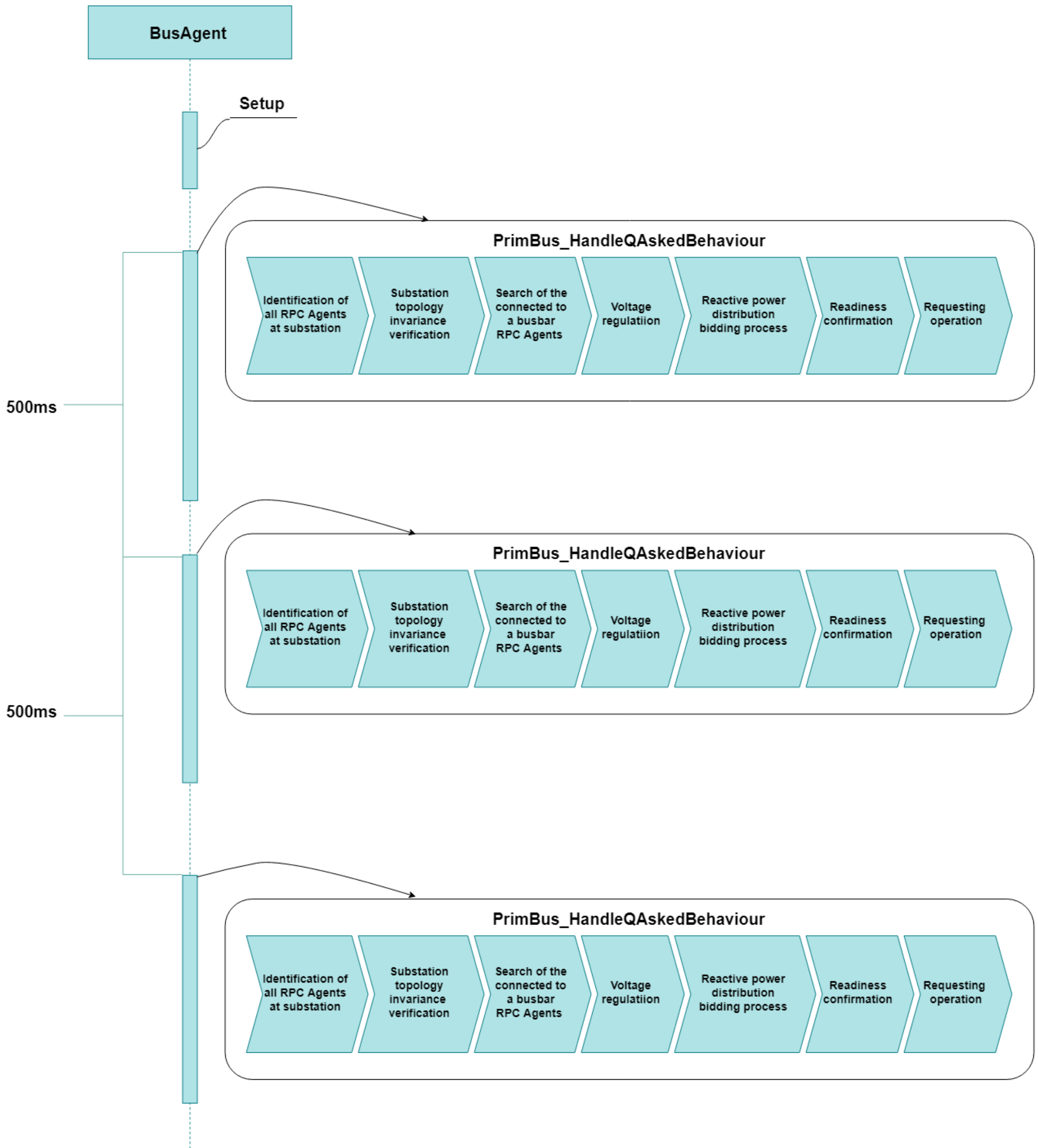


Fig. 19. Bus Agent's lifetime activity.

2.3.5 Search of the connected to a busbar RPC Agents

When working on this part of the project there have been used [11].

The subbehavior responsible for the search of all connected to a busbar RPC Agents is done only after a change in any of the CBs states takes place.

Implementation of this subbehavior the general idea of which was explained in the previous sections, is done in accordance with a FIPA-Request protocol. Diagram of the interaction process that takes place in a FIPA-Request protocol is shown in the Fig. 20.

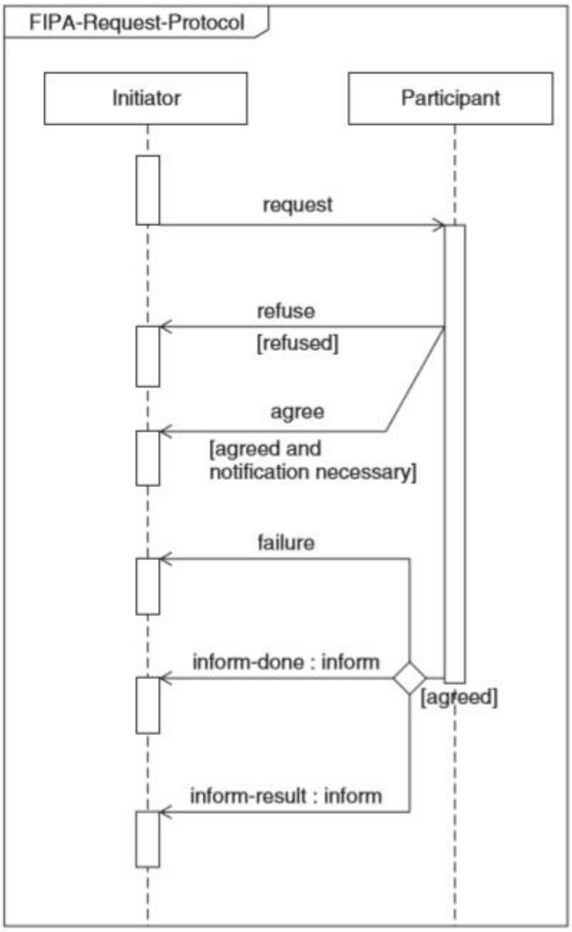


Fig. 20. FIPA-Request protocol [11].

According to the protocol in question there is an initiator that requests some one or several participants to perform some action. This is done by sending a request type of a message. A participant receives the requests and decides whether to agree to perform requested action or not. In case of a refusal, a participant respond with a message of a type refuse which can also include some information as to why the request was denied. In case if a participant agrees to perform an action it can respond with a message of a type agree and immediately proceeds with performing an action. Although sending an agree message is optional. After having performed the requested action a participant send either of the following types of messages:

- Failure – means that something went wrong and an action for some reason could not be done.
- Inform-done – means that the action was fully completed.
- Inform-result – means that the action was fully completed and includes a report on the result of that action.

To implement this protocol, two pre-written and ready-to-use classes are provided in the JADE: “AchieveREResponder” for a participant agent and “AchieveREInitiator” for an agent initiator. These classes make up finite state machines (FSM) that have a number of states where in each state an agent performs some action or behavior and when some conditions are met it moves from one state to another. The actions performed while in a state are done by means of executing a corresponding method. For instance, for the state PREPARE_INITIATIONS, the method with a similar name “prepareInitiations” is executed. Diagrams of these FSMs for these classes are presented in the Fig. 21 and Fig. 22.

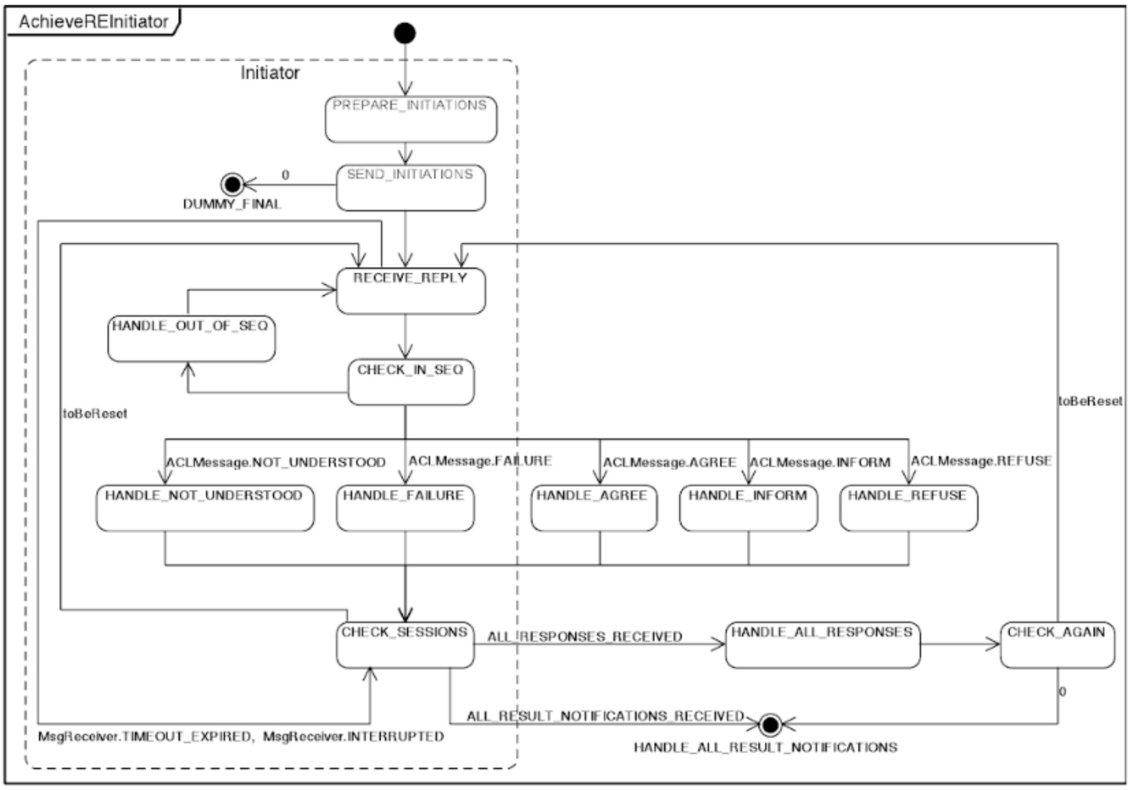


Fig. 21. FSM of a “AchieveREInitiator” class [12].

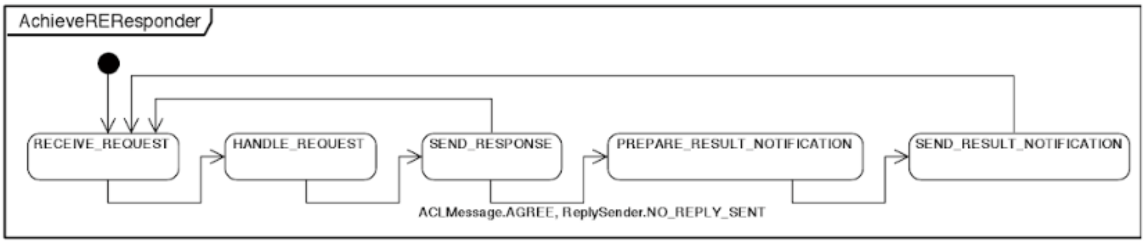


Fig. 22. FSM of a “AchieveREResponder” class [12].

Therefore, for this work when realizing the search of the RPC Agents connected to the busbar in accordance with a FIPA-Request protocol these two classes were used.

In this project, a Bus Agent acts as an initiator requesting its adjacent agents to find all RPC Agents that has been looked up in register earlier on while these adjacent agents or any other agents who received these kind of requests act as participants in the FIPA-Request terms. That is why, for the subbehaviour that does the search of a RPC Agents connected to the busbar within a “PrimBus_HandleQAskedBehaviour” discussed earlier there has been created a behavior class “AchieveRE_Initiator” which extends “AchieveREInitiator”. A FSM of a “AchieveRE_Initiator is the same as it is for a “AchieveREInitiator”.

Every single agent (CB Agent, or RPC Agent etc.) has a behavior “AchieveRE_Responder” which extends a class “AchieveREResponder” discussed earlier. However, an extended version of this behavior is implemented in way that within of one of its states of a FSM resides another FSM-like behavior in “AchieveRE_Initiator”. This state which encapsulates another FSM is called PREPARE_RESULT_NOTIFICATION. This means that when this state is entered an agent being externally in a “AchieveRE_Responder” FSM becomes an agent initiator with its own FSM within the PREPARE_RESULT_NOTIFICATION state. FSM of a created behavior class “AchieveRE_Responder” is shown in the Fig. 23.

Next, there will be considered the order of actions and operations of agents in the implementation of the search of RPC Agents connected to the busbar.

A Bus Agent while being in the state SEND_INITIATIONS, sends requests to its adjacent agents or agents whose equipment is connected to the busbar to find RPC Agents.

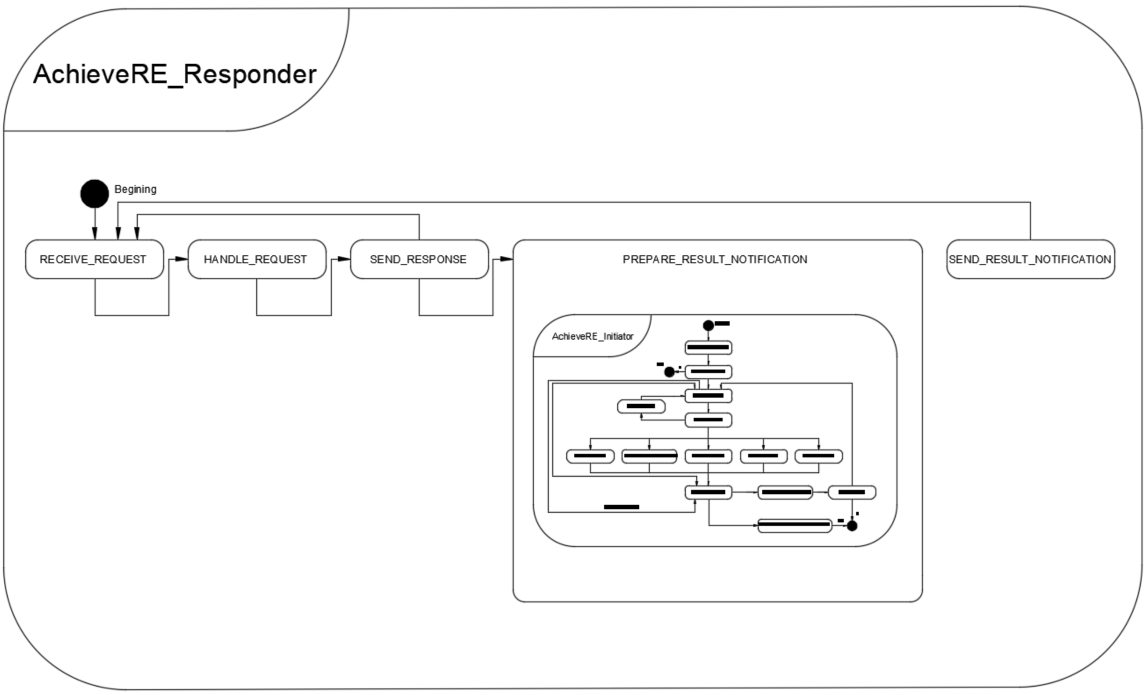


Fig. 23. FSM of a “AchieveRE_Responder” class.

