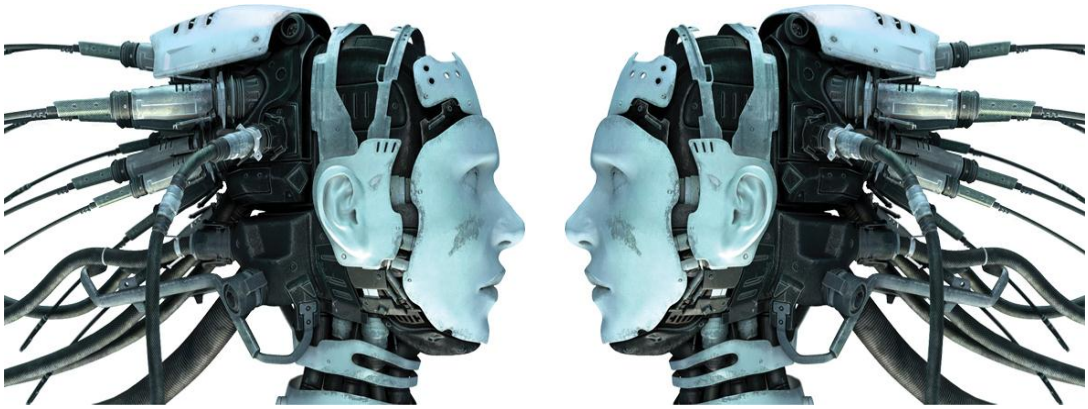
V.A.I.C™

VIRTUAL ARTIFICIAL INTELLIGENCE CHARACTER



CODENAME REVOLUTION™

DOCUMENTATION

CONTENTS

❖ ABOUT V.A.I.C™ PROJECT

Introduction
Theory
Project contributions and facts
Functional basics of V.A.I.C™
Current client program (V.A.I.C™ Codename Revolution™)
Current server program (V.A.I.C™ Global Server™)
Summary

❖ ABOUT MYSQLCR SERVICE

Install MySQLCR Service
Reinstall MySQLCR Service
Uninstall MySQLCR Service

❖ CODENAME REVOLUTION™ FEATURES

Visual introduction
Main features
Basic features for a dialogue with the program
Advanced features for a dialogue with the program
Additional features

❖ CODENAME REVOLUTION™ TRAINING

Before training
Program training with the responses to the user input
Program training with the synonyms and word corrections
Program training with the context keywords
Program training with the contextual responses
Codename Revolution™ commands
Codename Revolution™ special parameters
Integrated help system

ABOUT V.A.I.C™ PROJECT

Introduction

Are you interested in artificial intelligence? How do you understand this concept? There are many types of artificial intelligence, and it can be seen in movies, you can read about it in the internet or in scientific literature books etc. Everyone has his/her own theory or understanding of it. We are working on creation of our type of artificial intelligence and we have named it V.A.I.C™ - Virtual Artificial Intelligence Character. It is a computer program which is getting information from a person in a dialogue. Basing on the analysis of this information, the program can answer to the person by using the necessary communicational means (e. g., chat programs or graphic interface), which enable the possibility of this conversation. The main goal of the program is to learn all the possible information from a person and to create its own personality based on the acquired information.

Theory

While the information technologies in the world still develop, also people lifestyle becomes significantly dependent on them. People use the computer not only for their work but also for entertainment and in communication. Most people use chat or e-mail for these purposes. If a person is looking for new contacts or just wants to have some fun, he or she usually uses chat. We launched this project with the goal to invent a computer program which would be like a universally knowledgeable person. The program has to be openly available in the internet so that people using chat or another way of communication would be able to engage in conversation with the program. Basically, the program should carry out dialogues with people and give the required information. It would also be possible to create virtual assistant or psychologist from the program and to adjust it so it would be able to perform specific diagnostic tests.

Project contributions and facts

The current project V.A.I.C™ will help to understand options of the artificial intelligence. V.A.I.C™ Artificial Intelligence kernel should communicate with people and analyze their personalities. The program has to create its own personality by combining these personalities. Of course, there is a lot of work to be done in order to create such a program. Every new algorithm has to be embedded in the program and tested. When the algorithm is introduced, the program should be published in the internet in order to train it and otherwise communicate with it. It is important that the instruction process of the program is carried out by people who belong to different groups: philosophers, psychologists, poets, otherwise-minded people etc. All the shortcomings of the program can be discovered only in a longer period of time. No doubt each new algorithm extends the required testing time. First, the program is tested in the workgroup among the programmers and only afterwards it is published.

