University of Salzburg

Institute of Computer Science

Visualization of NetJEN problem domains and results

YARB

Yet Another Result Browser

Thomas Aschauer

Erich Eichberger

Thorsten Engl

Christian Zödl

Date:	May 2, 2000
Version:	1.0

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter shows an overview of the problem and the solution, that this project should be.

1.1 The problem

Imagine you have got an neural network simulator and you want to do some stock rate prediction. So you have an chart, representing the stock rate, and you want to have a graphic, which displays the prediction of the rate. You have to write a tool that is able to show the chart. But what, if you want do do some other things with the simulator for NNs[1], like character recognition, and display the results? For these other tasks you have to develop similar tools which do the visualization for each problem.

1.2 The project

The project is going to be a graphical tool, written in java, for the visualization of netJEN problem domains. The aim is to show the pattern file[2] before and after netJEN evaluated it (they are called learning and result pattern). The pattern files are in the pattern file format of the SNNS [3]. Thus the program has to be able to read this files and display the information contained in a manner that depends on the problem domain. This, perhaps, could be an image from a satellite that should be classified or prognosis of stock rates.

1.3 Titling

Since we started working on this project we have been looking for a name. After hours of brain storming we finally found one:
YARB [4]

[1]Short form of **N**eural **N**etwork, sometimes used as synonym for ANN (**A**rtificial **N**eural **N**etwork)
[2]The data neural network simulators work with is called pattern
[3]**S**tuttgart **N**eural **N**etwork **S**imulator
[4]**Y**et **A**nother **R**esult **B**rowser

1.4 Where are NNs used?

Neural Networks are very often used in industry nowadays, and there is a wide spread of applications where the use of NNs could bring big advantages. So it is useful to examine where NNs are already used today. Additionally, we have tried to determine how the problem could be visualized and in which problem domain it could fit.

1.4.1 Applications for NNs

Manufacturing

Literature: [1], [2], [3]

- process optimizing
  Visualization: Graph
  Problem Domain: Optimization, Time series prediction, Parameter determination

- process monitoring and controlling
  Visualization: 3D - Chart
  Problem Domain: Signal analysis and prediction

- determining operational policies
  Visualization: Graph
  Problem Domain: Parameter determination

- predictive control of quality
  Visualization: 2D - Chart
  Problem Domain: Signal analysis and prediction, Time series prediction

- flexible manufacturing scheduling
  Visualization: Graph
  Problem Domain: Rule extraction, Time series prediction

Chemical Industry

Literature: [1], [4]

- sensor analysis
  Visualization: Line - Chart
  Problem Domain: Signal recognition

- spectroscopy
  Visualization: Bar - Chart
  Problem Domain: Signal recognition, Classification

- molecular modeling
  Visualization: 3d - Model
  Problem Domain: Modeling

- data rectification for dynamic processes
  Visualization: 2D / 3D - Chart
  Problem Domain: Signal recognition, Time series prediction
- process modeling for fault detection
  Visualization: Graph / 3D - Chart
  Problem Domain: Rule extraction, Optimization
- statistical analysis
  Visualization: 2D / 3D - Chart
  Problem Domain: Time series prediction

Medicine

Literature: [1], [5]

- diagnosis of coronary artery disease
  Visualization: Graph / Chart
  Problem Domain: Classification
- structure analysis of virus proteins
  Visualization: 3D - Model
  Problem Domain: Parameter determination
- diagnosis of epilepsy
  Visualization: Graph / Chart
  Problem Domain: Classification
- rehabilitation of persons with incomplete spinal cord injury
  Visualization: Graph
  Problem Domain: Optimization, Rule extraction
- analysis and classification of EEG signals
  Visualization: Bar - Chart
  Problem Domain: Classification, Signal analysis and prediction

Nuclear engineering

Literature: [1], [6]

- spectral analysis
  Visualization: Bar - Chart
  Problem Domain: Signal recognition, Parameter determination
- signal prediction in nuclear power plants
  Visualization: Line - Chart
  Problem Domain: Signal analysis and prediction, Time series prediction
- classification of energy levels
  Visualization: Bar - Chart
  Problem Domain: Classification