When an agent participant receives this request, it moves to the state HANDLE_REQUESTS and processes the request. Processing is done by the overridden method "handleRequests". The following actions are taken while executing this method. First being in the state in question, an agent checks if its equipment is in operation. For instance, should a CB Agent receive this request, it checks whether its CB is in operation. If it is not, an agent immediately responds with a refuse message, otherwise an agent checks the next condition – should this agent forward the same request to its own adjacent agents would this request be sent to an agent that has already sent this very request before. In other words, a request is going to enter a loop. A simple block diagram of an agent decision making process within a HANDLE_REQUESTS state by means of executing "handleRequests" method is shown in the Fig. 24.

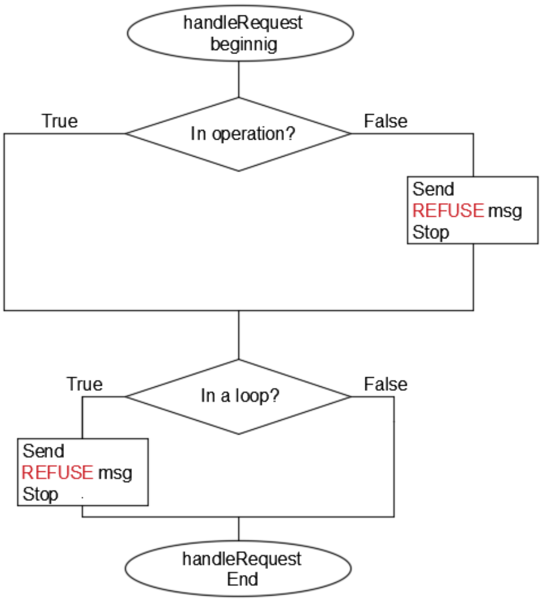


Fig. 24. Block diagram of a method "handleRequest".

Once an agent has sent a refuse message it moves back to the state RECEIVE_REQUEST waiting for a new request to arrive.

If an agent-participant is not to send a refuse message it will move to the PREPARE_RESULT_NOTIFICATION state going past the SEND_RESPONSE state in which it will not send anything.

Being in the state PREPARE_RESULT_NOTIFICATION an agent participant within that state becomes an agent-initiator and enters a PREPARE_INITIATIONS state of a FSM-like behavior “AchieveRE_Initiator”. A method “prepareInitiattions” shown in the Fig. 25. runs when an agent is in that state. This method first checks, whether this agent is the agent being searched. If an agent-participant who has received this request is among those who are being searched, an agent prepares an inform message to send it to an agent-initiator and sends no requests while being internally an initiator itself which results in terminating AchieveRE_Initiator behavior. In case of a not being an agent being searched, an agent participant forwards received request to its adjacent agents and waits for their responses which it turn do the very same thing upon receiving this request.

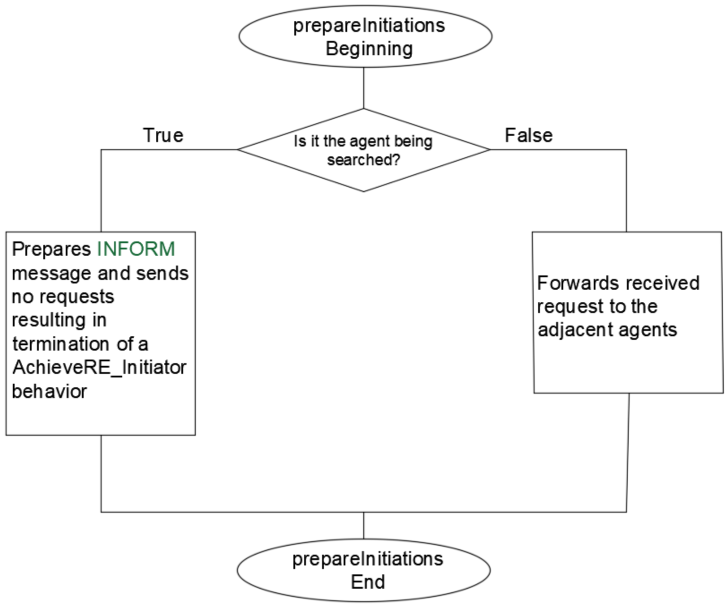


Fig. 25. Block diagram of a method “prepareInitiatons”.

After an agent that forwarded received request to its adjacent agents, this agent periodically checks its mailbox for the arrival of inform messages from its adjacent agents or agents-participants with respect to the agent in question being an initiator who did not send a refuse message. As soon as an agent gets all of the expected inform messages, it switches states to HANDLE_ALL_RESULT_NOTIFICATIONS and a method “handleAllResultNotifications” shown in the Fig. 26 is executed.

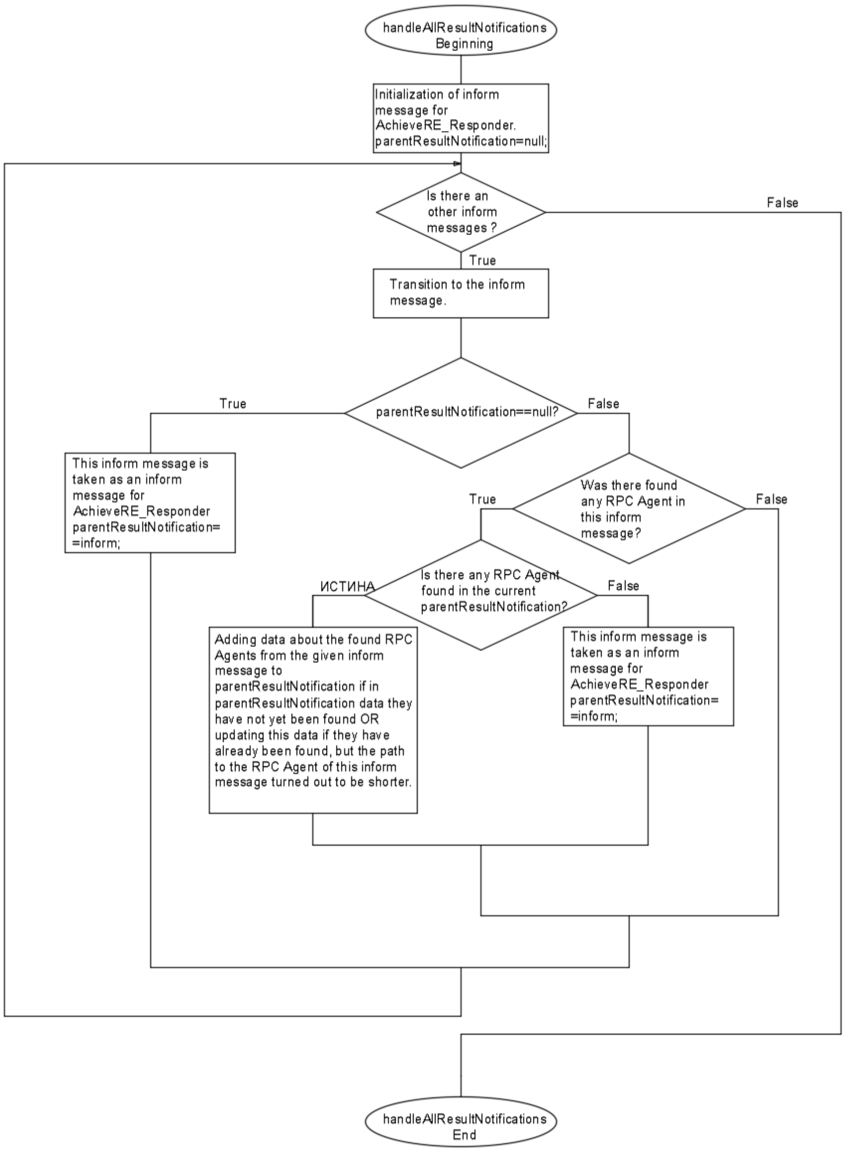


Fig. 26. Block diagram of a method "handleAllResultNotifications".

In the beginning of this method an inform message "parentResultNotification" is initialized, which will eventually be sent as an inform message to this agent's initiator. Then, in the loop every received inform message that was prepared within a PREPARE_RESULT_NOTIFICATION state is looked into. The very first inform message is assigned to a "parentResultNotification" by the default. For the next messages it is checked whether there was found at least one RPC Agent. If no RPC Agents were found, the next inform message is looked into. Otherwise, it is checked whether there have been found at least one RPC Agent in the earlier assigned "parentResultNotification". If not, this new inform message is assigned to "parentResultNotification". Otherwise, the data of a new message is merged with the data of “parentResultNotification” so that if an RPC Agent’s path turned out to be shorter in terms of the number of equipment in between a busbar and a RPC Agent then its data is substituted with the one from currently considered inform message or if any of the RPC Agents that had not been found in the “parentResultNotification” was found in the considered inform message then the data about the path to this RPC Agent is added to “parentResultNotification”.

After the end of the “handleAllResultNotifications” method, the “AchieveRE_Initiator” behavior is terminated and an agent moves to a SEND_RESULT_NOTIFICATION state from which is sends “parentResultNotification” message of a type inform.

Eventually Bus Agent will receive from its adjacent agents inform messages with the data on what RPC Agents are connected to the busbar with the shortest path to them in terms of a number of equipment in between.

2.3.6 Reactive power distribution bidding process

Bidding for the distribution of reactive power between agents is initiated by a Bus Agent.

The implementation of the reactive power distribution process, the general idea of which was described in the previous section, was carried out in accordance with the FIPA-ContractNet protocol. The diagram of this protocol is shown in Fig. 27.

According to this protocol, there is an initiator agent, which, solicits proposals on the execution of a certain action from one or more other participant agents. Soliciting is done by sending messages of a type call for proposal (cfp) in the content of which there is a specification on requested action. Then, participant agents respond either with a refusal, sending messages of the type refuse, or send their proposals or a bids to perform the requested action with the cost of that action. After an initiator agent has received all of the proposals or bids it selects the cheapest bid and sends a message of a type accept-proposal to the agent that had made that bid. Everyone else receives a rejection message of a type reject-proposal. After the selected agent completes the attempt to perform the requested action, it sends the following possible message types to the agent initiator:

- Failure – means that something went wrong and an action for some reason could not be done.
- Inform-done – means that the action was fully completed.
- Inform-result – means that the action was fully completed and includes a report on the result of that action.

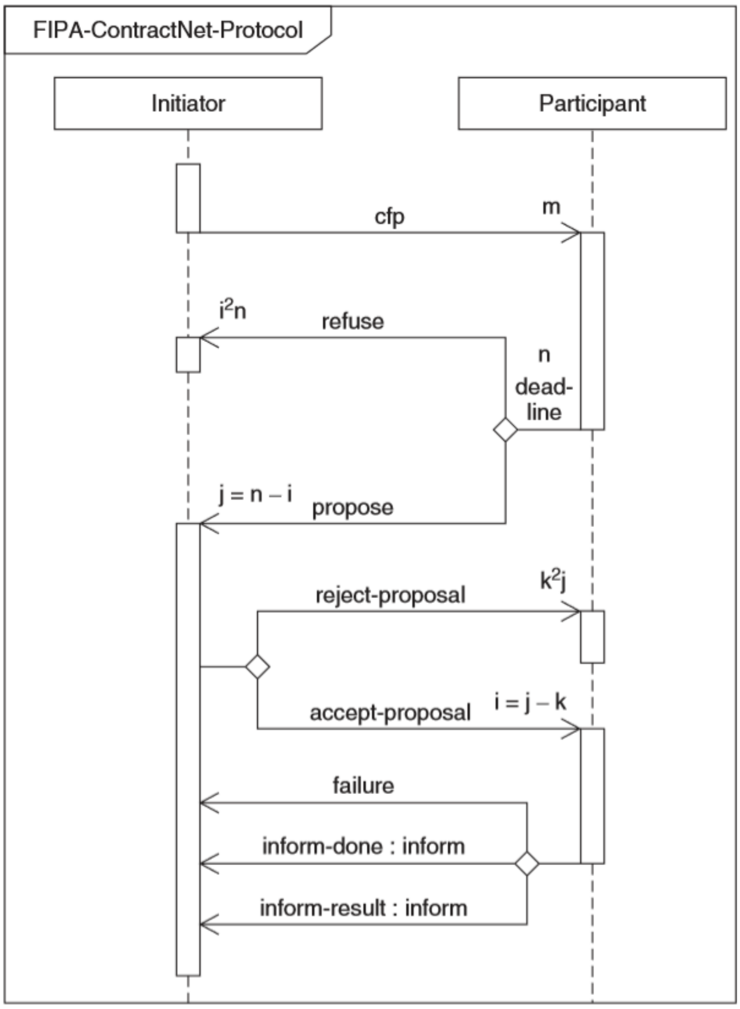


Fig. 27. FIPA-ContractNet protocol [13].

To implement this protocol, two pre-written and ready-to-use classes are provided in the JADE: “ContractNetResponder” for a participant agent and “ContractNetInitiator” for an agent initiator. Just like in FIPA-Request protocol, these classes make up finite state machines (FSM). Diagrams of these FSMs for these classes are presented in the Fig. 28 and Fig. 29.