Also it has to know at least 10 000 reactions (questions and answers) before it can be published. The most appropriate moment for its publishing is when the program already knows 100 000 reactions, as it is harder to discover that it is a program.

Functional basics of V.A.I.C™

V.A.I.C™ consists of two parts – the client console and the global server. It is called global server, because in contrast with clients it can be only one or few. In case they are few, they are linked together. The client is the Artificial Intelligence which carries out different logical and interactive actions with information. It has an individual knowledge database which corresponds to the virtual personality. This is the part of the program one person or several people simultaneously can communicate with. The program remembers every person it has communicated with and keeps records in its memory about every one of them. It allows continuing conversation with the same person also after a longer period of time. The server receives all the information processed and received by the client. It analyzes the incoming information and creates united (global) knowledge base. Afterwards the other clients are able to receive this information as feedback, thus creating a united knowledge base also for clients. As a result the V.A.I.C™ system has a high efficiency level. Moreover, V.A.I.C™ works with the "MySQL" database server which is one of the most popular database servers in the world and provides high-speed data exchange. It would be impossible to implement this program without this type of server.

Current client program (V.A.I.C™ Codename Revolution™)

Meanwhile we have created a program which will help to analyze the dialogue between the program and a person, program's options and potential errors. In the course of time it will be possible to determine which algorithms should be continued and complemented and which algorithms would be necessary in making the conversation equal to a dialogue between two persons. Current client "Codename Revolution™" can work (can be linked to) with IRC (Internet Relay Chat) server which enables chat between people and program. This type of chat can be integrated in every internet homepage and there a lot of freeware programs enabling it. In our opinion it is the best way of introducing the program to the people for the time being. It is possible to specify all needed parameters in the client console so that V.A.I.C™ could take part in chat. You can configure the main functions in the console, but generally the program can run automatically. There are control buttons in the client console which enable starting, temporary stopping and full shutdown of the program. The program can answer to questions and react to several chat users' questions (sentences) simultaneously. It is also possible to train the program to answer these questions (to program the dialogue) using simple commands, or to make the program try it itself. The program can automatically send the information to the V.A.I.C™ Global Server after learning it and the server can teach it to other V.A.I.C™ clients as a feedback.

Current server program (V.A.I.C™ Global Server™)

Current V.A.I.C™ Global Server program can receive queries about V.A.I.C™ client updating and create these updates. The server program can also receive the newly acquired information from every client which has the server address in its parameters. Server administrator is able to create new updates and to choose when the new information has to be published for all clients. All these functions can be carried out also by the server alone. In other words, the server's main task is to exchange information with all clients.

Summary

Our goal is to develop an interesting and smart information database which would be controlled by artificial intelligence. This program would be able to virtually communicate with every person and in such a way it would develop its own personality. Test and full versions of V.A.I.C™ can be downloaded from our homepage. Currently we are working on the version "Codename Revolution™" which will be improved and complemented continuously. We have our own IRC server where we are testing our programs. Also you can take part in program testing and training. The training instructions are available in the forum and in the program documentation. This is the only project of this type in Latvia (however, this practice is known abroad long ago). We would be glad for any support for our project.

ABOUT MYSQLCR SERVICE

MySQLCR service is a specially configured MySQL server service, necessary in order to run V.A.I.C™ Codename Revolution™ application, providing full functionality. Normally this service is installed automatically running V.A.I.C™ Codename Revolution™ application for the first time. Do not use “Install MySQLCR Service”, „Reinstall MySQLCR Service” or „Uninstall MySQLCR Service” shortcuts from the start menu without any exceptional purpose.

Install MySQLCR Service

Use this tool to install MySQLCR service on your computer, if application fails to install service automatically. After service is installed you should hear one or more beep signals (depending on the OS) in your computer speaker. It means that service is installed successfully.

Reinstall MySQLCR Service

Use this tool to reinstall MySQLCR service on your computer, if application fails to install service automatically. After service is installed you should hear one or more beep signals (depending on the OS) in your computer speaker. It means that service is reinstalled successfully.