- alarm handling in nuclear power plants
  Visualization: Line - Chart
  Problem Domain: Signal analysis and prediction, Time series prediction

Miscellaneous applications

Literature: [1], [7], [8], [3], [4], [5], [9], [6], [10], [2]

- classification of satellite pictures
  Visualization: Areas / 3D - Model
  Problem Domain: Classification

- sensor analysis
  Visualization: 2D / 3D - Chart
  Problem Domain: Signal recognition

- optical character recognition
  Visualization: Picture / Text
  Problem Domain: Classification, Signal recognition

- voice recognition
  Visualization: Line / Bar - Chart / Text
  Problem Domain: Signal recognition, Classification

- stock rate prognosis
  Visualization: Line - Chart
  Problem Domain: Time series prediction

- texture recognition
  Visualization: Bar - Chart / Text
  Problem Domain: Classification

- temperature regulation
  Visualization: Line - Chart
  Problem Domain: Signal analysis and prediction, Time series prediction, Rule extraction, Parameter determination

- sunspot evaluation
  Visualization: 2D / 3D - Chart
  Problem Domain: Time series prediction, Classification, Rule extraction

- business prognosis and optimization
  Visualization: Graph / 3D - Chart
  Problem Domain: Optimization, Time series prediction

- sale prognosis
  Visualization: 2D / 3D - Chart / Graph
  Problem Domain: Time series prediction

1.4.2 Problem domains

The next task was to itemize all of the problem domains we found.

- classification
- time series prediction
- parameter determination
- signal recognition
- signal analysis and prediction
- rule extraction
- optimization
- modeling

1.5 How to visualize these problem domains?

After we figured out the problem domains listed above, we have to determine how the visualization could look like. On then next pages you can see some examples.

1.5.1 Summary of possible visualizations

- picture data as bitmap
  (see Figure 1.1)
- classification as areas
  (see Figure 1.2)
- Various charts: Bar, Line, Circle, ...
  (see Figures 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9 and 1.10)
- 3D-graphics
  (see Figures 1.11, 1.12, 1.13 and 1.14)
- Frequency diagram
  (see Figure 1.15)
- graphs
  (see Figure 1.16)
- text
- plain pattern text