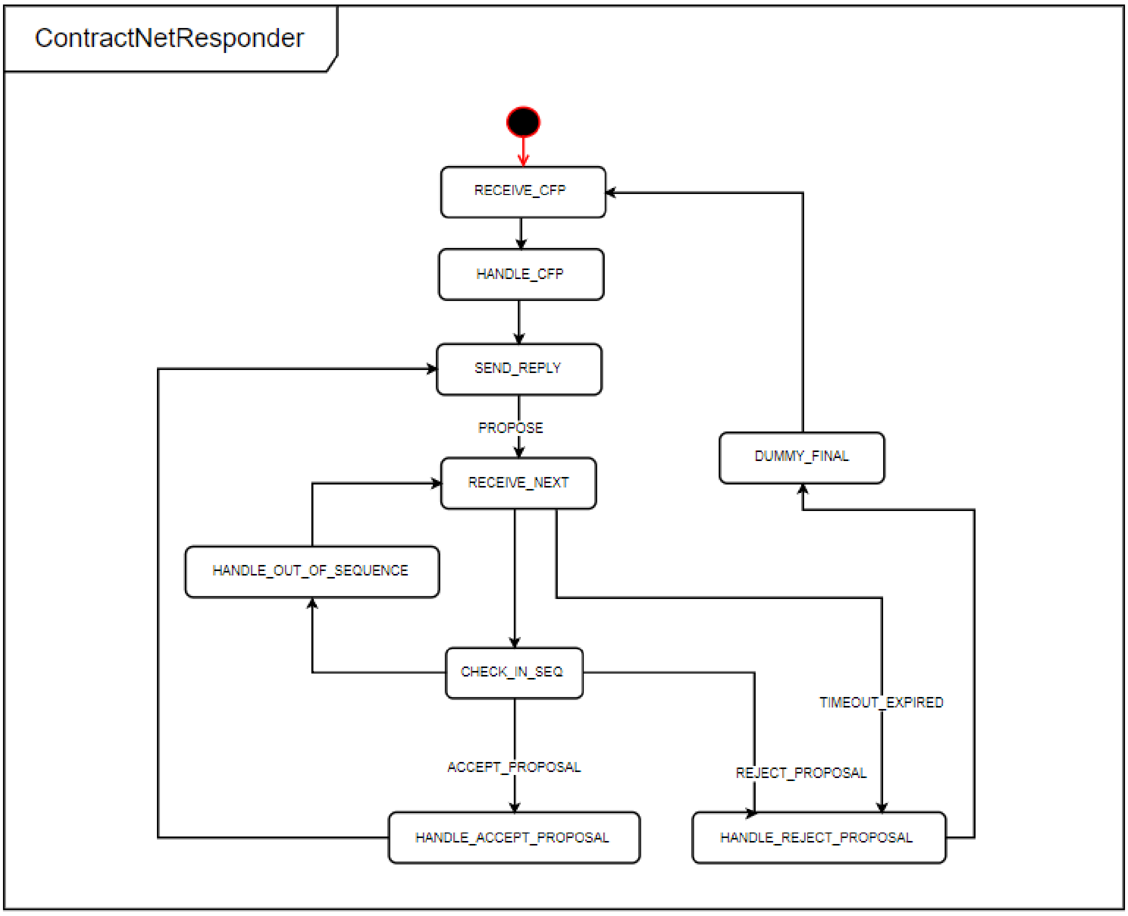


Fig. 28. FSM of a “ContractNetResponder” class.

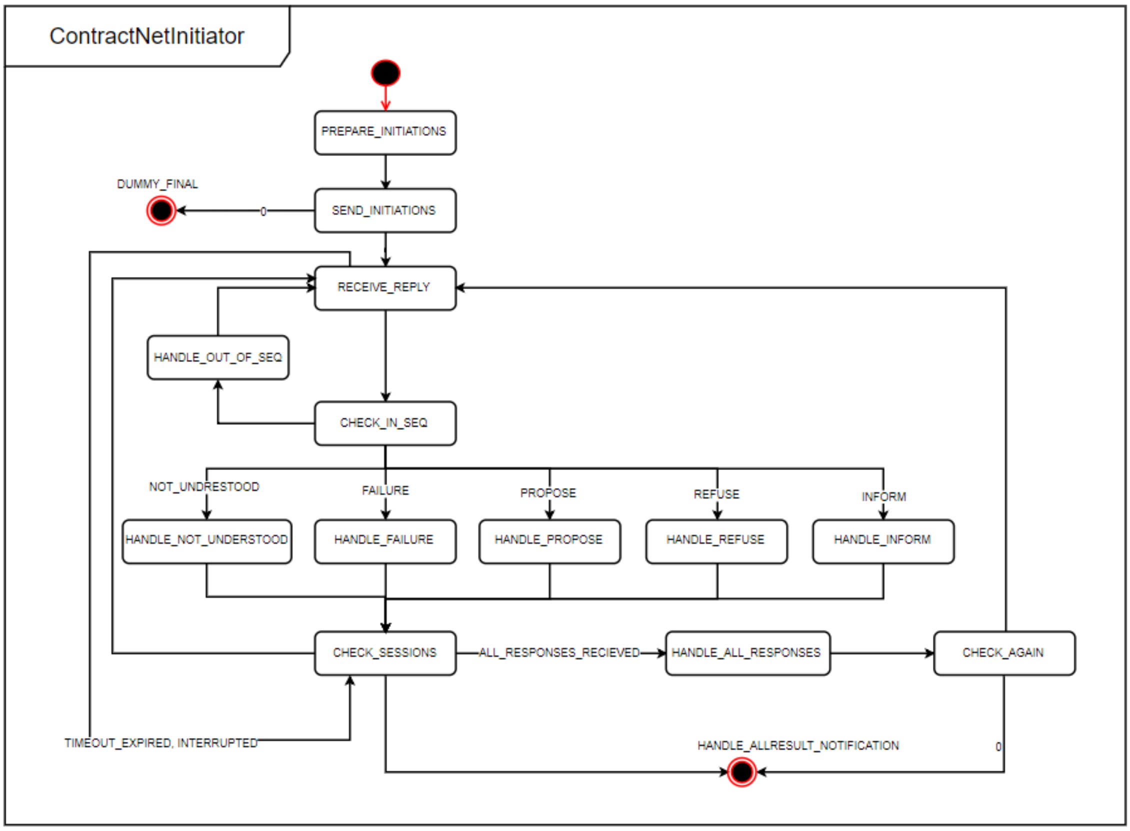


Fig. 29. FSM of a "ContractNetInitiator" class.

Therefore, for this work when implementing reactive power distribution bidding process in accordance with a FIPA-ContractNet protocol these two classes were used.

In this project, a Bus Agent acts as an initiator soliciting proposals for providing reactive power from all RPC Agents connected to its busbar that have been identified through the search described in the previous section. These RPC Agents act as participants in the FIPA-ContractNet terms. Therefore, for the subbehaviour that is responsible for the reactive power distribution bidding process within a "PrimBus_HandleQAskedBehaviour" discussed earlier there have been created a behavior class "ContractNet_Initiator" which extends "ContractNetInitiator" and a behavior class "ContractNet_Responder" which extends "ContractNetResponder". A FSM of a "ContractNet_Initiator" is the same as it is for a "ContractNetInitiator".

Every RPC Agent has a behavior "ContractNet_Responder". However, in this case, the behavior performed by the agent inside the HANDLE_CFP state is redefined to the behavior of “ContractNet_Initiator” like it was done for a “AchieveRE_Responder” but for a different state. This means that when this state is entered an agent being externally in a “ContractNet_Responder” FSM becomes an agent initiator with its own FSM within the HANDLE_CFP state. FSM of a created behavior class “ContractNet_Responder” is shown in the Fig. 30.

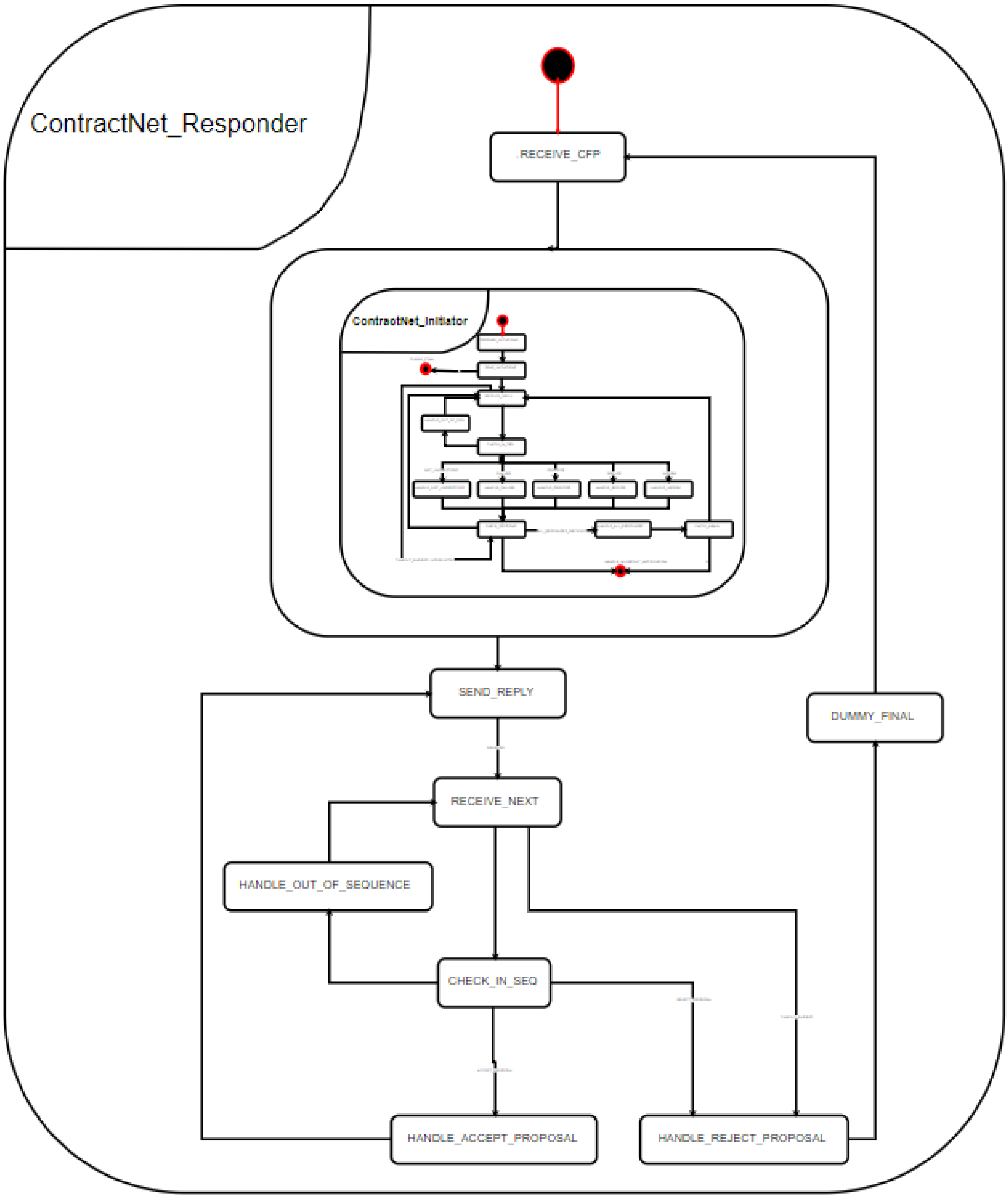


Fig. 30. FSM of a “ContractNet_Responder” class.

Bidding starts with the Bus Agent sending messages with the call for proposal to all connected to its busbar RPC Agents. These messages include information on what amount of reactive power is required for compensation along with a list of all RPC Agents connected to a busbar “availableRPCUAgents”.

When an agent participant receives a call for proposal message from either a Bus Agent or a RPC Agent while being in the state RECEIVE_CFP, it moves to the HANDLE_CFP state. A behavior an agent has within that state is redefined to “ContarctNet_Initiator”. That is why an agent will switch to the PREPARE_INITIATIONS state of a “ContarctNet_Initiator” behavior in which a method “prepareCfp” is executed. Block diagram of this method is shown in the Fig. 31.

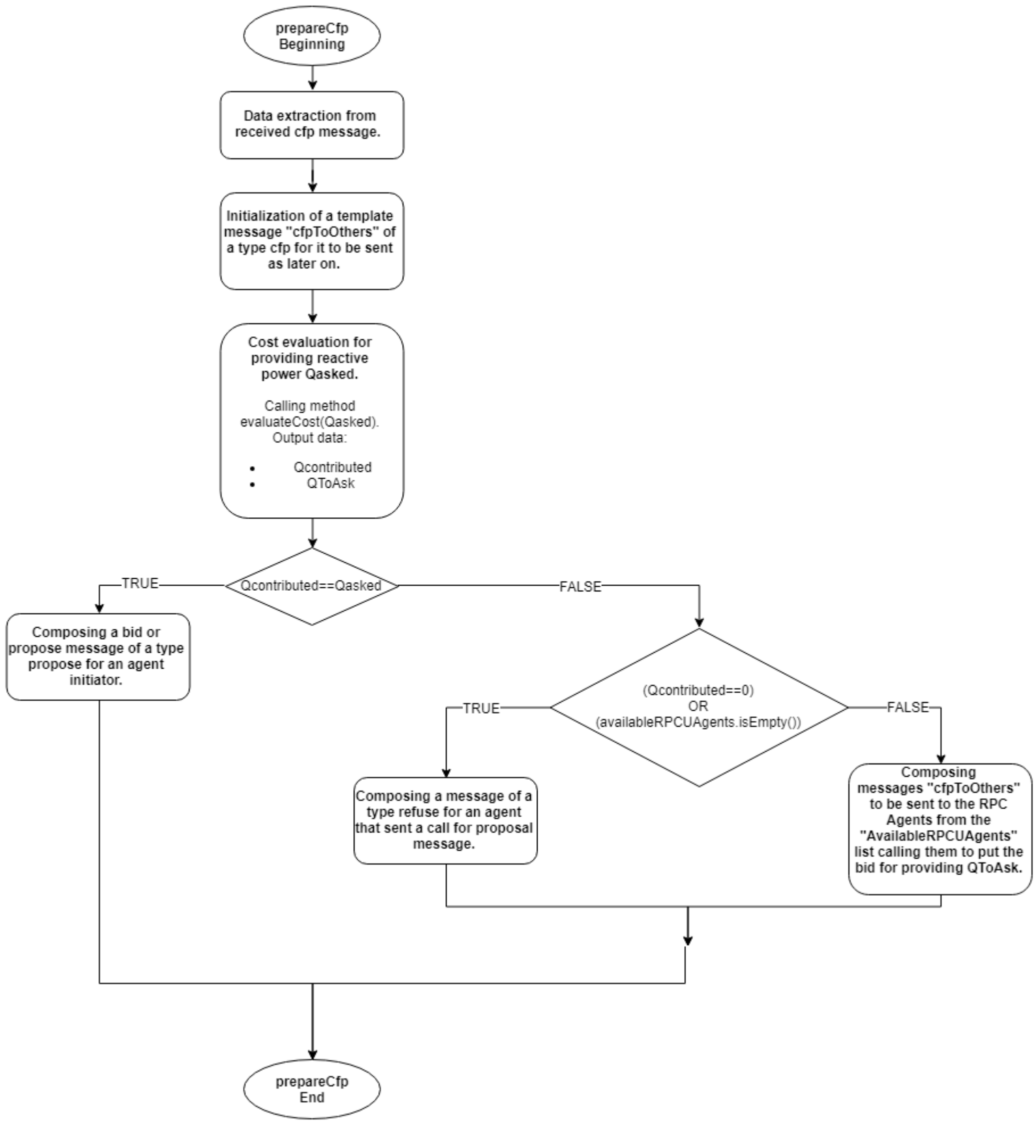


Fig. 31. Block diagram of the “prepareCfp” method.