Uninstall MySQLCR Service

Use this tool to fully uninstall MySQLCR service from your computer. When uninstalled, V.A.I.C™ Codename Revolution™ application is unable to work properly until MySQLCR service is installed again. We recommend using this tool only if you have problems uninstalling V.A.I.C™ Codename Revolution™ application from your computer.

CODENAME REVOLUTION™ FEATURES

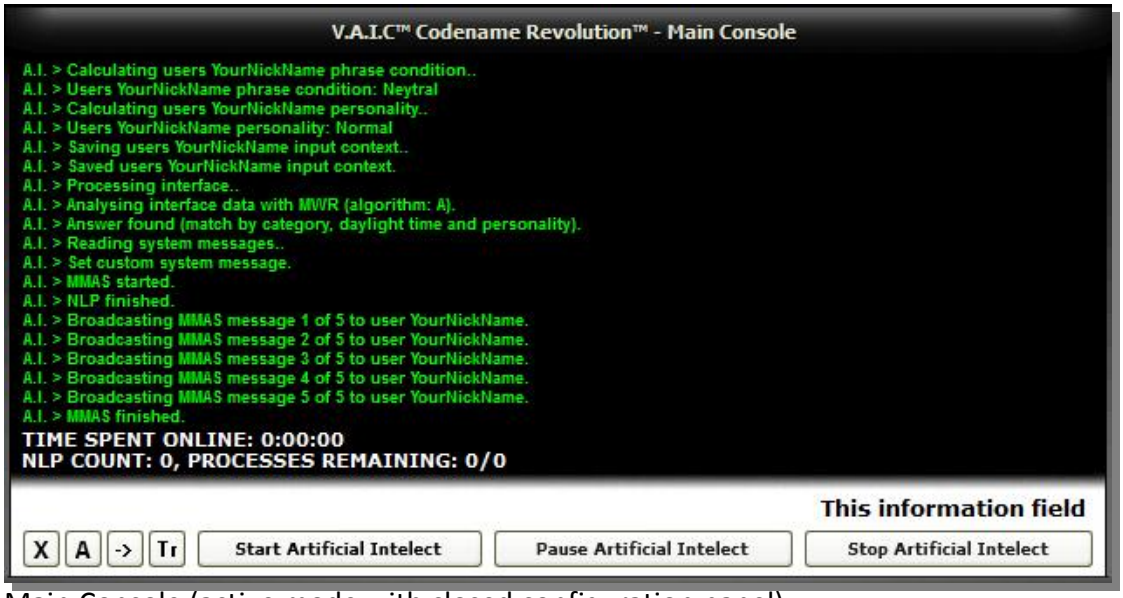
Visual introduction

Here you see the general graphic interface of the program – the layout of the main console information output area and the program options, additional console and the appearance of other program forms.

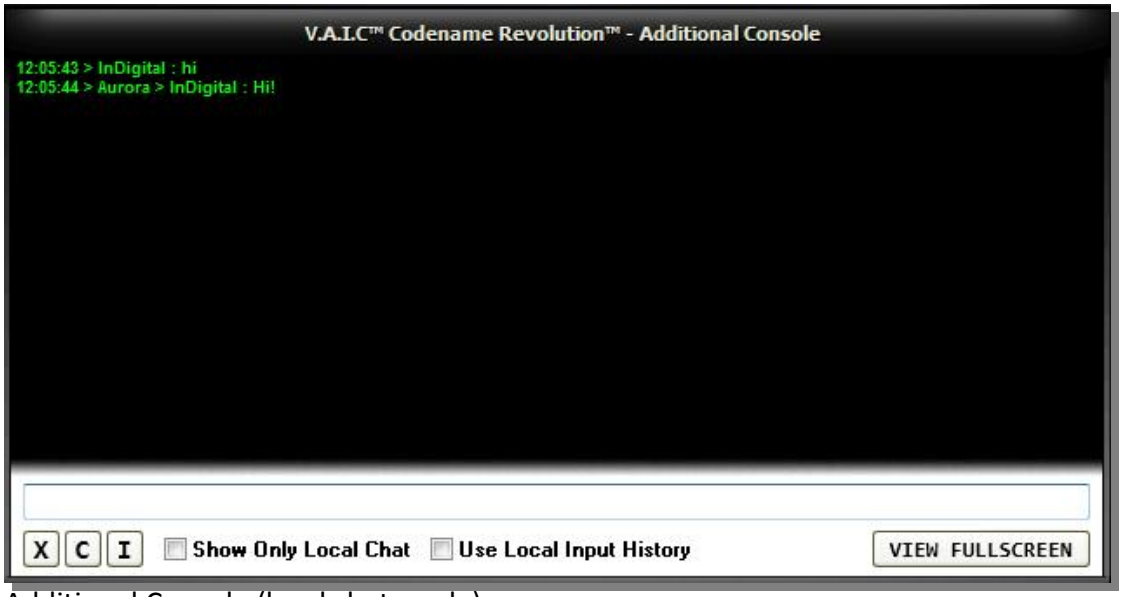
See the appearance of V.A.I.C™ Codename Revolution™ below. In this picture you can see the programs main console in the inactive mode and with open configuration panel.