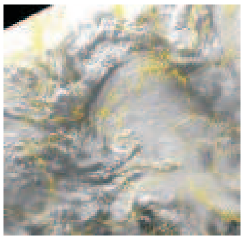

Figure 1.1: Example: Satellite bitmap

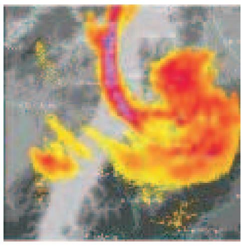

Figure 1.2: Example: Satellite bitmap with result areas

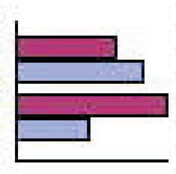

Figure 1.3: Example: Bar chart

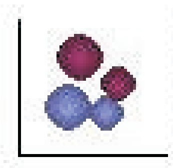

Figure 1.4: Example: Bubbles chart

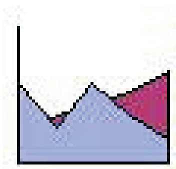

Figure 1.5: Example: Area chart

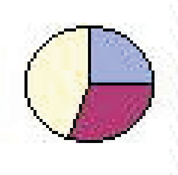

Figure 1.6: Example: Circle chart

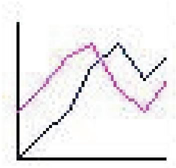

Figure 1.7: Example: Line chart

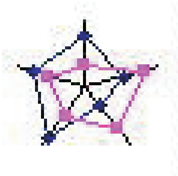

Figure 1.8: Example: Net chart

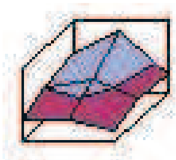

Figure 1.9: Example: Surface chart

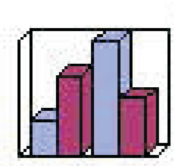

Figure 1.10: Example: Pillar chart

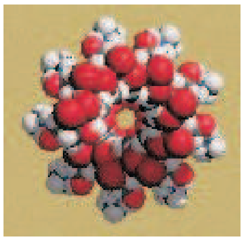

Figure 1.11: Example: Molecule

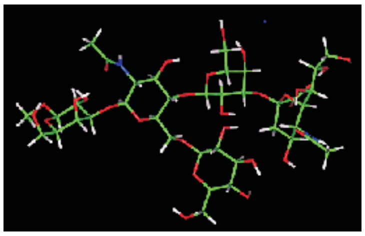

Figure 1.12: Example: Molecule Structure

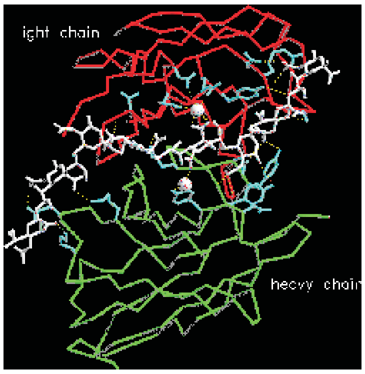

Figure 1.13: Example: Molecule Structure 2

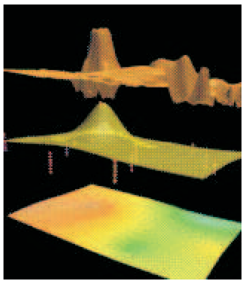

Figure 1.14: Example: Landscape

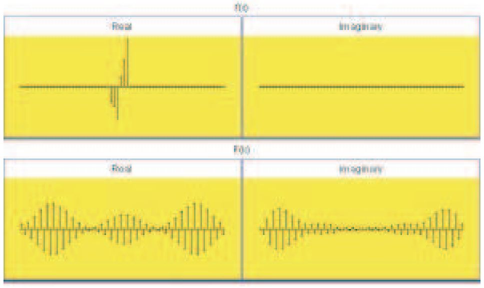

Figure 1.15: Example: Frequency diagram

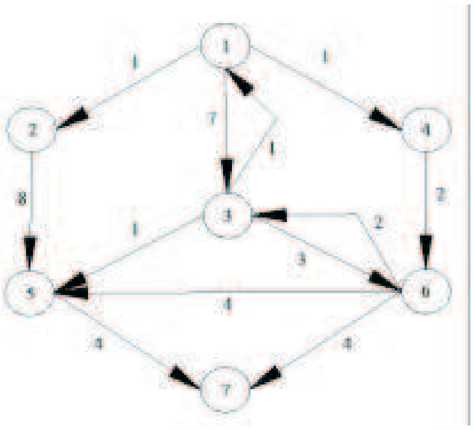

Figure 1.16: Example: Graph

1.5.2 Examples

We have collected some examples for visualizations. These examples are shown in figures 1.1,1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12, 1.13,1.14, 1.15 and 1.16.

1.6 How YARB uses NetJENs Files

1.6.1 General file concepts in NetJEN

If you want to use NetJEN you have to format your problem into pattern files. These pattern files are used by NetJEN, the generated output is saved to the so called result files, which are having the same file format.
During the learning evolution of the ANN the pattern files are used to describe the input data and the corresponding output. After this learning phase has completed, the pattern files are used to feed the ANN with input data. The output of NetJEN is written into result files.

All files used are in SNNS-file formats (see [11]).

1.6.2 General file concepts in YARB

Because YARB is a browser for pattern and result files, the same files are used.
In some problem domains the raw input data has many redundant information. In these cases only the essential part of the data is converted to a pattern file. If the users wants the whole information from the input to be displayed, a second file is needed, GIF, JPEG or another pattern file.

1.7 Custom Configuration of YARB

While analyzing the problem we figured out that for some problem domains its very important to customize the appearance of the visualization (Scale, Zoom, ...). To prevent the user to set the display properties again and again we introduced the concept of configuration files. These files contain all the information needed to display pattern- and result files in a postulated way.

Chapter 2

Project specification

This chapter describes the specification this project should match.

2.1 Main Appointment

The user should be able to get different visualizations of various problem domains.