At the beginning of the "prepareCfp" method, data is extracted from the received message of a type cfp. Next, a message "cfpToOthers" of a type cfp is initialized that at this point does not have any recipients. These messages assuming there are several recipients would be sent when an agent enters SEND_INITIATIONS state. After having initialized a cfp message, an agent calls its own method "evaluateCost" which estimates a cost for providing reactive power Qasked specified in the call for proposal message. The output data from this method are the following values:

- Qcontributed – reactive power that could be compensated by this RPC Agent.
- Cost – the abstract value which takes into account the current state of the equipment, harmonic output and power losses should this compensator be operated.
- QToAsk – reactive power that still should be compensated if this compensator is to be operated. For instance, in the case when the RPC Agent does not have enough regulation range to fully compensate for Qasked, or in the case when counterbalance compensation is necessary, for example, due to the operation of a shunt reactor agent, it is most likely necessary to compensate for the difference between the rated power of the shunt reactor and Qasked.

If the Qcontibuted value is equal to the Qasked value, the agent can compensate for the Qasked value and composes a propose message for the agent who sent the cfp message and then an earlier initialized message "cfpToOthers" request which does not have any recipients is composed to be sent later on in the state SEND_ INITIATIONS. Sending cfp message to no recipients terminates the "ContractNet_Initiator" behavior.

If the Qcontibuted value is not equal to the Qasked value, it is checked if at least one of the following conditions is true:

- Qcontributed is equal to zero, which means that the RPC Agent cannot provide Qasked.
- If a list of RPC Agents "availableRPCUAgents" is empty, which means that the agent cannot send anyone else calls for proposal for compensation of Qasked. A list of RPC Agents "availableRPCAgents" which is included in the cfp message when a Bus

Agent initiates a bidding process consists of all RPC Agents connected to its busbar.

If at least one of these conditions is true, a message of a type refuse is composed for the agent who sent the cfp message. Otherwise, an agent composes its own call for proposal messages from the template “cfpToOthers” that was initialized at the beginning of this method with a call to provide QToAsk. At the same time, in these calls for proposal, the list “availableRPCUAgents” is included but with the exclusion of this very agent, so that the agent that received this call for proposal does not send its own call for proposal back to this agent. These messages are formed to be sent to all RPC Agents left in the list “availableRPCAgents”.

Next, the agent goes into the SEND_INITIATIONS state and sends composed call for proposal messages. However, like was said earlier, if the “cfpToOthers” has no recipients the behavior of ContractNet_Initiator is finished and transition to the SEND_REPLY state of the behavior of ContractNet_Responder is performed. If there are some recipients, an agent after having sent the calls for proposal will wait for the responses to come with the bids to for providing QToAsk.

When an agent receives bids it selects the cheapest bid of all and puts together a final aggregate bid for providing Qasked, consisting of compensation of a Qcontributed by itself and QToAsk by the winning agent with a cheapest bid. Thus, the aggregate bid or proposal will include a list of RPC Agents with the values of the reactive power that is to be compensated by each of them and the costs of their compensation. After that, an agent will send to all participant agents or bidders, including the winning agent, reject messages, so that all agents that have sent their bids could go to the state RECEIVE_CFP having gone through the state HANDLE_REJECT_PROPOSAL and be available for receiving other calls for proposals. This completes the behavior “ContractNet_Initiator”, and the agent enters the SEND_REPLY state of the behavior “ContractNet_Responder”.

In the SEND_REPLY state the following scenarios can take place:

- An agent sends a message of a type refuse because it cannot fully compensate Qasked or it partially can but there is no one left in the list “availableRPCUAgents” to ask for a QToAsk to be compensated by.

- An agent sends a bid the message of which is of a type propose and in this bid Qasked is compensated solely by this agent.
- An agent sends an aggregate bid in which several RPC Agents take part in compensating Qasked and the combined cost or price is put in this aggregate bid. The message of this bid is of a type propose as well.

Next, this agent will receive a reject-proposal type message, regardless of whether its proposal is the best or not so that he can switch to the RECEIVE_CFP state and wait for new calls for proposals to come.

Eventually, a Bus Agent will receive bids from each of the RPC Agents where every bid is somewhat the best aggregate bid which involves this very RPC Agent. A Bus Agent then again selects the best bid of these and ultimately will have the best combination of RPC Agents to provide reactive power that is needed to be compensated.

2.3.7 Readiness confirmation

After a Bus Agent has determined what would be the most reasonable combination of RPC Agents to operate in terms of power losses, harmonic outputs, and increase in power equipment lifespan, it can then query these RPC Agents involved if they are ready to operate.

Confirmation, just like the bidding process and the search of connected to the busbar RPC Agents is initiated by the Bus Agent.

This behavior is done in accordance with a FIPA-Query protocol. Interaction diagram between an initiator and a participant is shown in Fig. 32.

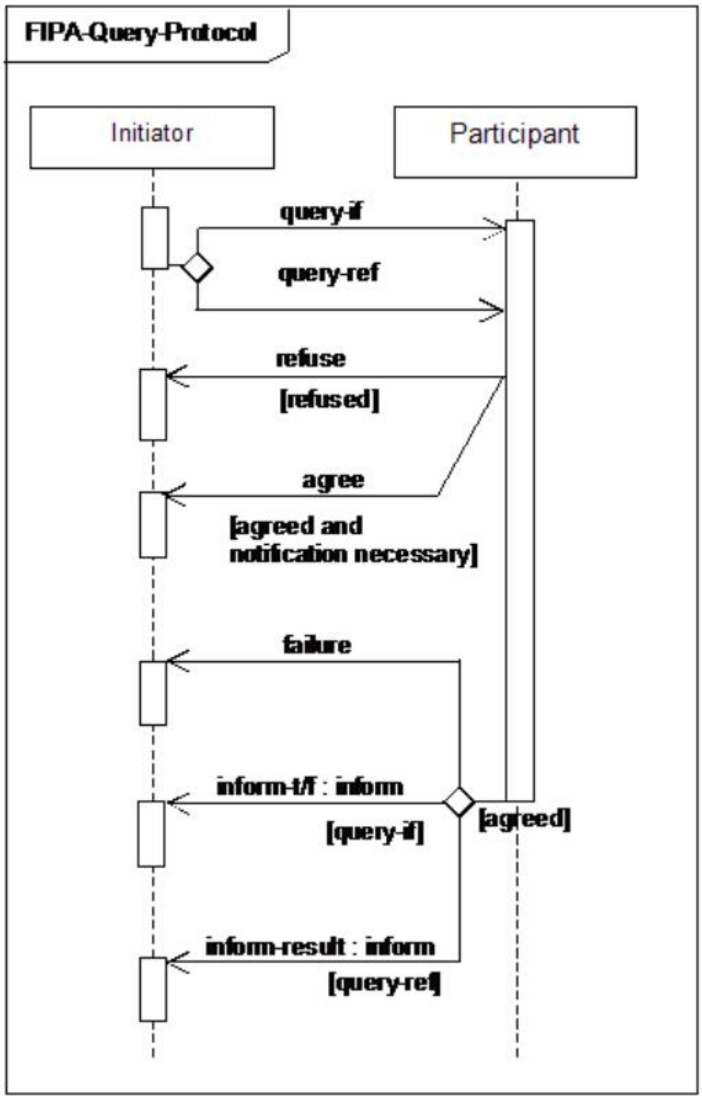


Fig. 32. FIPA-Query protocol [14].

This protocol is very similar to FIPA-Request where an initiator instead of performing an action requests clarification on whether something is true or not. In other words, it makes a query request. That is why, to implement this protocol in JADE the same classes as for FIPA-Request protocol are made use of which are "AchieveREInitiator" and "AchiveREResponder".

A Bus Agent acts as an initiator which queries winning RPC Agents in the bidding process earlier on about whether they are ready to provide previously requested reactive power. To do that it uses a behavior "ConfirmationQuery_Initiator" that extends "AchieveREInitiator".

As for RPC Agents, each one of them has a "ConfirmationQuery_Responder" that allows them to respond to the queries sent by a Bus Agent. This behavior extends "AchieveREResponder" class, but has a behavior within a PREPARE_RESULT_NOTIFICATION state redefined to the "ConfirmationQuery_Initiator" behavior.

Everything starts with a Bus Agent sending messages of a type query_if to the winning RPC Agents asking them to confirm that they are ready to change their reactive power output. These messages include by how much the output needs to be changed for this RPC Agent and also the path between the busbar and this RPC Agent's compensator. The path in question is a list of a power equipment that links the busbar and a compensator. For instance, if a SVC is connected via power transformer and two CBs then this list would include these three agents representing this equipment.

When a RPC Agent receives a query_if message, it immediately switches to the PREPARE_RESULT_NOTIFICATION state because according to this protocol it is not necessary an agree message. In this state, the "ConfirmationQuery_Initiator" behavior is started and the transition into the inner state PREPARE_REQUESTS of this behavior is made. In this state a method "prepareRequests" is executed. Block diagram of this method is shown in the Fig. 33.

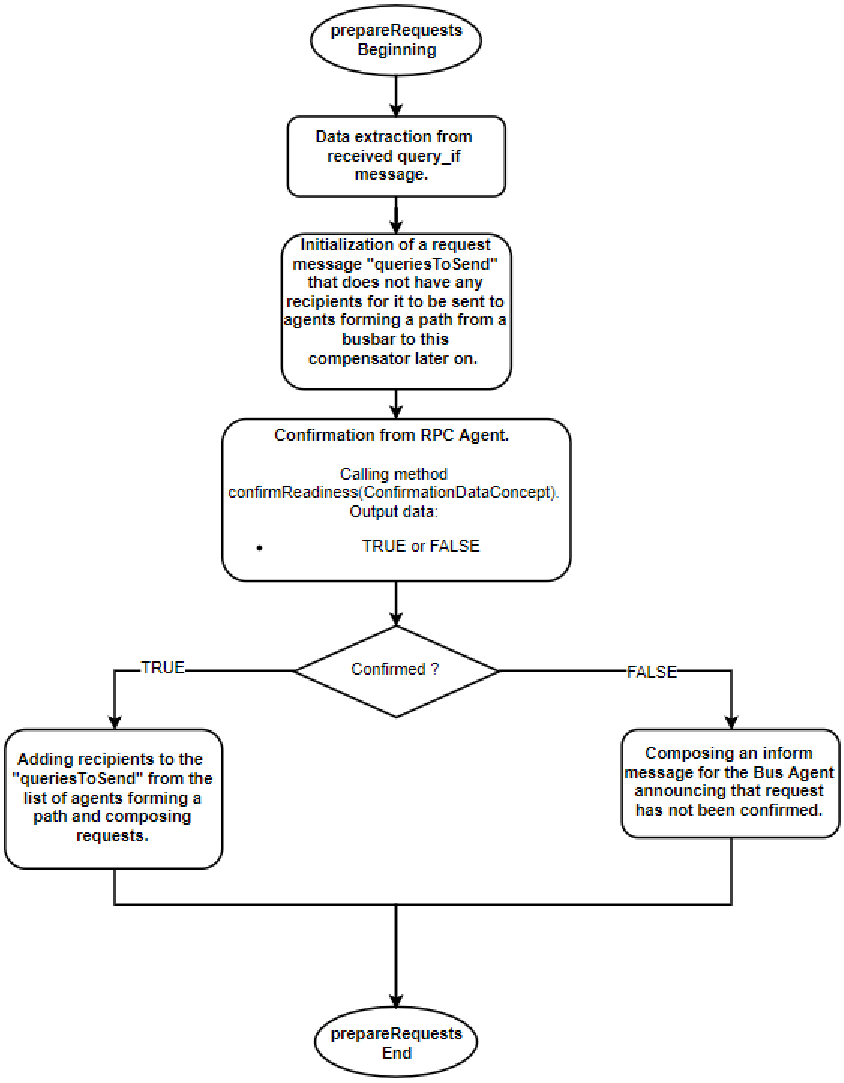


Fig. 33. "prepareRequests" method of the "ConfirmationQuery_Responder" behavior.

In the beginning of the "prepareRequests" method a content of a query_if message is extracted. Then the query request message "queriesToSend" is initialized of a type query_if that would be sent once the state is switched to SEND_INITIATIONS of the behavior "ConfirmationQuery_Initiator". It should be noted that at this point the message "queriesToSend" does not have any receivers nor it has any data included in it.

After that, a RPC Agent call its own method "confirmReadiness" that verifies the ability of a RPC Agent to provide asked reactive power. This method returns either true or false. If it is a false, a RPC Agent for some reason no longer can change its reactive power output and it composes an inform-type-of-message for Bus Agent in the content of which it is specified that readiness has not been confirmed. And then "queriesToSend" remains the same without any recipients which will prompt termination of the “ConfirmationQuery_Initiator” behavior after the state SEND_INITIATIONS. If it is true, the messages of a type query_if are composed for the agents whose equipment forms a path between a busbar and a compensator by specifying recipients in the “queriesToSend” message. The essence of this query is to ask mainly all CB Agents that form a path whether all of CBs retained their states while there was a bidding process.

Then, when an agent moves to a SEND_INITIATIONS state it sends earlier prepared “queriesToSend”. Like was already said, the “ConfirmationQuery_Initiator” is terminated if there are no recipients and the state is switched to SEND_RESULT_NOTIFICATION of a “ConfirmationQuery_Responder” behavior from which an earlier composed in the “prepareRequests” method inform message is sent.

If “queriesToSend” has recipients, after having sent the queries a RPC Agent will wait for inform messages from each one of the agents that form a path.

After having received all of the inform messages, being in the state HANDLE_ALL_RESULT_NOTIFICATIONS, a RPC Agent checks if all of the agents forming a path have retained their state. If at least one of them changed its state this RPC Agent will composes an inform message saying that it is not ready and sends it to a Bus Agent when in the state SEND_RESULT_NOTIFICATION of “ConfirmationQuery_Responder” behavior.

Then when a Bus Agent has received all of the inform messages from all winning RPC Agents about confirmation it moves to a HANDLE_ALL_RESULT_NOTIFICATION state of a behavior “ConfirmationQuery_Initiator” and checks if everyone is ready. And if it is the case, then the next step is taken which is operation request.