1) The program logo and information output area (reflects A.I. processes).
2) Information about some program real-time processes.
3) The sliding text help system (explains console options and other objects).
4) The main program control buttons.
5) The configuration panel with grouped configuration tabs (can be hidden).

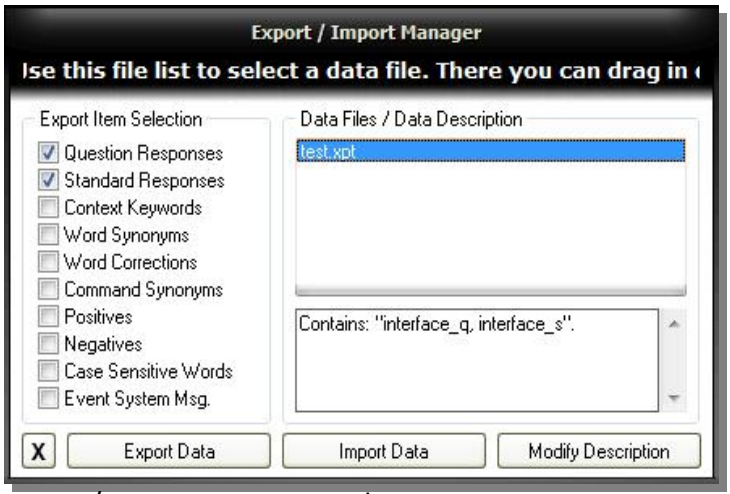
- Some visual elements and options of the program can be added and even changed during program development/improvement.



Main Console (active mode with closed configuration panel).



Additional Console (local chat mode).



Export/Import Manager and Data Restore Box.

Main features

- The program setup is simple. It takes one step to install the program (with a built-in MySQL server containing its database) and to configure the system for operation with the server. The program can be installed in any place, on any hard disk, even if there is other MySQL server on the computer.

- The program interface is neatly designed and it is complemented with skins and sounds. Besides the program contains many options which help to adjust the program's behavior.

- The program can be launched automatically with Windows startup, it can update itself through the internet and it can restore connection with IRC server in case of internet disconnection or failure of IRC server.

- You can export and import the database data, make backup copies for the database, restore the database using the backup, check its errors and clean separate parts of the database using different cleaning criterions (conditions).

- You can chat with the program and train it locally and in the IRC server. When the program runs in IRC server, security terms are initiated. Each user can be assigned with a multilevel trust degree which determines the ways the program can be trained, whether the user can only chat with the program or can be fully ignored.

- The program has built-in options to protect IRC channels from spam, flood, swearwords etc. These functions can be configured, and you can also edit the list of users to whom the general channel rules apply or do not apply.

- The program's one of the main features is filtering of content, which ensures that V.A.I.C™ will act appropriately in IRC public channels. It means that the program cannot be provoked to react inappropriately (spam, flood, repeating and other annoying behaviors), besides it knows when to react on public messages and when to avoid it.

- The possibility to use V.A.I.C™ Global Server™ helps to filter and mutually synchronize information for all V.A.I.C™ clients (programs). It means that clients (with server synchronization switched on) will be able to receive data from other clients. It facilitates progression of common V.A.I.C™ system in the minimal period of time.

Basic features for a dialogue with the program

- Responses of the program can be added, erased and corrected. Besides you can do it easily - in a dialogue with the program.

- Each response of the program can be taught so that the program gives answers in several lines. Thus, it is possible to teach the program a story, poem etc.

- To seem more realistic, the program transforms its responses using synonyms included in the database (both for separate words and sentences). Thus, you do not have to teach the program two similar responses.

- You can teach the program to reply with different (several) variants of responses to each asked question.