2.1.1 Mandatory criteria

- Visualize pattern in SNNS pattern files
- Visualize pattern in SNNS result files
- Choose between visualization options
- Customize the visualization options

2.1.2 Additional criteria

- The ability to be called from other programs (RMI [1])
- Easy to implement new types of visualization

2.1.3 Delimitation

- No modifying features of SNNS files

2.2 Product use

2.2.1 Areas of application

- All possible problem domains

[1] **R**emote **M**ethod **I**nvocation

2.2.2 Target group

- NetJEN users

2.3 Operational environment

2.3.1 Software

- All platforms supporting Java

2.3.2 Orgware

- Eventual systems that support RMI

2.3.3 Product interfaces

- Ability to read SNNS files
- Ability to read JPEG and GIF files
- Eventual RMI

2.4 Product functions

- Open pattern and result files
- Visualize data by manually choose type
- Visualize data by using a wizard
- Visualize picture data (Gif, Jpeg)
- Change visualization configuration
  - Scaling
  - Axis titles
  - ...
- Save visualization configuration
- Load visualization configuration
- Change visualization type

2.5 Product data

- Configuration files

2.6 Product performance

Not specified

2.7 User interface

- Standard Java GUI with Swing components
- Tree view of possible problem domains and their visualization
- Every file to be displayed in a single MDI [2] window

2.8 Development environment

- ArgoUML for UML [3] designs
- Forte for Java as IDE [4]

[2] **M**ultiple **D**ocument **I**nterface
[3] **U**nified **M**odeling **L**anguage
[4] **I**ntegrated **D**evelopment **E**nvironment

Chapter 3

Software design

Literature:

- General software analysis and design: [16, pages 741-822]
- UML: [15]
- Software pattern (Design pattern): [12], [13], [14]

3.1 Use Cases

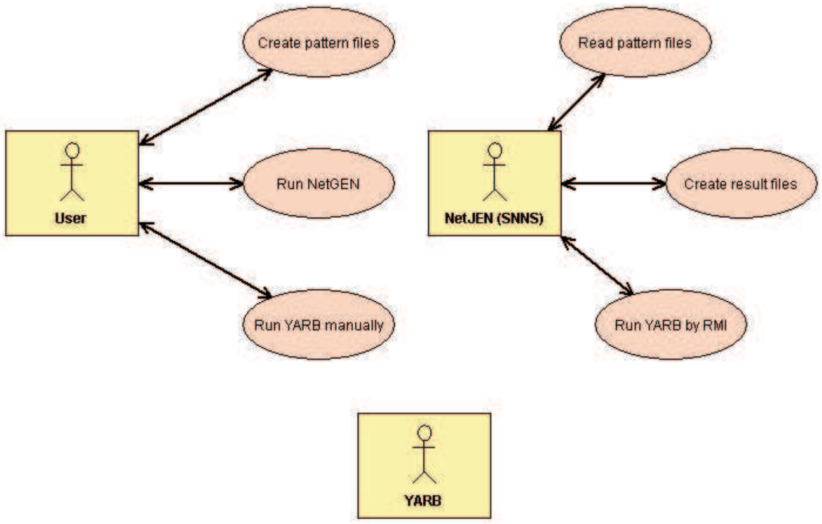

Figure 3.1: UCD: Interaction of User, NetJEN and YARB

The Use Case diagram in figure 3.1 shows the general interaction between the user, netJEN and YARB. The Use Case for running YARB manually is shown in figure 3.2. A Use Case refinement for the *Configure YARB manually* is shown in figure 3.3. The Use Case in figure 3.4 shows YARB run by RMI.