2.3.8 Operation request

After RPC Agents have confirmed their readiness to change their reactive power output Bus Agent requests these RPC Agents to perform the change in the reactive power output. For instance, a Bus Agent may request a SR to turn on and an TCR reduce its reactive power consumption by 35 %.

The operation request has been implemented in accordance with FIPA-Request protocol. And there have been created the following classes: “OperationRequest_Initiator” which extends “AchieveREInitiator” and also “OperationRequest_Responder” which extends AchieveREResponder.

Bus Agent acts as an initiator in this protocol and therefore uses “OperationRequest_Initiator” behavior while all RPC Agents acting as participants have “OperationRequest_Responder” behavior.

Everything starts with a Bus Agent that while being in the state PREPARE_REQUESTS of “OperationRequest_Initiator” checks whether there was a confirmation from winning RPC Agents to change their reactive power output. If not confirmed – there will be composed empty messages with no recipients that after trying to send them in the SEND_INITIATIONS state will prompt termination of the behavior “OperationRequest_Initiator”. If confirmed – a Bus Agent will compose messages of a type request requesting RPC Agents to change their reactive power output and after entering SEND_INITIATIONS state will send them to these agents. These messages include information on by how much the output needs to be changed for a certain compensator.

After receiving this request, a RPC Agent moves to PREPARE_RESULT_NOTIFICATION state of the “OperationRequest_Responder” since responding with an agree message is not necessary according to this protocol. In this state, a RPC Agent call its method “initiateRequestedAction” that actually operates either a CB in case of a SR and MSC or the thyristor firing angle of a TCR to change its reactive power output. This method returns either a true for a success of an operation or false for a failure.

Then after having moved to SEND_RESULT_NOTIFICATION state, RPC Agent sends inform message into which it includes whether the operation was successful or not.

3 RESULTS

3.1 Experimental setup

3.1.1 Control object model

To test the developed multi-agent system of reactive power compensators control, the RTDS (Real Time Digital Simulator) Simulator was used.

The RTDS Simulator is a real time power system simulation tool used for the closed-loop testing of protection and control equipment, HVDC and FACTS scheme testing, phasor measurement unit (PMU) simulation, power electronics simulation, distributed generation studies, and more.

With be RTDS Simulator there installed I/O cards that allow the Simulator to be interfaced with external equipment such as protective relays.

Models of electric power grid, for their subsequent launch on RTDS, are prepared in the RSCAD which is designed specifically for interfacing to the RTDS Simulator hardware and allows user to view and analyze results.

In RSCAD the model is made using DRAFT module which is a graphical assembly and data input for simulation circuit. The assembled model of the power grid and the substation for this work is shown in the appendix 1.

Also there's an additional module RUNTIME in RSCAD which allows a control and acquiring results from the running model. Part of a RUNTIME model for this project is shown in the appendix 2.

The modeled substations from Siberia Power Grid and their load parameters are given in the appendix 3.

A closer look at the modeled “Novoanzherskaya” substation, the group of compensators at which is regarded as a control object is given in the appendix 4.

3.1.2 Compensators control

Control of a SR is done by sending commands to SR’s CB by respective SR Agent whether to open a CB or close.

Control of each SVC at the substation is divided into the control of 4 separate parts:

- A separate control for each of a two MSCs.
- A separate control system for each of a two TCRs.

MSC’s controls is done in the same way as it is for SR, by sending commands by its respective MSC Agent to open or close its CB.

Control systems of four TCRs (two per SVC) are presented in the Fig. 34. A TCR Agents is responsible for controlling its respective TCR by sending analog commands ranging from -1.0 to 0.0 with respect to its nominal reactive power. For instance, -1.0 would mean a nominal power and -0,5 would mean 50 % of the nominal power. These values from -1.0 to 0.0 are then converted into the thyristor firing angles which are then fed into firing pulse generator.

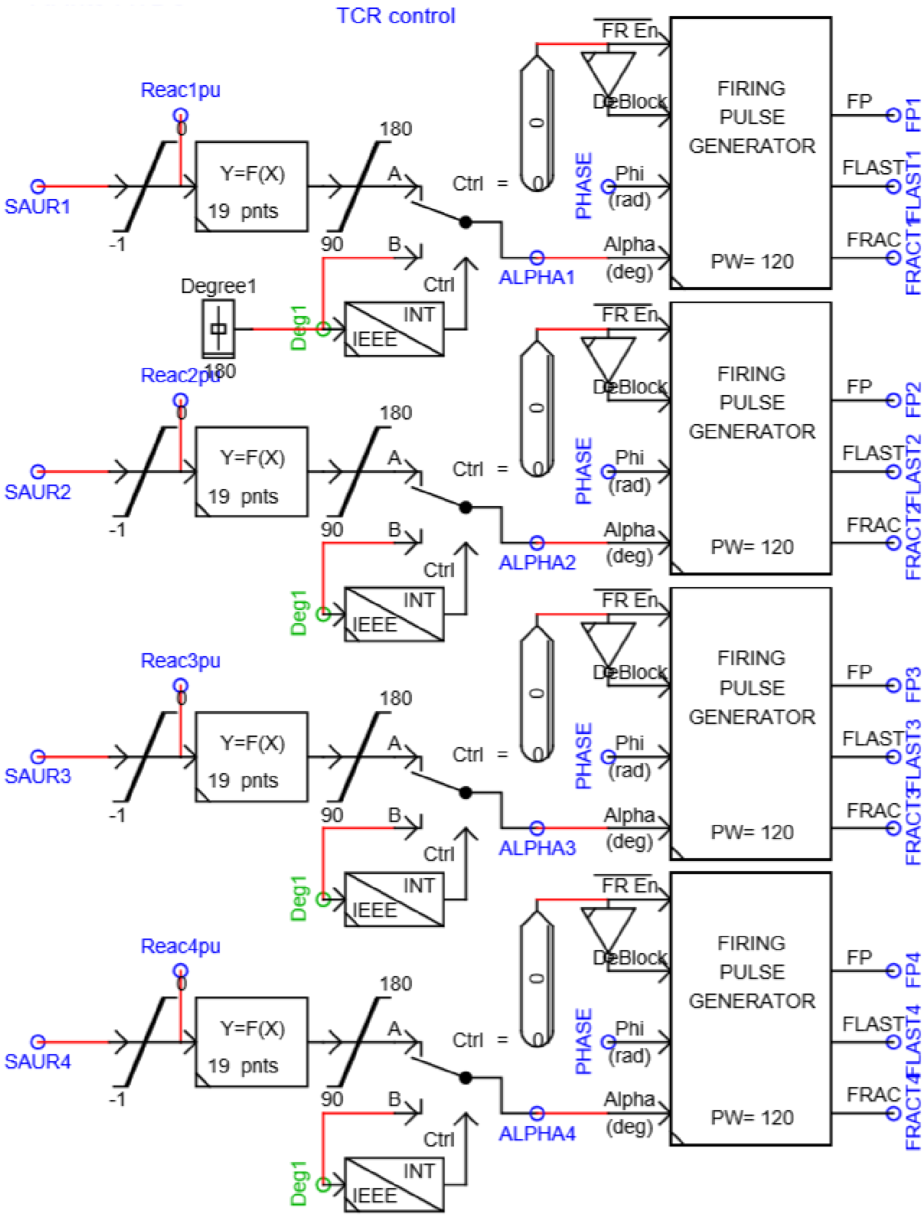


Fig. 34. TCR control system.

3.1.3 Transmission of measurement data and control commands

To test the developed MAS, there should be configured the transmission of the measurement data from RTDS to MAS and control commands from the MAS to RTDS.

Measurement data and commands transmission is done according to the IEC-60870-5-104 protocol. This protocol is restricted to point-to-point and multiple point-to-point configurations [15]. That is why there was created an additional Data Agent acting as an interface that allows transport of the data and commands in accordance with IEC-60870-5-

104 between RTDS and all of the agents in the MAS. All data and commands are passed via Data Agent as show in the Fig. 35. Whenever a certain agent needs data on something it would request a Data Agent to provide that data and whenever, for instance SR Agent wants to close its CB it would request a Data Agent to pass that command to RTDS.

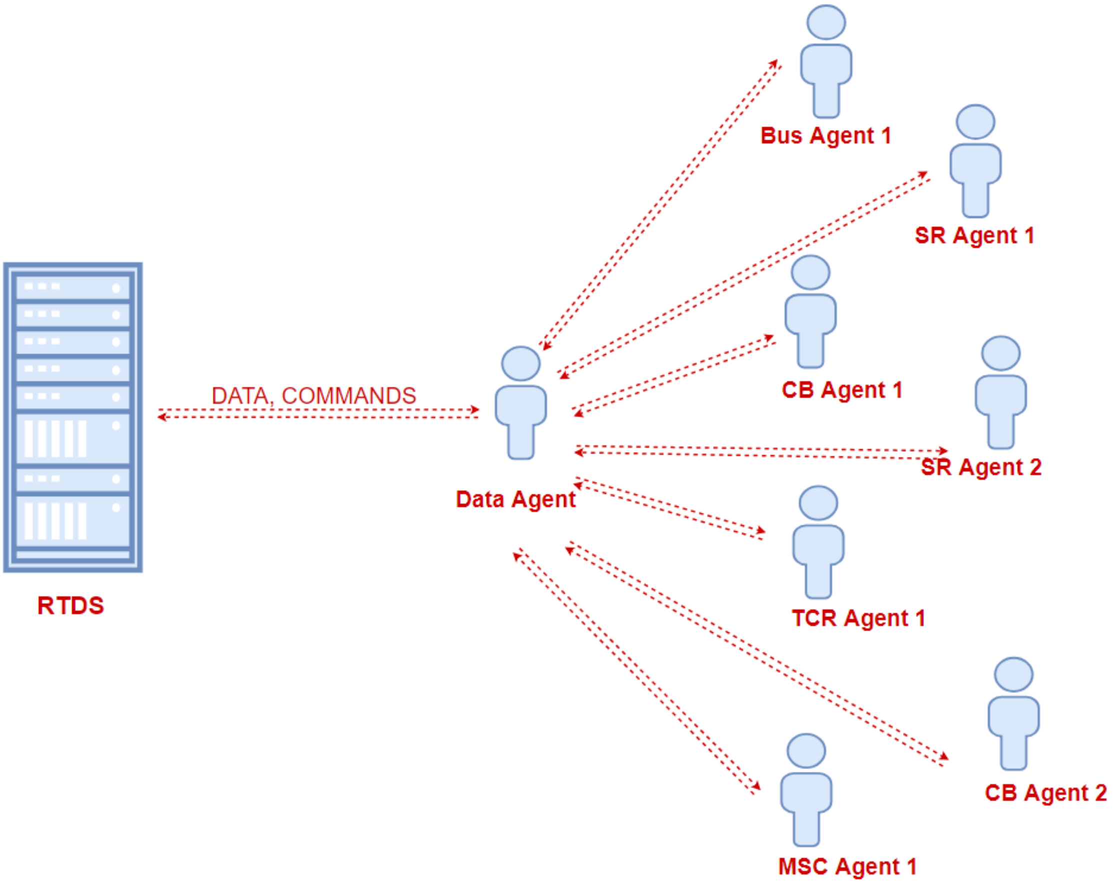


Fig. 35. TCR control system.

In the RSCAD model, the transmission of data and commands via the IEC-60870-5-104 protocol is configured using the GTNET-IEC104 block. The image of this block is shown in Fig. 36.

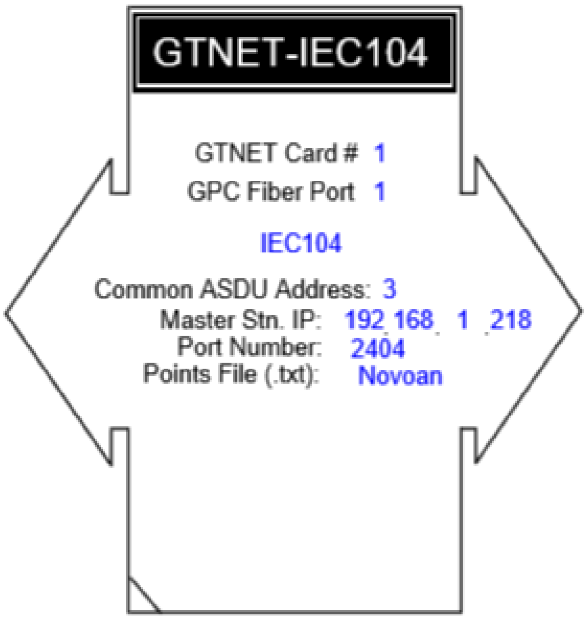


Fig. 36. IEC-60870-5-104 communication configuration block.

Also, there should be composed a special configuration file of ".txt" format, which maps data retrieval and command tags to each input and output data ports in the model. Each agent stores tags that are related to the equipment this agent represents. For instance, CB Agent has a tag to retrieve data about the state of its CB or a TCR Agent has a tag for a current load of the TCR and also a tag to control its load. The content of this file is shown in appendix 5.

Input and output ports from the RSCAD model are presented in the Fig. 37. Command signals displayed in the figure are in the BI (binary input) and AI (analog input) squares. AI are for TCR control and BI is for MSC's and SR's CBs control. AO (analog output) is for the current TCR load and BO (binary output) is for all of the states of all of the CBs modeled within the substation. Binary output port names for CB states match with their respective names in DRAFT model.