- You can enable simulation of writing for the program (if necessary). This option allows simulating natural speed of writing, which depends on the length of sentences and their complexity.

- The program can be used in English and Latvian as well. You can use automatic translation of diacritic marks for Latvian, thus you can chat with the program both using diacritic marks and their translation.

Advanced features for a dialogue with the program

- Each answer taught to the program is stored in a specific way. The program records the time of response teaching, calculates the possible theme and personality of the person. This allows for the program to reply with the responses corresponding to the real time, theme and particular type of personality. It means that a wide range of answers allows the program to chat with every person in a slightly or even considerably different manner.

- The program is designed to respond according to context. It means not just responding to a particular question with a particular answer, but also an ability to discuss a theme accordingly to the entered information by the program's user.

- The program is able to chat with several users simultaneously, making a conversation with each of them about their own (different) topic. Besides the program remembers every user with whom it has chatted. It takes records helping to understand the personality of a particular person (make idea of him/her), and basing on these presumptions the type of further communication is determined.

- To search answers in the database, the program uses the algorithm of multilevel search. It enables the program responding to every sentence with a most effective answer in case an exact answer does not exist. It means the program selects an answer which is the most appropriate according to the current context and subject.

- Using Smart Input Correction algorithm, the program first corrects the written errors in the entered sentences, facilitating the program's ability to understand and process these sentences better.

Additional features

- The program has two consoles – main console and additional console. Main console reflects processes of the actions incoming from IRC server and local chat. Additional console reflects autonomous processes which mostly do not depend on user actions.

- The program automatically creates chat and NLP (Natural Language Processing) logs of each user. These logs can be accessed from Start Menu shortcuts (Logs Directory).

- Integrated help system ensures a possibility (for the IRC server users) to get basic control commands of the program and to receive introducing information about V.A.I.C™.

- The sliding text help system on the program console provides the program's owner with a possibility to see an explanation for any object which is located on the console.

CODENAME REVOLUTION™ TRAINING

Before training

The program should be trained in the form of dialogue. This dialogue consists of user's text input and program's response. Each input is processed by Smart Input Correction algorithm, which modifies the input correcting erroneous and unnecessary characters. The program deprives the text of unnecessary characters leaving only letters and necessary punctuation marks, except when the program recognizes web links which begin with prefixes such as www, http, https, ftp, ftps. Further the program should process the input and try to respond. If the program knows the answer to the input, then the program responds with an existing answer from the knowledge (interface) database. If the program is unable to respond to the input, then the program picks up some response from the event system. Event System is a database which keeps built-in various responses for different situations (program events) and the algorithm that processes these events. All these event messages can be modified in event system message editor which is located on the program console. Each event can store various messages (responses).

Program training with the responses to the user input

Considering the previous information, we can begin program training with responses to user input acquiring program training with one response to user input, various responses to one input and extending existing responses with additional sentences. See an example of teaching responses below:

1) Input some text for the program and wait for the response. If the program does not respond correctly or does not know the answer then…

2) Input desired answer you want to teach to your last input in the following format: #your desired answer (# is a response teaching command).

3) Program will announce that the program has learned the answer.

Now you can input your text again and the program will respond to you with your answer.

EXAMPLE FROM THE LOG FILE 01:
2009.01.17. at 0:25:12 | USER > hello robot.
2009.01.17. at 0:25:13 | A.I. > Sorry but I don't know the answer to your input.
2009.01.17. at 0:25:18 | USER > #hello human!
2009.01.17. at 0:25:18 | A.I. > Thank you for the new information!
2009.01.17. at 0:25:23 | USER > hello robot.
2009.01.17. at 0:25:24 | A.I. > hello human!

The previous example was simple. It is also possible to teach various responses to the one user input. In this case you just need to repeat phase 2. See another example:

EXAMPLE FROM THE LOG FILE 02:
2009.01.17. at 0:40:36 | USER > Who are you?
2009.01.17. at 0:40:37 | A.I. > Sorry but I don't know the answer to your question.
2009.01.17. at 0:40:41 | USER > #I am a robot.
2009.01.17. at 0:40:42 | A.I. > Thank you for the new information!
2009.01.17. at 0:40:48 | USER > #I don't know who I am.
2009.01.17. at 0:40:48 | A.I. > Thank you again.
2009.01.17. at 0:40:53 | USER > #I am your friend.
2009.01.17. at 0:40:54 | A.I. > I am getting smarter.
2009.01.17. at 0:41:03 | USER > Who are you?
2009.01.17. at 0:41:05 | A.I. > I am your friend.