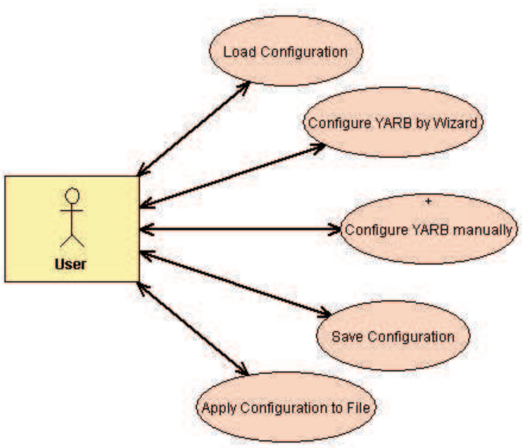

Figure 3.2: UCD: Run YARB manually

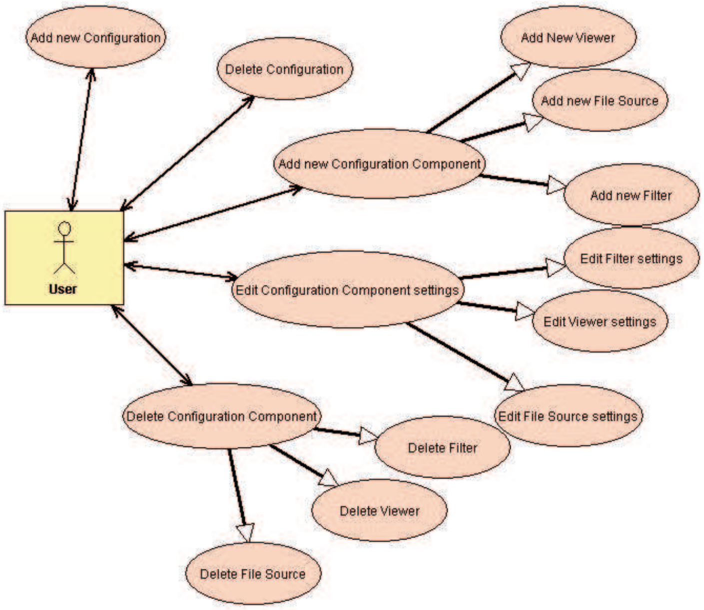

Figure 3.3: UCD: Configure YARB manually

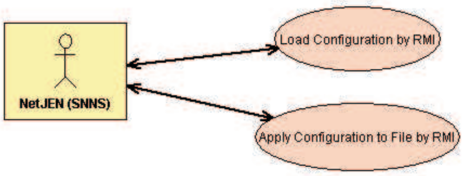

Figure 3.4: UCD: Run YARB by RMI

3.2 Class Diagrams

The first task to do was to model the interfaces for the data model of YARB. The results can be seen in figure 3.5.