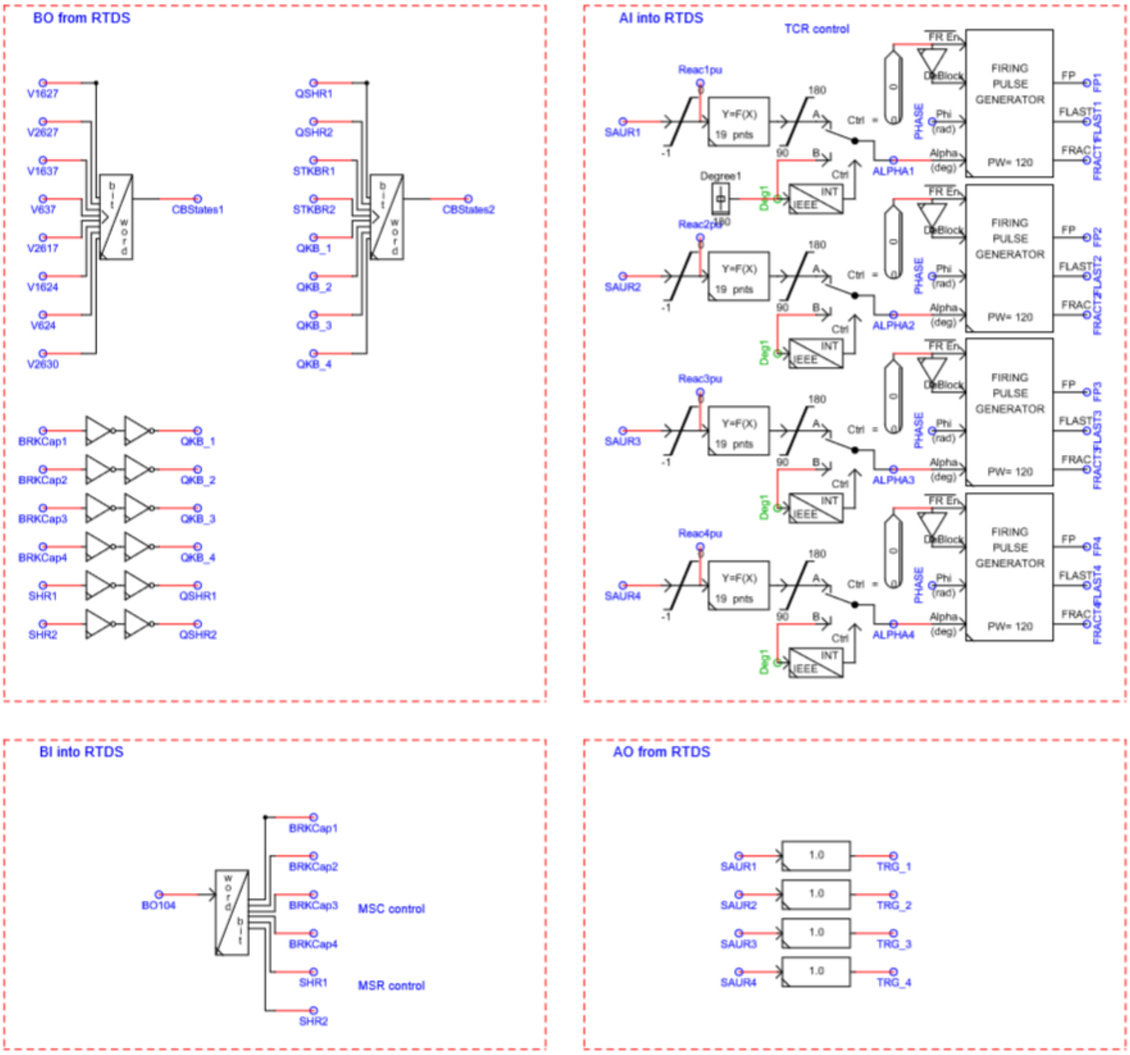


Fig. 37. Input and output ports from the RSCAD model.

3.2 Testing of the devised Multi-Agent System

In the following tests of the devised Multi-Agent System of control of a group of reactive power compensators, there will be shown how Multi-Agent System operates in general and how it adapts to changes in the substation configuration, and how it carries out distribution of reactive power that needs to be compensated across different compensators at the substation according to the current substation configuration.

Default configuration of the substation is shown in the Fig. 38.

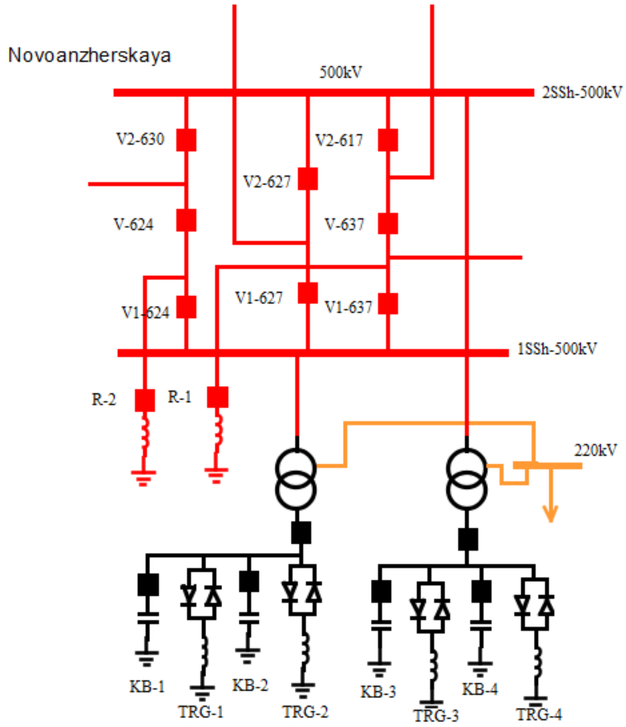


Fig. 38. “Novoanzherskaya” substation.

Voltage will be controlled at the busbar “1SSh-500 kV” and the reference voltage that is maintained at the busbar – 507 kV. MAS of compensators control is activated after the steady state in the simulation is established with no compensators deployed.

And also for simplicity reasons the part of substation with nominal voltage of 220 kV, 110 kV and 10 kV are not considered. Thus, this equipment is also not represented by agents. This means that when busbars “1SSh-500 kV” and “2SSh-500 kV” are not connected via CBs on the 500 kV side, different set of compensators is available for controlling the voltage at the “1SSh-500 kV” busbar.

When testing the developed system, 2 scenarios with respect to single line diagram configuration are considered. These two scenarios will demonstrate how MAS adapts to the changes in the configuration and identifies all of the compensators that are connected to the bus voltage on which is controlled by its respective agent.

1. Scenario 1 - When the following CBs are opened:
    - “V1-627”
    - “V-637”

- “V-624”

  That means there’s no connection between busbars “1SSh-500 kV” and “2SSh-500 kV”. Thus it is expected that Bus Agent of the “1SSh-500 kV” will only utilize compensators that are connected to the busbar that is to say MSCs “KB-1” and “KB-2”, and TCRs “TRG-1” and “TRG-2”, and SRs “R-1” and “R-2”. And the bidding process will be carried out among the agents that represent these compensators.

2. Scenario 2 - When the following CBs are opened:
   - “V1-627”
   - “V-637”

   That means there’s connection between busbars “1SSh-500 kV” and “2SSh-500 kV”. Thus it is expected that Bus Agent of the “1SSh-500 kV” will utilize all of the compensators since now they are all connected to the busbar that is to say. And the bidding process will be carried out among all of the reactive power compensator agents.

Plots of the voltage at the busbar “1SSh-500 kV” and the loads of all compensators for scenario 1 are shown in the Fig. 39.

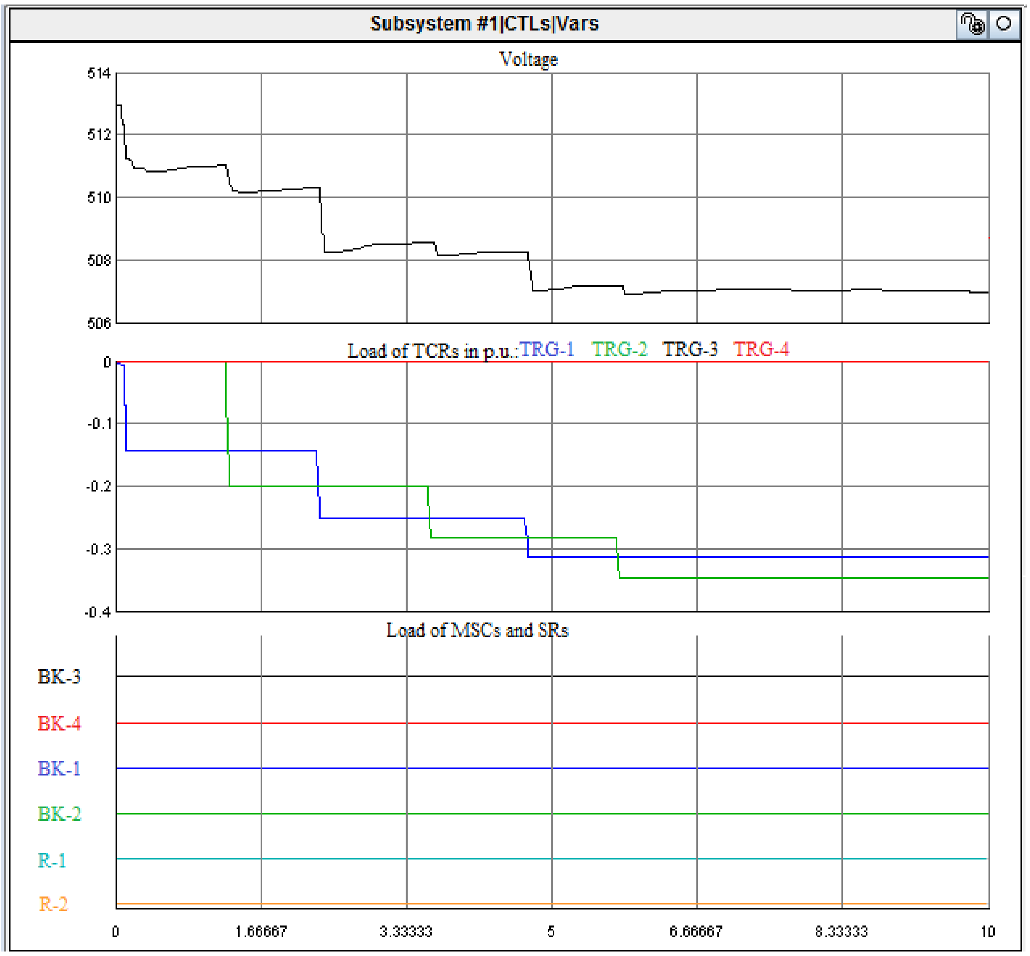


Fig. 39. Voltage and compensator loads plots. Scenario 1.

The plot at the top of the figure illustrates how voltage comes to the reference value as compensators change their loads as a result of MAS distributing the reactive power across all available compensators. The plot in the middle shows how load of the TCRs at the substation changes. Namely, it can be seen that only the load of TCR “TRG-1” and “TRG-2” is changed as they are the only TCRs connected to the busbar according to the single line diagram under consideration.

A plot of the voltage at the busbar “1SSh-500 kV” and the loads of all compensators for scenario 2 are shown in the Fig. 40.

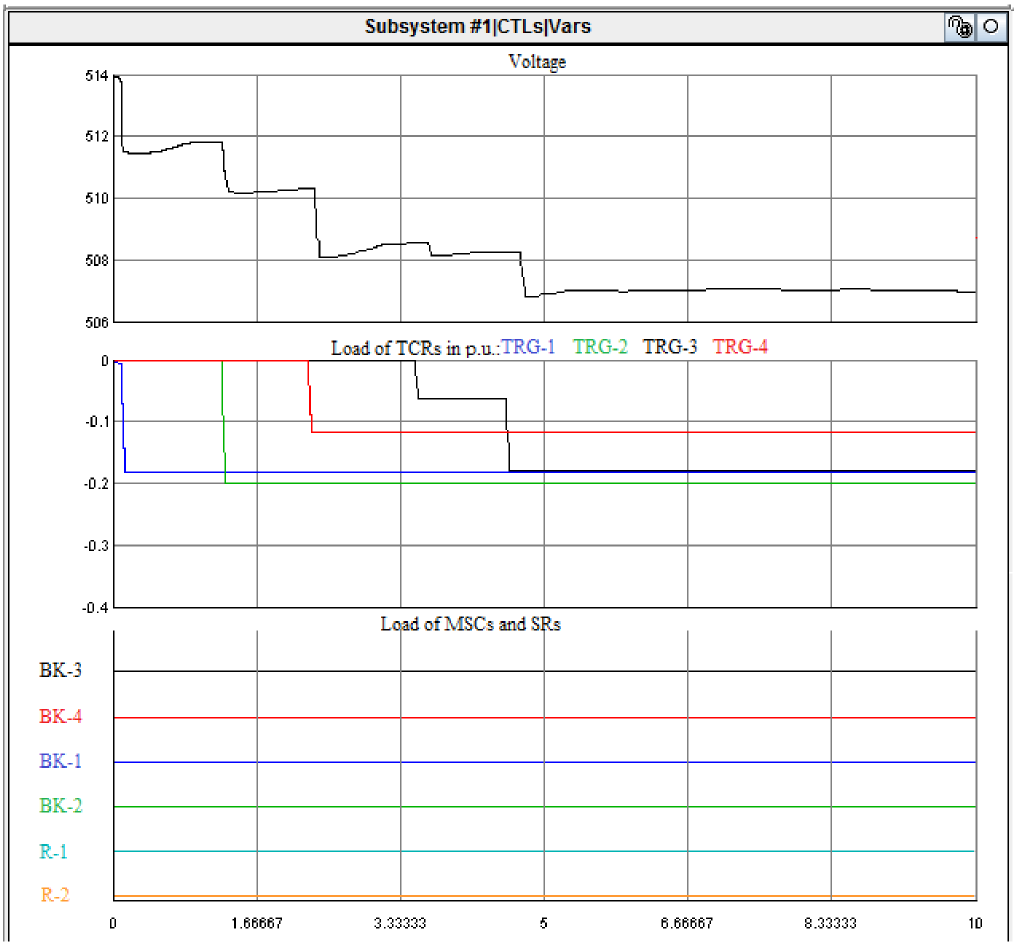


Fig. 40. Voltage and compensator loads plots. Scenario 2.

From this figure it could be seen that since now there is a connection between busbars "1SSh-500 kV" and "2SSh-500 kV" which now means all TCRs are connected to the busbar "1SSh-500 kV", all of the TCRs are taking part in maintaining the voltage. This shows adaptability of MAS to changes in the substation single line diagram.

As for SRs and MSCs, in both scenarios they are not operated because during the bidding process the combination of compensators involving any of these compensators would also involve some of the TCRs which is less favorable or costly in bidding process terms as opposed to deploying just a TCR when it is within the regulation limits.

To illustrate the interaction of agents, the sniffer monitoring mode was used in the JADE human-machine interface, which allows viewing how agents interact with each other, what kind messages they send etc. And also it is a very helpful tool for debugging any MAS.

A diagram of the interaction of agents for the scenario 1 is shown in Fig. 41. For the simplicity and observability reasons, Data Agent was excluded from the monitoring so there are less excessive messages and it is clearer to view the interaction patterns.

In a diagram there could be seen four distinctive sections of agent's interaction. The first one is search of connected to a busbar RPC Agents. The second section is a part when connected RPC Agents are engaged in a bidding process. It should be noted that since busbars "1SSh-500 kV" and "2SSh-500 kV" are not connected via CBs on the 500 kV, reactive power distribution bidding process is carried out only among those that are connected to "1SSh-500 kV" which are:

- MSC "KB-1"
- MSC "KB-2"
- TCR "TRG-1"
- TCR "TRG-2"
- SR "R-1"
- SR "R-2"

After the bidding process is completed, the confirmation part could be seen on the diagram. And the last section is operation request.

