

Woord vooraf

Deze masterproef is een laatste stap in de opleiding van Academische Master Industriële Wetenschappen, Elektronica-ICT. Hiermee zetten we onze opgedane kennis om in de praktijk. Graag willen we dan ook de personen bedanken die mee hebben geholpen deze masterproef mogelijk te maken.

Eerst willen we onze promotor en opdrachtgever prof. Dr. Ir. Wim Dehaene bedanken . Hij heeft ons de mogelijkheid gegeven deze masterproef uit te voeren. Ook willen we onze schoolpromotor ing. Leo Rutten bedanken voor zijn advies en begeleiding. Tevens gaat een woord van dank uit naar de heer Jos Creyns voor het nakijken van de tekst.

Verder bedanken we ook onze ouders. Zonder hen hadden we de mogelijkheid niet gehad om deze opleiding te voltooien. Ten slotte willen we onze familie en vrienden, in het bijzonder de heer Renaat Sijbers, bedanken. Zij gaven ons morele en praktische steun tijdens dit eindwerk in het bijzonder en onze gehele opleiding in het algemeen.

Inhoudsopgave

Woord vooraf	1
Inhoudsopgave	2
Lijst van illustraties	4
Verklarende woordenlijst	5
Abstract	6
1 Inleiding	8
1.1 Situering	8
1.2 Globale probleemstelling	9
1.3 Doelstellingen.....	10
2 Ontwerptechnieken.....	12
2.1 Inleiding	12
2.2 Programmeertaal	12
2.2.1 High-level programmeertalen	13
2.2.1.1 Visual Basic .NET	14
2.2.1.2 C++.....	14
2.2.1.3 C#	14
2.2.1.4 Java.....	15
2.2.1.5 Conclusie.....	15
2.3 Programmavoorstelling	16
2.3.1 UML	16
2.3.2 API	17
2.3.3 Javadoc	18
2.4 Ontwikkelomgeving	19
2.5 Subversion.....	20
2.6 Editorkit.....	21
2.6.1 JHotDraw	21
2.6.2 Jung	21
2.6.3 Grappa	22
2.6.4 Piccolo.....	22
2.6.5 Conclusie.....	22
2.7 XML	23
2.7.1 Parsers	23
2.7.2 Xstream	24
2.8 Logger bibliotheek	25
2.8.1 Java Logging	25
2.8.2 Log4j	25
2.8.3 Conclusie.....	25

3 Systeemontwerp	26
3.1 Inleiding	26
3.2 Hoofdvenster	27
3.3 Modules inlezen	28
3.3.1 Programma-architectuur	28
3.3.2 Inlezen	28
3.3.3 Configuratiemodules	30
3.3.4 Modules	31
3.4 Workspace	33
3.4.1 Programma-architectuur	33
3.4.2 Teken van een schakeling	35
3.5 De simulatie	37
3.5.1 Programma-architectuur	37
3.5.2 Bediening	38
3.5.3 Grafische weergave	39
3.5.4 Events	40
3.5.5 Logboek	40
3.6 Extra functionaliteiten	41
3.6.1 Opslaan/opslaan als	41
3.6.2 Laden	42
4 Modules toevoegen	43
4.1 Het XML-bestand	43
4.2 De module	44
4.3 De code	45
4.4 Conclusie	45
5 Besluit	46
Literatuurlijst	47
Bijlagen	49

Lijst van illustraties

Figuur 1: Voorbeeld van een UML	17
Figuur 2: Het hoofdvenster	27
Figuur 3: UML voor het inlezen vanuit XML.....	28
Figuur 4: Lijst van modules	29
Figuur 5: UML voor de configuratiemodules	30
Figuur 6: UML voor de modules.....	31
Figuur 7: UML voor het tekenen van een schakeling	33
Figuur 8: Voorbeeld van een linkset	34
Figuur 9: Voorbeeld van een workspace.....	35
Figuur 10: UML voor de simulatie	37
Figuur 11: Weergave bij de simulatie	39
Figuur 12: Opgeslagen voorbeeld van een workspace.....	41
Figuur 13: Opgeslagen tekstbestand	42

Verklarende woordenlijst

CPU.....	Central Processing Unit
API.....	Application Programming Interface
GUI.....	Graphical User Interface
JVM.....	Java Virtual Machine
JRE.....	Java Runtime Environment
IDE.....	Integrated Development Environment
UML.....	Unified Modelling Language
XML.....	Extensible Markup Language
HTML.....	HyperText Markup Language
DOM.....	Document Object Model
SAX.....	Simple API for XML
XPP3.....	XML Pull Parser 3rd edition

Abstract

Het departement Elektrotechniek (ESAT) van de KULeuven voert samen met de RVO-society het project “IR13: ingenieurs van 13 jaar” uit. Het doel van dit project is om jongeren kennis te laten maken met typische ingenieursvaardigheden. Zo maken de leerlingen een schakeling met logische modules, waarmee dan een omgebouwde legotrein wordt aangestuurd. Het doel van deze masterproef is om een programma te ontwerpen waarin de leerlingen de schakeling kunnen maken en simuleren.

De simulator is ontworpen in de programmeertaal Java en met behulp van XML. Tevens maakt het programma gebruik van 2 externe bibliotheken. Een eerste bibliotheek is Piccolo, die zorgt voor de grafische voorstelling van de schakelingen. Een tweede bibliotheek is Xstream, die de gegevens uit XML-bestanden inleest. Het systeem werd stap voor stap opgebouwd, waarbij alle tussenresultaten een werkend geheel vormen.

Het eindproduct is een simulator van het IR13-systeem met een gebruiksvriendelijke grafische *interface*. De grafische interactie bij de simulatie is zo dicht mogelijk bij de realiteit gehouden zodat ze duidelijk is voor de gebruiker. Bovendien worden enkele veelgemaakte fouten automatisch gedetecteerd en gesignaleerd. Verder kunnen aanpassingen aan de modules eenvoudig doorgevoerd worden in de XML-bestanden.