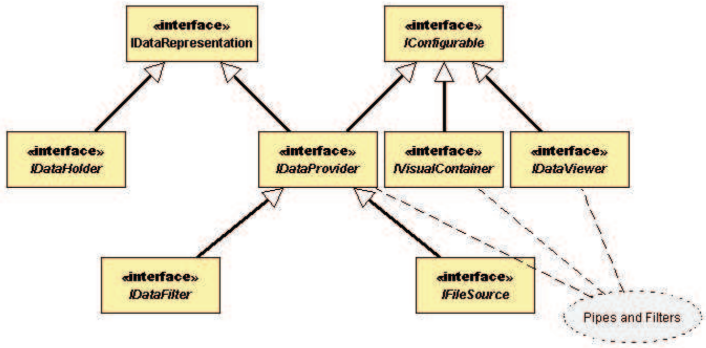

Figure 3.5: CD: YARB Interfaces

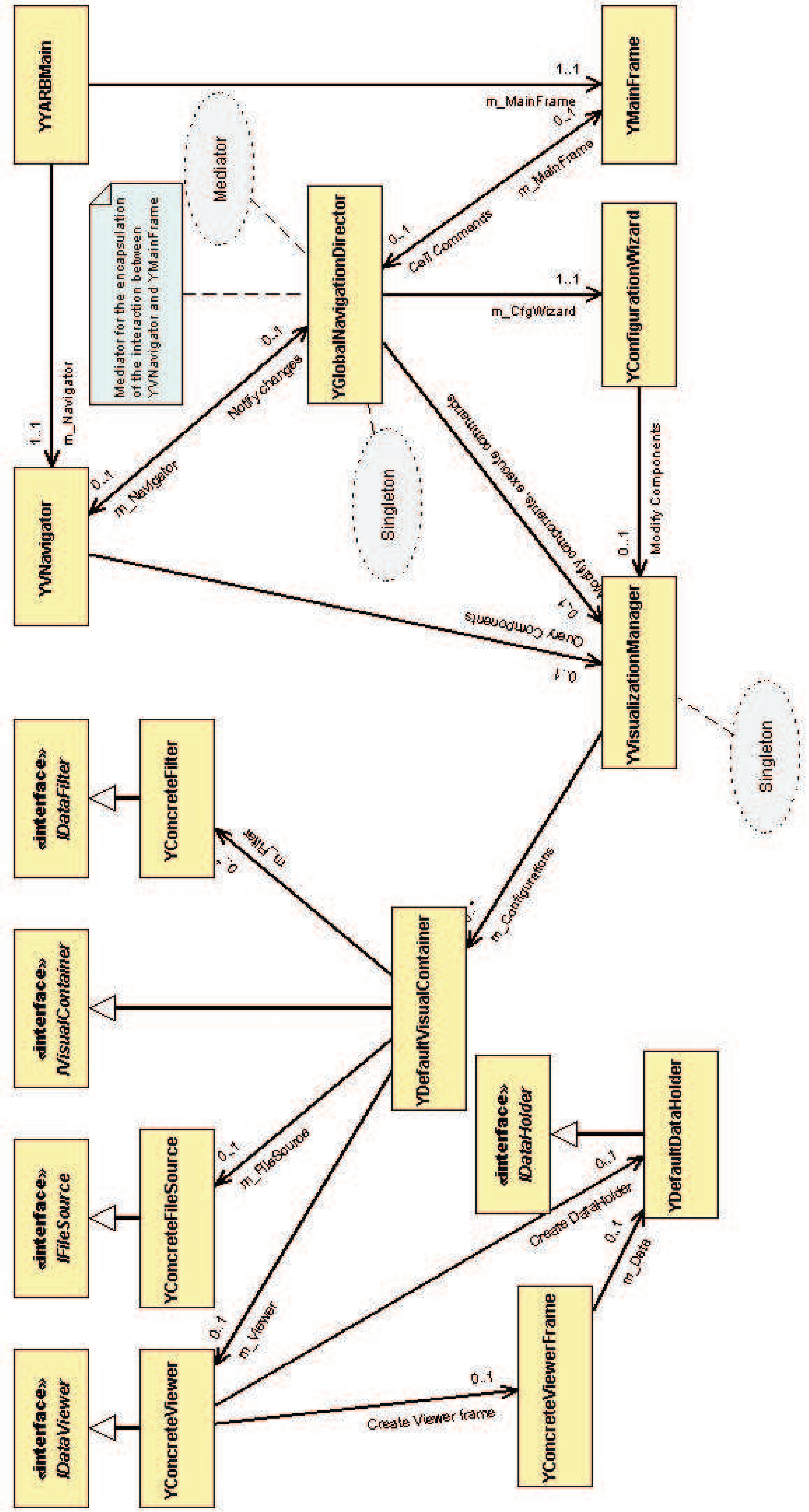

Figure 3.6: CD: Main class diagramm

Appendix A

Reports

This chapter contains our reports and results of the week.

A.1 1st. Protocol March 15, 2000

General

- *Present persons:*
  - DI Mag. Roland Schwaiger
  - Erich Eichberger
  - Thomas Aschauer
  - Thorsten Engl
  - Christian Zödl
- *Place:* DI Mag. Schwaigers Office
- *Time:* 16.00 - 16.30
- *Chief of protocol:* Erich Eichberger

General project definition

Theme: Visualization of netJEN problem domains and results

The project is going to be a graphical tool, written in java, for the visualization of netJEN problem domains. The aim is to show the pattern file[1] before and after netJEN evaluated it (they are called learning and result pattern). The pattern files are in the pattern file format of the SNNS [2]. Thus the program has to be able to read this files and display the information contained in a manner that depends on the problem domain. This, perhaps, could be an image from a satellite that should be classified or prognosis of tax courses.

Topics talked about

- Administrative details
- Sources for Information related to the topic
- Presentation of our group's web site
- The Neural Network Simulator netJEN and SNNS
- Problem domains, kinds of visualization

[1] The data neural network simulators work with is called pattern
[2] Stuttgart Neural Network Simulator

Task list

- Pictures of group-members (for web site)
- Find problem domains for NNs [3]
- Find NN-applications and problems that are solved with NNs

[3] NN... short form of Neural Network

A.2 Results of the week (March 15th to March 22nd)

The goal of this week was to find the applications and problem domains of NNs. First we discuss the NN-Applications we found, and then we list the problem domains we found within these applications.