You can also teach program responding with several rows (lines). These rows are called additional sentences. When you are chatting with the program and when it is necessary you can add additional sentences to any of the programs existing responses (except event system messages). In this case you have to use additional sentence teaching command (##).

EXAMPLE FROM THE LOG FILE 03:
2009.01.17. at 1:09:26 | USER > Hi robot!
2009.01.17. at 1:09:27 | A.I. > Hmm.
2009.01.17. at 1:09:28 | A.I. > I don't know what to say...
2009.01.17. at 1:09:33 | USER > #Hello human!
2009.01.17. at 1:09:34 | A.I. > Thank you for the new information!
2009.01.17. at 1:09:45 | USER > ##What is your name?
2009.01.17. at 1:09:45 | A.I. > Thank you for the additional information!
2009.01.17. at 1:09:47 | USER > Hi robot!
2009.01.17. at 1:09:48 | A.I. > Hello human!
2009.01.17. at 1:09:49 | A.I. > What is your name?

In this way you can teach the program responding with poetry or some interesting story. Use your imagination to make programs responses more interesting!

- If you teach response which is very similar to an existing response then program will replace the existing response with the new one. It means correction of the existing response.

- When you are training the program with responses, you should pay attention to the time when the response is taught. For example, teaching the response to the "good afternoon, good evening", it should be done in the corresponding daytime. If you like to teach the program responses that do not concern the time, then this feature can be ignored.

- Considering the fact that the program is a virtual identity, it should maintain a dialogue with people naturally and therefore the best way to implement this is by teaching the program the responses which correspond to your feelings, your nature, current daytime, etc.

Program training with the synonyms and word corrections

Synonyms are necessary for the program to understand better those sentences which have the same meaning. These can be synonyms for the words, phrases and even the program commands. The more synonyms the program knows, the more it is able to recognize similar sentences. Synonyms are understood by the program as the original word/phrase which they are linked to. The minimum length of a synonym that can be taught is limited to three symbols. Synonyms can be taught with simple and short commands. "=" is a learning command and "<>" is the erasing command. The synonym should precede the command and the original word/phrase should follow the command.

EXAMPLE FROM THE LOG FILE 04:
2009.01.17. at 4:26:28 | USER > my homestead=my house
2009.01.17. at 4:26:29 | A.I. > Added new synonym!
2009.01.17. at 4:26:31 | USER > Holy Scripture=bible
2009.01.17. at 4:26:32 | A.I. > Added new synonym!
2009.01.17. at 4:26:33 | USER > Word of God=bible
2009.01.17. at 4:26:34 | A.I. > Added new synonym!
2009.01.17. at 4:26:37 | USER > softheaded=crazy
2009.01.17. at 4:26:37 | A.I. > Added new synonym!
2009.01.17. at 4:26:38 | USER > screwball=crazy
2009.01.17. at 4:26:39 | A.I. > Added new synonym!
2009.01.17. at 4:26:41 | USER > automobile=car
2009.01.17. at 4:26:41 | A.I. > Added new synonym!
2009.01.17. at 4:26:42 | USER > machine=car
2009.01.17. at 4:26:43 | A.I. > Added new synonym!
2009.01.17. at 4:26:50 | USER > Erase this response=[erase]
2009.01.17. at 4:26:51 | A.I. > Added new synonym!
2009.01.17. at 4:26:56 | USER > Erase this response<>[erase]
2009.01.17. at 4:26:57 | A.I. > Erased existing synonym!
2009.01.17. at 4:26:56 | USER > my homestead<>my house
2009.01.17. at 4:26:57 | A.I. > Erased existing synonym!

In the same way you can teach the program to correct wrong words. The wrong word should precede the command and the corrected word should follow the command.

EXAMPLE FROM THE LOG FILE 05:
2009.01.17. at 4:30:34 | USER > corection=correction
2009.01.17. at 4:30:34 | A.I. > Added new word correction definition!
2009.01.17. at 4:30:38 | USER > spupid=stupid
2009.01.17. at 4:30:38 | A.I. > Added new word correction definition!
2009.01.17. at 4:30:42 | USER > stable=stabble
2009.01.17. at 4:30:42 | A.I. > Added new word correction definition!
2009.01.17. at 4:30:51 | USER > stable<>stabble
2009.01.17. at 4:30:52 | A.I. > Erased existing word correction definition!
2009.01.17. at 4:30:57 | USER > spupid<>stupid
2009.01.17. at 4:30:58 | A.I. > Erased existing word correction definition!