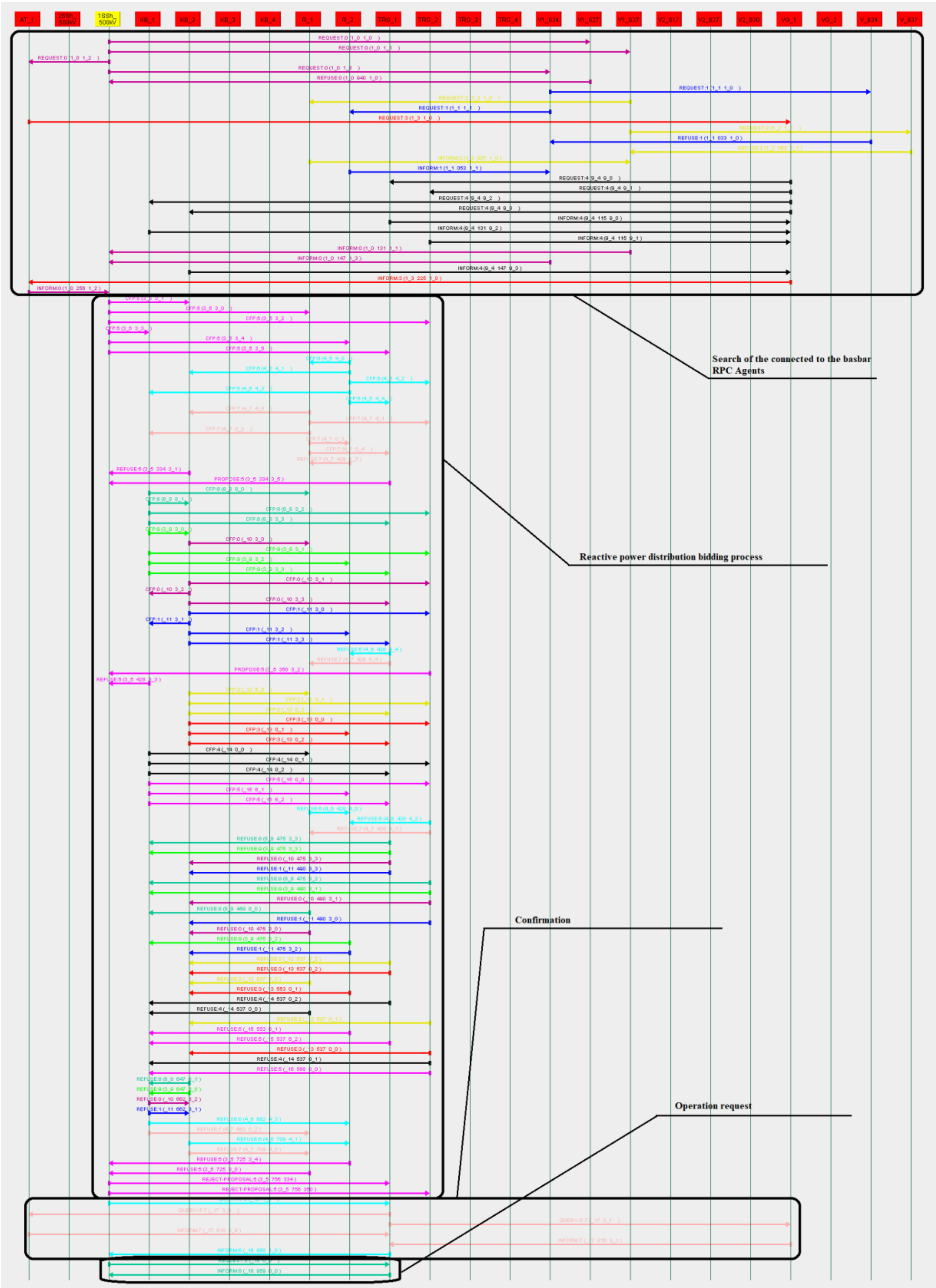


Fig. 41. Agents' interaction diagram. Scenario 1.

Also for the sake of demonstration of a content of the messages being sent between agents, the call for proposal message which is related to the reactive power distribution bidding process that is sent by the “1SSh-500 kV” Bus Agent to “TRG_2” TCR Agent is shown below.

```
((AsksForQ -0.01
     (sequence
       (agent-identifier
         :name R_1@127.0.0.1:1201/JADE
       (agent-identifier
         :name KB_2@127.0.0.1:1201/JADE
       (agent-identifier
         :name TRG_2@127.0.0.1:1201/JADE
       (agent-identifier
         :name KB_1@127.0.0.1:1201/JADE
       (agent-identifier
         :name R_2@127.0.0.1:1201/JADE
       (agent-identifier
         :name TRG_1@127.0.0.1:1201/JADE))))
```

As it can be seen, the content is pretty elegant and easily can be understood by a human as well as an agent that shares the same ontology with the sender of this message. In this example, Bus Agent requests reactive power of a 0.01 MVAR to be absorbed from a TCR Agent and there is also a list of other RPC Agents that are connected to this Bus Agent’s busbar. The list of other RPC Agents is needed for a TCR Agent to know whom it can send its own calls for proposals should it be unable to compensate reactive power asked by Bus Agent.

For the second scenario agents’ interaction diagram is shown in Fig. 42 and Fig. 43.

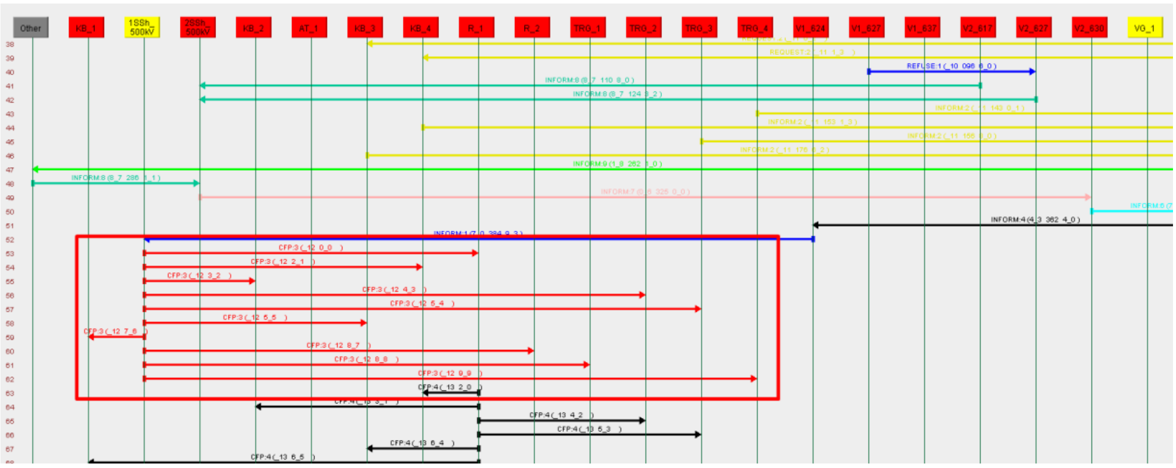


Fig. 42. Agents’ interaction diagram. Scenario 2. 1/2.

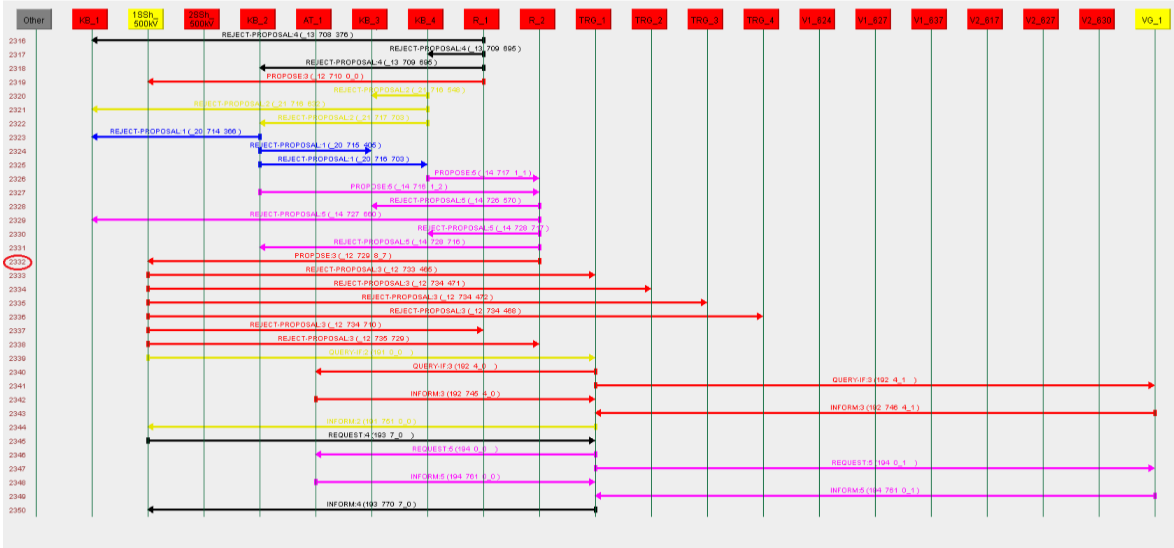


Fig. 43. Agents' interaction diagram. Scenario 2. 2/2.

This time it could be seen that a bidding process is carried out among all RPC Agents. Although, as it is shown in these figures, a bidding process involves a significant amount of messages which is also takes more time. The first cfp message is labeled with a number 53 and the last proposal received is a number 2332.

4 DISCUSSION AND CONCLUSIONS

There have been developed a Multi-agent system for control of a group of reactive power compensators at substation in order to maintain the voltage level on a busbar within the acceptable limits. In this Multi-agent system every power equipment is represented by its own agent which has its own objectives and functionalities. An agent that represents a busbar is responsible for maintaining the voltage level at its busbar, while the agents that represent compensators are responsible for providing the reactive power. Therefore, in order for a voltage at the busbar to be kept within the acceptable limits, an agent that represents a busbar collaborates with reactive power compensator agents. This collaboration includes a reactive power bidding process which is initiated by a busbar agent and all compensator agents are engaged in it. In this bidding process busbar agent solicits proposals for providing a reactive power from the compensator agents. These proposals made by the compensator agents include the cost which is formed based on what would be the power losses, what would be the total harmonic distortion which is applicable for thyristor-controlled reactors, and whether any circuit breaker should be operated. The cost would be higher if providing requested reactive power leads to an increase in power losses or increase in the total harmonic distortion, or opening or closing a circuit breaker. Therefore, when deciding what equipment needs to be operated in order to compensate some reactive power, a bidder with lowest cost is chosen. Thus, this bidding process leads to the following features:

- Reducing power losses and improving its quality.
- Even use of device's resources with regards to a CB on-off cycles thus increase in their lifespans.

Above all, agents that represent their equipment know of what other equipment its own equipment is connected to. Thus by interacting with one another agents can figure out which compensators are connected to the busbar and which compensators are currently in service. This way the adaptability to changes in the substation configuration diagram and in the number operating compensators at any given time is achieved.

In the previous section the developed system was tested upon a simulation model in Real-Time Digital Simulator and from the carried out tests, the following conclusions can be drawn:

1. Multi-agent system of compensators control can dynamically determine the available set of compensators that can be deployed for controlling the voltage at a certain busbar at any given time. This includes adaptability to changes in the group of compensators currently in operation that is to say when a certain compensator is put under maintenance, the multi-agent system will adapt to it and will work with what it has got. Also it includes adaptability to changes in the substation configuration in terms of the states of all circuit breakers, that might affect which compensators are connected to the busbar. In this case, multi-agent system will also adapt to these situations and will be using only connected to the busbar compensators.
2. The developed ontology facilitating the agent communication proved to be very effective and extremely convenient for viewing the content of messages to humans.
3. The developed multi-agent system for control of a group of reactive power compensators, however, turned out to be relatively slow which is why voltage was controlled at such a slow sample time of 500 ms. The reasons as to why developed multi-agent system was relatively slow are discussed below.
4. Search of the connected to the busbar compensators can take a long time. However, it is primarily due to the fact that for implementation of this functionality there have been used ready-made protocol behaviors provided by Java Agent Development Framework that put a lot of restrictions on how agents should interact with one another that leads to excessive messages being sent. For instance, it would be better if once the request messages have reached reactive power compensator agent, an agent would then send notification straight to the agent that represents a busbar. But since this protocol is used there have to be responses sent back to the initiator to comply with a protocol.
5. As for the bidding process, if there are many compensator agents taking part as it was shown in the testing of a scenario 2, the amount of messages being sent drastically increases which in turn slows down multi-agent decision making. That is why there should be included deadlines for receiving responses that would ensure a fixed time compliance.
6. The use of a single Data Agent that provides simulation data to all agents also puts some restrictions on the speed of the multi-agent system decision making since all data is acquired from a single agent. In reality, it should be made so that each agent has access to the data related to its own equipment without any intermediaries.

5 SUMMARY

In this thesis, there have been designed and implemented a multi-agent system for control of a group of reactive power compensators installed at substation “Novoanzherskaya” from Siberia Power Grid in Russia. Multi-agent system was built using JADE and tested upon a simulation model of the substation and an adjacent power grid in question that were built in the RSCAD software and run on the RTDS. The communication between RTDS and Multi-agent system was done in accordance with IEC 60870-5-104 protocol.

All of the requirements outlined in the thesis objectives for the multi-agent system have been met.

The developed multi-agent system proved to be resilient to changes in the substation configuration diagram and in the number operating compensators at any given time. Also, decision making of multi-agent system when it comes to deciding which compensators need to operated at any given time is aimed at:

- Reducing power losses and improving its quality.
- Increase in the lifetime of the circuit breakers of compensators such as shunt reactors and mechanically switched capacitors.

However, the developed multi-agent system turned out to be relativeley slow due to the reasons described in the discussion and conclusion section.

The work on multi-agent system for control of a group of compensators will be continued and outlined flaws will be eliminated in the future work.

REFERENCES

[1] M. Tümay, S. Bal, İ. Rahmi, B. Dogru, and A. Mahmut, ‘A review of magnetically controlled shunt reactor for power quality improvement with renewable energy applications’, *Renew. Sustain. Energy Rev.*, vol. 77, no. August 2016, pp. 215–228, 2017.

[2] P. Neelesh, ‘Magnetically controlled shunt reactors – a new type of FACTS device’, *energize*, no. May, pp. 27–29, 2014.