Applications for NNs

Manufacturing

- process optimizing
- process monitoring and controlling
- determining operational policies
- predictive control of quality
- flexible manufacturing scheduling

Chemical Industry

- sensor analysis
- spectroscopy
- molecular modeling
- data rectification for dynamic processes
- process modeling for fault detection
- statistical analysis

Medicine

- diagnosis of coronary artery disease
- structure analysis of virus proteins
- diagnosis of epilepsy
- rehabilitation of persons with incomplete spinal cord injury
- analysis and classification of EEG signals

Nuclear engineering

- spectral analysis
- signal prediction in nuclear power plants
- classification of energy levels
- alarm handling in nuclear power plants

Miscellaneous applications

- classification of satellite pictures
- sensor analysis
- optical character recognition
- voice recognition
- stock rate prognosis
- texture recognition
- temperature regulation
- sunspot evaluation
- business prognosis and optimization
- sale prognosis

Problem domains

- classification
- time series prediction
- parameter determination
- signal recognition
- signal analysis and prediction
- rule extraction
- optimization
- modeling

Ideas for possible visualization

- picture data as bitmap
- classification as regions (graphs)
- bar chart
- line chart
- plain pattern text
- graphs
- text

A.3 2nd. Protocol March 22nd, 2000

General

- *Present persons:*
  - DI Mag. Roland Schwaiger
  - Erich Eichberger
  - Thomas Aschauer
  - Thorsten Engl
  - Christian Zödl
- *Place:* DI Mag. Schwaigers Office
- *Time:* 16.00 - 16.30
- *Chief of protocol:* Erich Eichberger

Topics talked about

- Results form last week
- Possible visualization of problem domains
- Possible names of problem domains (like classifikation, ...)

Task list

- Adding bibliography information
- Finding problem domains, their possible visualization and names
- Pictures of possible visualization
- Grouping NN-Applications to problem domains
- Work into ArgoUML

A.4 Results of the week (March 22nd to March 28th)

Adding bibliography information

- Added BibTeX file
- Added all our Internet links

Finding problem domains, their possible visualization and names

- Analyzed the application for ANNs and classified the problem domains
- Added the corresponding problem domains to all applications
- Collected sample pictures of possible visualization, added cross references to all applications
- Got names for problem domains

A.5 3rd. Protocol March 29th, 2000

General

- *Present persons:*
  - DI Mag. Roland Schwaiger
  - Erich Eichberger
  - Thomas Aschauer
  - Thorsten Engl
  - Christian Zödl
- *Place:* DI Mag. Schwaigers Office
- *Time:* 16.00 - 16.30
- *Chief of protocol:* Erich Eichberger

Topics talked about

- Results form last week
- Structure of the project paper
- File concepts of NetJEN
- File formats

Task list

- Restructuring of the project paper
- Research the format of the used files
- Add introduction to project paper
- Take a look at UML
- Think about possible realization

A.6 Results of the week (March 29th to April 12th)

Restructuring the project paper

- Changed the layout to book
- Added introduction chapter

Research the format of the used files

- Took a look at the SNNS book
- Took a look at the samples provided by Mr. Schwaiger

Think about possible realization

- Reconfigured the analyzing phase
- Thought about possible use of design patterns
- Introduced the concept of the configuration files
- Basic approach to use cases

A.7 4rd. Protocol May 2nd, 2000

General

- *Present persons:*
  - DI Mag. Roland Schwaiger
  - Erich Eichberger
  - Thomas Aschauer
  - Thorsten Engl
  - Christian Zödl
- *Place:* DI Mag. Schwaigers Office
- *Time:* 16.30 - 17.00
- *Chief of protocol:* Erich Eichberger

Topics talked about

- Results form last week
- Use cases
- Class diagrams
- Handling of the different problem domains
- Configuration file with XML format

Task list

- Design use case diagrams
- Design class diagrams
- Develop a prototype GUI
- Search for existing XML parsers
- Create a prototype configuration file

A.8 Results of the weeks (April 12th to May 2nd)

Design use case diagrams

- Interaction between User, NetJEN and YARB
- Run YARB manually
- Configure YARB manually
- Run YARB by RMI

Design class diagrams

- YARB Interfaces
- Main class diagram

Develop a prototype GUI

The Test-GUI for playing with configurations, file sources, data filters and data viewers is almost finished.