Program training with the context keywords

The program has a separate database for words and short phrases like *yes, no, maybe, I don't know, I know, of course, why not*, etc. These words and short phrases can help building context of the dialogue. Let us name these words context keywords or context key-phrases. For example, when user responds with a context keyword to a program's question, the program connects the keyword and its own question creating a unique context condition in the dialogue, to which a response can be taught. The minimum length of a context keyword that can be taught is limited to three symbols. Context keywords can be taught with simple and short commands. ">>" is the learning command and "<<" is the erasing command.

EXAMPLE FROM THE LOG FILE 06:
2009.01.17. at 5:23:31 | USER > yes >>
2009.01.17. at 5:23:32 | A.I. > Added new context keyword!
2009.01.17. at 5:23:36 | USER > no >>
2009.01.17. at 5:23:36 | A.I. > Added new context keyword!
2009.01.17. at 5:23:40 | USER > maybe >>
2009.01.17. at 5:23:40 | A.I. > Added new context keyword!
2009.01.17. at 5:23:49 | USER > why >>
2009.01.17. at 5:23:50 | A.I. > Added new context keyword!
2009.01.17. at 5:23:55 | USER > maybe <<
2009.01.17. at 5:23:56 | A.I. > Erased existing context keyword!
2009.01.17. at 5:24:00 | USER > yes <<
2009.01.17. at 5:24:01 | A.I. > Erased existing context keyword!

- Context keywords that are linked with taught program responses (contextual responses) cannot be simply erased. First you have to erase these linked responses. Of course it is possible to disable this security option in the programs console, but it is not recommended, as thus you can damage dialogue context tree for all other responses that are linked with the erased context keyword. The dialogue context tree can be restored again (automatically), but only when the appropriate program option is set and the erased context keyword is taught again.

Program training with the contextual responses

When the program knows the context keyword, it can be used in program training, to link the context keyword with the previous program response or simply to teach a response to the context keyword itself. While training the program with responses, it always checks, whether the previously entered user input is a context keyword/key-phrase, or not. In case the program detects a context keyword it offers to choose one of the previously mentioned training methods. The contextual training is applicable to any response of the program. Let us look at the example of contextual training.

EXAMPLE FROM THE LOG FILE 07:
2010.08.29. at 9:33:30 | USER > Yes >>
2010.08.29. at 9:33:34 | A.I. > Added new context keyword!
2010.08.29. at 9:33:40 | USER > I like to talk with you robot.
2010.08.29. at 9:33:44 | A.I. > Do you like me?
2010.08.29. at 9:33:49 | USER > Yes.
2010.08.29. at 9:33:53 | A.I. > Sorry, but I can't answer to your input contextually!
2010.08.29. at 9:33:55 | USER > #Nice, I like you too.
2010.08.29. at 9:34:01 | A.I. > The phrase you are teaching me is recorded in my memory as the contextual key-phrase.
2010.08.29. at 9:34:03 | A.I. > Type [A] to teach me the response in the context of the previous sentence.
2010.08.29. at 9:34:06 | A.I. > Type [B] to teach me the response to the contextual key-phrase itself.
2010.08.29. at 9:34:56 | USER > A.
2010.08.29. at 9:35:02 | A.I. > I learned a new response.
2010.08.29. at 9:35:06 | USER > #Ok.
2010.08.29. at 9:35:11 | A.I. > The phrase you are teaching me is recorded in my memory as the contextual key-phrase.
2010.08.29. at 9:35:15 | A.I. > Type [A] to teach me the response in the context of the previous sentence.
2010.08.29. at 9:35:19 | A.I > Type [B] to teach me the response to the contextual key-phrase itself.
2010.08.29. at 9:35:22 | USER > B.
2010.08.29. at 9:35:25 | A.I. > Learning done!
2010.08.29. at 9:35:28 | USER > I like to talk with you.
2010.08.29. at 9:35:30 | A.I. > Do you like me?
2010.08.29. at 9:35:32 | USER > Yes.
2010.08.29. at 9:35:36 | A.I. > Nice, I like you too.
2010.08.29. at 9:35:40 | USER > Yes.
2010.08.29. at 9:35:42 | A.I. > Ok.