[3] L. Grigsby, Ed., *Electric Power Generation, Transmission, and Distribution, Third Edition*, Third. Boca Raton: CRC Press, 2011.

[4] B. M. Weedy, D. J. Cory, N. Jenkins, J. B. Ekanayake, and G. Strbac, *Electric Power Systems*, Fifth. Chichester: A John Wiley & Sons Ltd, 2012.

[5] A. Gomez-Exposito, A. J. Conejo, and C. Canizares, Eds., *Electric Energy Systems Analysis and Operation*. Boca Raton: Taylor & Francis Group, 2009.

[6] M. Wooldridge, *An Introduction to MultiAgent Systems*. Chichester: JOHN WILEY & SONS, LTD, 2002.

[7] S. D. J. Mcarthur *et al.*, ‘Multi-Agent Systems for Power Engineering Applications — Part I : Concepts , Approaches , and Technical Challenges’, *IEEE Transacations power Syst.*, vol. 22, no. 4, pp. 1743–1752, 2007.

[8] I. Damjanovic, F. Tomasevic, I. Pavic, B. Filipovic-Grcic, and A. Zupan, ‘Harmonic Performance Analysis of Static Var Compensator Connected to the Power Transmission Network’, *Cigre*, no. June, p. 11, 2018.

[9] *IEEE Loss Evaluation Guide for Power Transformers and Reactors*. USA, 1992, p. 28.

[10] F. Bellifemine, G. Caire, and D. Greenwood, *Developing multi-agent systems with JADE*. Chichester: JOHN WILEY & SONS, LTD, 2007.

[11] *FIPA Request Interaction Protocol Specification*. Switzerland, 2002, p. 8.

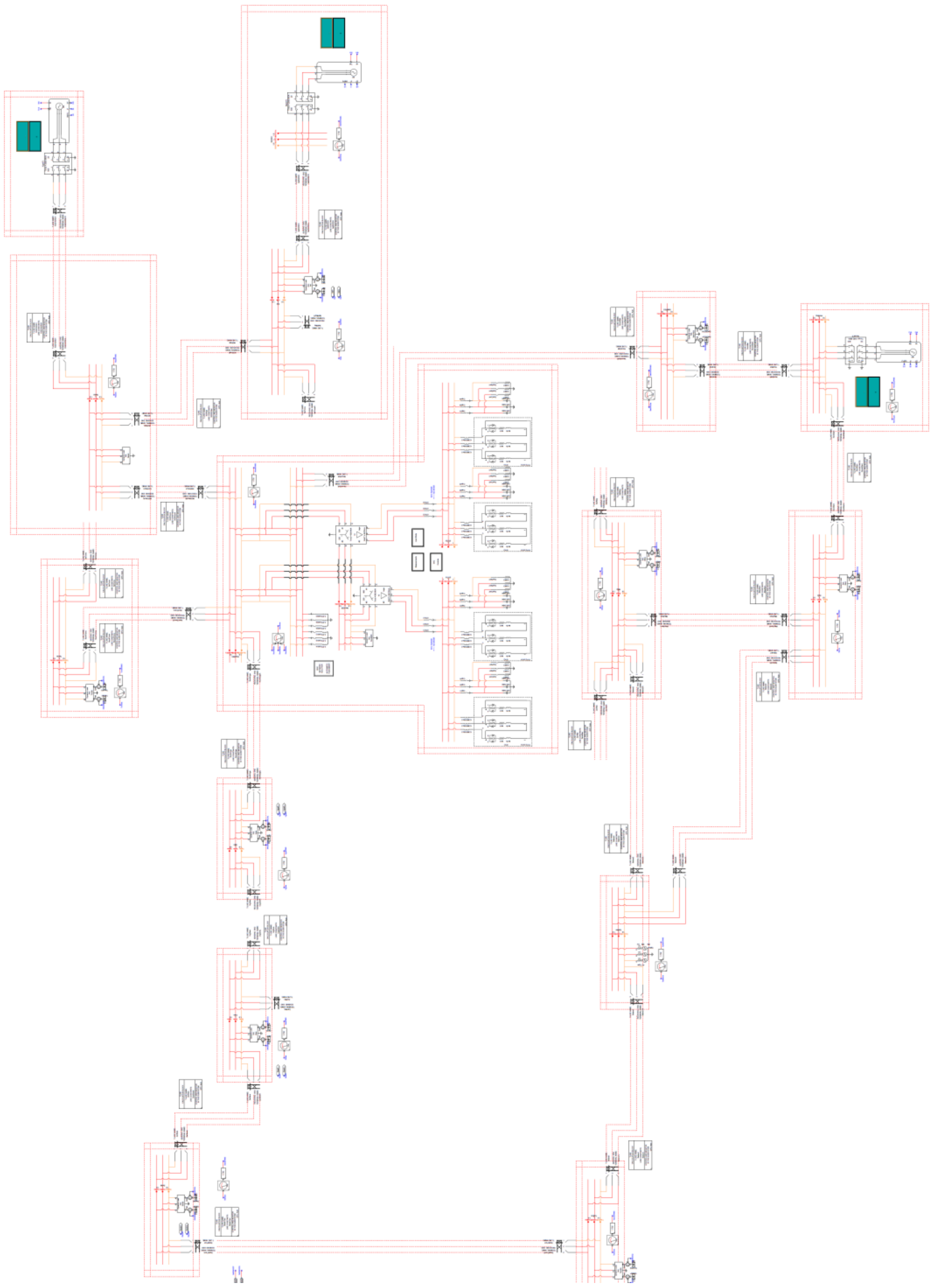
[12] M. Dastani, J. Dix, and A. E. F. Seghrouchini, ‘Programming Multi-Agent Systems’, 2003, p. 230.

[13] *FIPA Contract Net Interaction Protocol Specification*. Switzerland, 2002, p. 9.

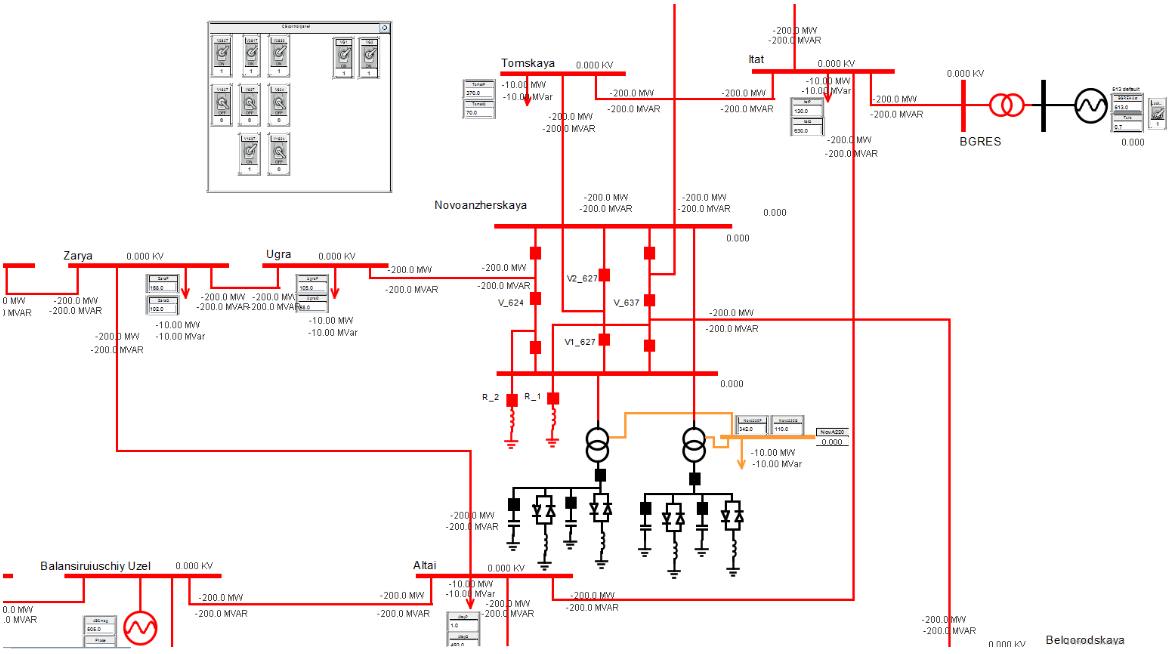
[14] *FIPA Query Interaction Protocol Specification*. Switzerland: FIPA TC Communication, 2002, p. 8.

[15] P. Matoušek, ‘Description and analysis of IEC 104 Protocol Technical Report’, Brno, 2017.

APPENDIX 1. DRAFT model of the power grid and substation in RSCAD.



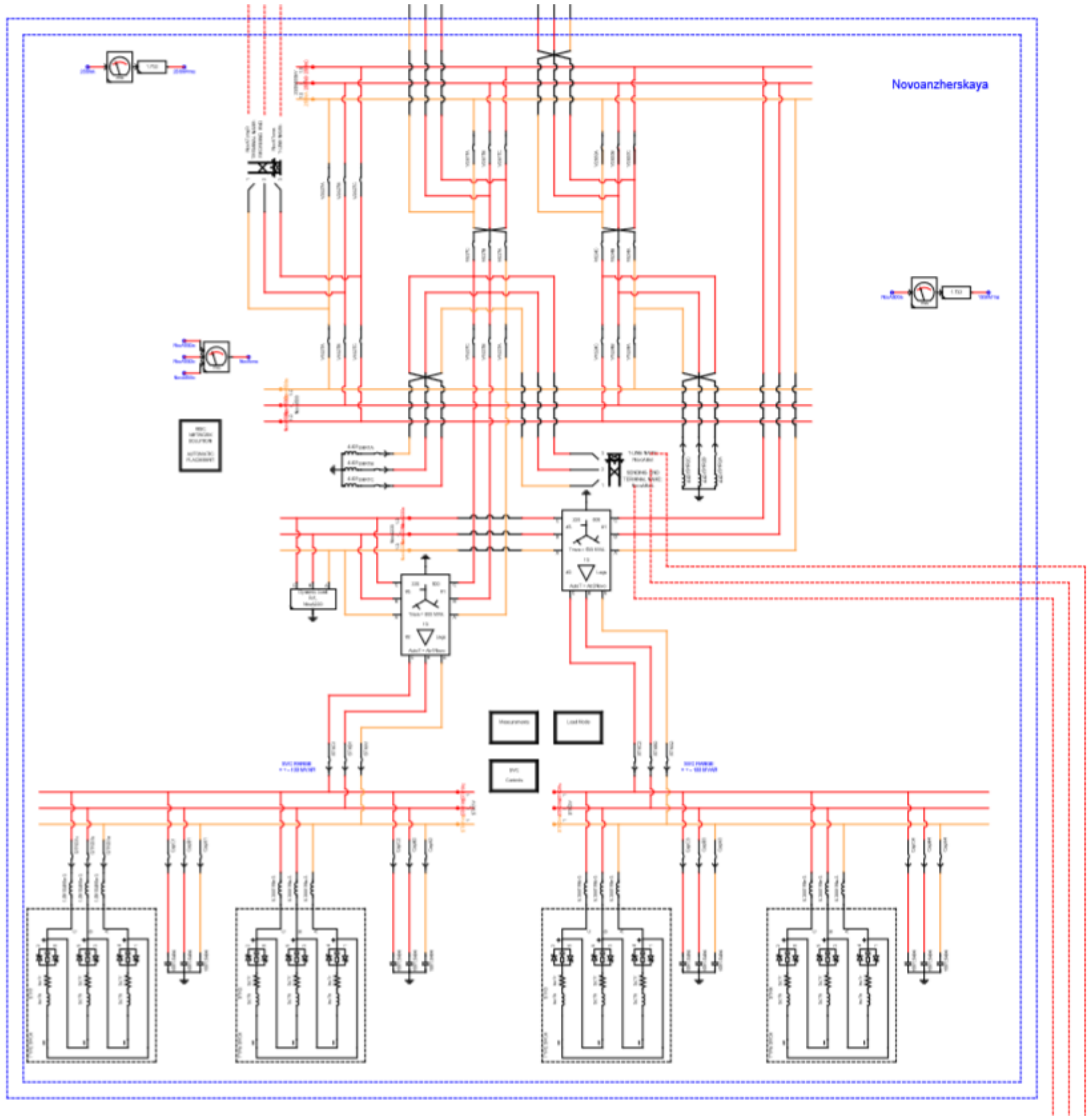
APPENDIX 2. RUNTIME model of the power grid and substation in RSCAD.



APPENDIX 3. Modeled substations and their load parameters in RSCAD.

Substation name	Active power, MW	Reactive power, MVAR
Novoanzherskaya 500 kV	342	from -540 till 200
Ugra 500 kV	105	58
Zaria 500 kV	168	102
Barabinskaya 500 kV	130	220
Tavricheskaya 500 kV	414	490
Barnaulskaya 500 kV	1	493
Novokuznetskaya 500 kV	361	275
Belgorodskaya 500 kV	0	0
Nazarskaya 500 kV	121	216
Itat 500 kV	0	0
Tomskaya 500 kV	130	630

APPENDIX 4. Modeled "Novoanzherskaya" substation in RSCAD.



APPENDIX 5. Configuration file for data retrieval from the model and transmission of commands to the model.

```
BI:   0    V1_627     CBStates1     0
BI:   1    V2_627     CBStates1     1
BI:   2    V1_637     CBStates1     2
BI:   3    V_637      CBStates1     3
BI:   4    V2_617     CBStates1     4
BI:   5    V1_624     CBStates1     5
BI:   6    V_624      CBStates1     6
BI:   7    V2_630     CBStates1     7
BI:   8    QR_1       CBStates2     0
BI:   9    QR_2       CBStates2     1
BI:   10   VG_1       CBStates2     2
BI:   11   VG_2       CBStates2     3
BI:   12   QKB_1      CBStates2     4
BI:   13   QKB_2      CBStates2     5
BI:   14   QKB_3      CBStates2     6
BI:   15   QKB_4      CBStates2     7

AI:   0    TRG_1     0.2%
AI:   1    TRG_2     0.2%
AI:   2    TRG_3     0.2%
AI:   3    TRG_4     0.2%
AI:   4    1SShVrms  0.0002%
AI:   5    2SShVrms  0.0002%

AO:   0    SAUR1     0.0
AO:   1    SAUR2     0.0
AO:   2    SAUR3     0.0
AO:   3    SAUR4     0.0

BO:   0    KB_1 BO104     0    0
BO:   1    KB_2 BO104     1    0
BO:   2    KB_3 BO104     2    0
BO:   3    KB_4 BO104     3    0
BO:   4    R_1  BO104     4    0
BO:   5    R_2  BO104     5    0
```