XML parsers

We decided to use the new XML format to represent our configuration files. After a while I decided to use Sun's XML parser, also called JAXP. I had to figure out how XML is defined, afterwards I wrote some exmaple files. I used the Document Object Model to parse my files, because you get a tree like result,and displayed it in a Java Swing Tree.

Task management

After finishing the design phase we had to split the project implementation into different tasks and to fix the task distribution.

- **Aschauer Thomas:**

  Software architecture These are the Use Cases, Class Diagrams, persistence model

  Interface Design Includes all interface classes

  Main application logic Implementation of the main frame, the YARB Navigator, ...

- **Eichberger Erich:**

  Data Viewer Implementation and design of the different viewer classes. See 1.5.1 on page 9 for details.

- **Engl Thorsten:**

  XML - parse and persistence engine Implementation of an existing XML parser, like Suns JAXP, to read and store different configurations.

  File Sources Parser for existing SNNS files.

- **Zödl Christian:**

  Data Filter Identify all needed filters, design and implement them. A filter that executes the external LASSPTools filters is planned too.

Bibliography

[1] The centre for neural computing applications. http://www.brunel.ac.uk/research/cnca/links/related.html.

[2] Einfuehrung in den stuttgarter neuronale netze simulator, 1996. http://www-i6.Informatik.RWTH-Aachen.de/Colleagues/HeKonn96/abstract/snns.html.

[3] Neural network applications in manufacturing. http://www.emsl.pnl.gov:2080/proj /neuron/neural/bib/manufacturing.html.

[4] Neural network applications in chemistry. http://www.emsl.pnl.gov:2080/proj /neuron/neural/bib/chemistry.html.

[5] Neural network applications in medicine. http://www.emsl.pnl.gov:2080/proj /neuron/neural/bib/medicine.html.

[6] Neural network applications in power plants. http://www.emsl.pnl.gov:2080/proj /neuron/neural/bib/power.html.

[7] Nn-at gruppenprofile (8-apr-1997). http://www.ai.univie.ac.at/nn-at-97/gruppen.html.

[8] Problem domain. http://www.cis.hut.fi/ jhollmen/dippa/node36.html.

[9] Neural network applications in molecular modeling. http://www.emsl.pnl.gov:2080/proj /neuron/neural/bib/molecular.html.

[10] Neural network applications in process control. http://www.emsl.pnl.gov:2080/proj /neuron/neural/bib/molecular.html.

[11] Andreas Zell; Günter Maimer; Michael Vogt; Ralh Hübner; Sven Döring; Kai-Uwe Herrmann; Tobias Soyez; Michael Schmalzl; Tilman Sommer; Artemis Hatzigeorgiou; Dietmar Posselt; Tobias Schreiner; Bernward Kett; Gianfranco Clemente; Jens Wieland; Niels Mache. *SNNS Stuttgart Neural Network Simulator User Manual.* 1.4th edition, 1995. Report No. 6/95.

[12] Erich Gamma; Richard Helm; Ralph Johnson; John Vlissides. *Design Patterns - Elements of reusable object oriented software.* Addison-Wesley, 1st edition, 1995. ISBN 0-201-63361-2.

[13] Erich Gamma; Richard Helm; Ralph Johnson; John Vlissides. *Entwurfsmuster - Elemente wiederverwendbarer objektorientierter Software.* Addison-Wesley, 1st edition, 1996. ISBN 3-89319-950-0.

[14] Frank Buschmann; Regine Meunier; Hans Rohnert; Peter Sommerlad; Michael Stal. *Pattern-orientierte Software-Architektur. Ein Pattern-System.* Addison-Wesley, 1st edition, 1998. ISBN 3-8273-1282-5.

[15] Rainer Burkhardt. *Unified Modeling Language. Objektorientierte Modellierung für die Praxis.* Addison-Wesley, 2nd edition, 1999. ISBN 3-8273-1407-0.

[16] Bjarne Stroustrup. *Die C++ Programmiersprache.* Addison-Wesley, 3rd edition, 1998. ISBN 3-8273-1296-5.