- In case the program is unable to respond to a sentence in a context (cannot find a contextual response), it answers with a taught response to the context keyword. If there is no response taught for the context keyword, the program notifies that it cannot respond contextually.

Codename Revolution™ commands

There is a list of commands that can be used through the dialogue with the program. Any of these commands can be replaced with the synonyms. Standard commands (#, ##, =, <>, >>, <<) can be also replaced with the synonyms.

[ERASE]
Use this command to erase programs last response from the memory. It is possible to erase only taught responses. Event system messages can be erased only using event system message editor.

[ERASEALL]
The same as [ERASE], but also all the other response variants referring to the previously entered text/question will be erased.

[ENABLECONVERTER]
Use this command to enable so-called translate function. This function is working only with the Latvian language.

[DISABLECONVERTER]
Use this command to disable so-called translate function. This function is working only with the Latvian language.

[ANSWEROWNER]
Use this command if you want to know the nickname of the teacher of the programs last response.

[STOPLONELY]
Use this command to stop "Lonely Talking" algorithm.

[STOPMAS]
Use this command to stop "MMAS" algorithm.

[POSITIVE]
Use this command to teach the program a positive word.

[NOPOSITIVE]
Use this command to erase an existing positive word.

[NEGATIVE]
Use this command to teach the program a negative word.

[NONEGATIVE]
Use this command to erase an existing negative word.

[CSWORD]
Use this command to teach the program a character-sensitive word.

[NOCSWORD]
Use this command to erase an existing character-sensitive word.

Codename Revolution™ special parameters

You can insert special parameters (a string of characters) into each taught sentence. These parameters are replaced automatically in the programs output text (response) with the symbolic representation of the current parameter. This feature makes the program more interesting and more realistic.

[ACTION]
Include this parameter in the sentence (at the beginning of the sentence), to make program respond this sentence with action (used only for IRC).

[USERNICK]
This parameter will be replaced with the speaking user's nickname.

[SOMENICK]
This parameter will be replaced with a random user's nickname from the IRC chat channel.

[DATE]
This parameter will be replaced with the current date.

[TIME]
This parameter will be replaced with the current time.

[OWNERSNICK]
This parameter will be replaced with the program owner's nickname.

[VAICNICK]
This parameter will be replaced with the programs nickname.

[ANSWEROWNER]
This parameter will be replaced with the nickname of the programs last response teacher.

[ACK]
This parameter will be replaced with analyzed context keyword. Analyzed context keyword is a context keyword which is linked with a contextual response of the programs database.

[HISTMINDIF]
This parameter will be replaced with the number of minutes passed between the current entered sentence and the same sentence in the dialogue history.

[HISTPHRASE]
This parameter will be replaced with the history phrase. The history phrase is a previously entered phrase by the user.

[NUMINTERFACEQ]
This parameter will be replaced with the number of interface questions in database.

[NUMINTERFACES]
This parameter will be replaced with the number of interface standard sentences in database.

[NUMSYNONYMCMD]
This parameter will be replaced with the number of defined command synonyms in database.

[NUMSYNONYMWRD]
This parameter will be replaced with the number of defined word synonyms in database.

[NUMSYNONYMCOR]
This parameter will be replaced with the number of defined word corrections in database.

[NUMCONTEXTWRD]
This parameter will be replaced with the number of defined context keywords in database.

[NUMPOSITIVESW]
This parameter will be replaced with the number of defined positive words in database.

[NUMNEGATIVESW]
This parameter will be replaced with the number of defined negative words in database.

[NUMCASESENSWD]
This parameter will be replaced with the number of defined case sensitive words in database.

[NUMSYSMESAGES]
This parameter will be replaced with the number of defined event system messages in database.

[NUMKNOWNUSERS]
This parameter will be replaced with the number of known users for the program.

[NUMDTUNSOLVED]
This parameter will be replaced with the number of unsolved sentences for the program.

Integrated help system

The program has integrated (built-in core) help system that can be accessed by typing “!help” command in public IRC channel or programs private section. The integrated help system can be used by any user on IRC server. The integrated help system contains basic information about program, basic technical information, user commands, program owner and co-owner commands, training commands, developers contact information, etc.