Automatic Circuit Equations Formulation
Software Documentation

Project VAL-AMS. ANR Project ANR-06-SETIN-018-0

Deliverable 6

Vincent Acary, Olivier Bonnefon

Date : January 27, 2009

Table of Contents

Table of Contents

1 Nonsmooth Automatic Circuit Equation Formulation Software NS-ACEF - Project Overview

1.1 Release Information

Project:	NS-ACEF
Internal Release Number:	1.0
Last update:	January 27, 2009

1.2 Mission and Scope

1.2.1 What is the purpose and scope of this document ?

This document describes the functionalities of the software and the major constraints of development and exploitation. It plans the software design and development. It concerns users and software framework builders.

1.2.2 What is the scope of this project?

This project is a part of the following ANR project:

- Acronym: VAL-AMS High-confidence validation of analog and mixed-signal circuits
- Theme : Sûreté des systèmes informatisés (Thème 2)

1.2.3 Goal of the project

The goal of the project is to answer to the work package WP3 described in the VAL-AMS documents:

- WP3: Advanced numerical analysis techniques.
    1. Developing numerical analysis techniques based on the non-smooth approach.
    2. Development of an automatic D.A.E. and L.C.P. formulation tool.

1.2.4 Project proposal, main software functionalities

The first step consists in developing a tool able to get a circuit equation formulation from an electrical circuit description, that is a Nonsmooth Automatic Circuit Equation Formulation software (NS-ACEF).

The second step, the software must be able to perform the transient analysis using the Siconos platform.

1.2.4.1 Software input

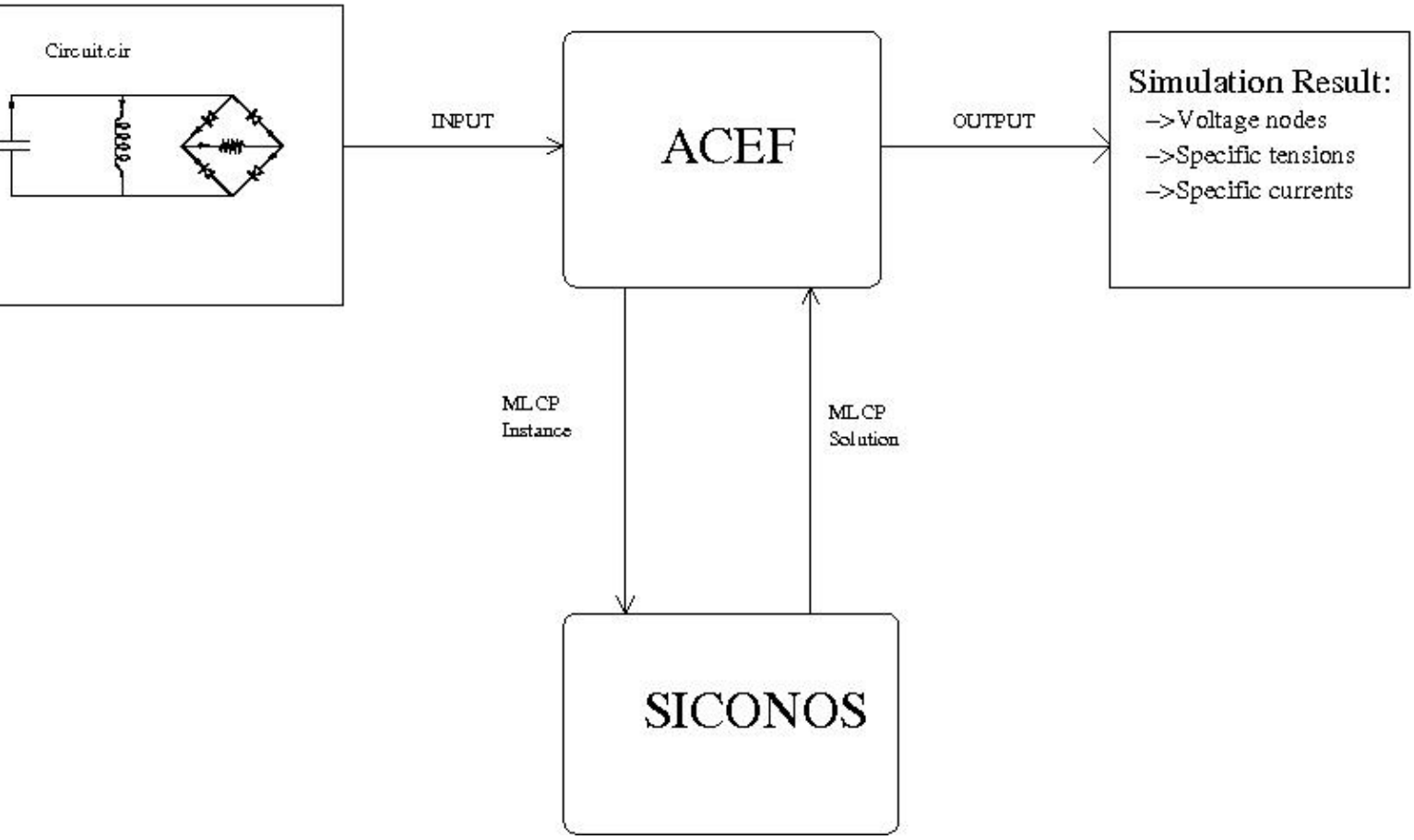
The input is an electrical circuit description in the SPICE format usually called a Netlist. A Netlist is a file with an extension *.cir* contains the circuit description. NS-ACEF uses the same Input file format.

1.2.4.2 Software output

The software output is the transient analysis and the circuit equations.

1.2.4.3 Relation with Siconos

NS-ACEF needs SICONOS to time integrate Complementarity systems (CS) and to solve Complementarity Problems (LCP, MLCP, NCP). The relation with the Siconos Software is sketched in the following figure



1.3 Status

The project is currently under development.

1.4 Feasibility study

Existing circuit analysis	MNA	Presents the Modified Nodal Analysis (MNA) which is used in SPICE to automatically formulate circuit equations.
LCP circuit analysis	MNA adaptation	Describe how to adapt the MNA for a CS formulation.
Examples	Diode bridge	Validation the previous analysis.

1.5 Software Documents

Software Requirement Document.	Functionalities Feature details Delivery and release	Specification and user requirements. List of Software functionalities. It aims to define precisely the software to realize. It describes the functionalities and characteristics of the software and the constraints of development and exploitation. It plans the project developpement.
Architectural and detailed Design	Architectual and detailed Design Details Design and spefication	Definition of the software global architecture. Overview of the detailed implementation

Document:	implementation	
Quality and Validation Plan	Tests Plan	Checks that the project is consistent
	Project organization	
Software delivery	Delivery description	Describes the first delivery

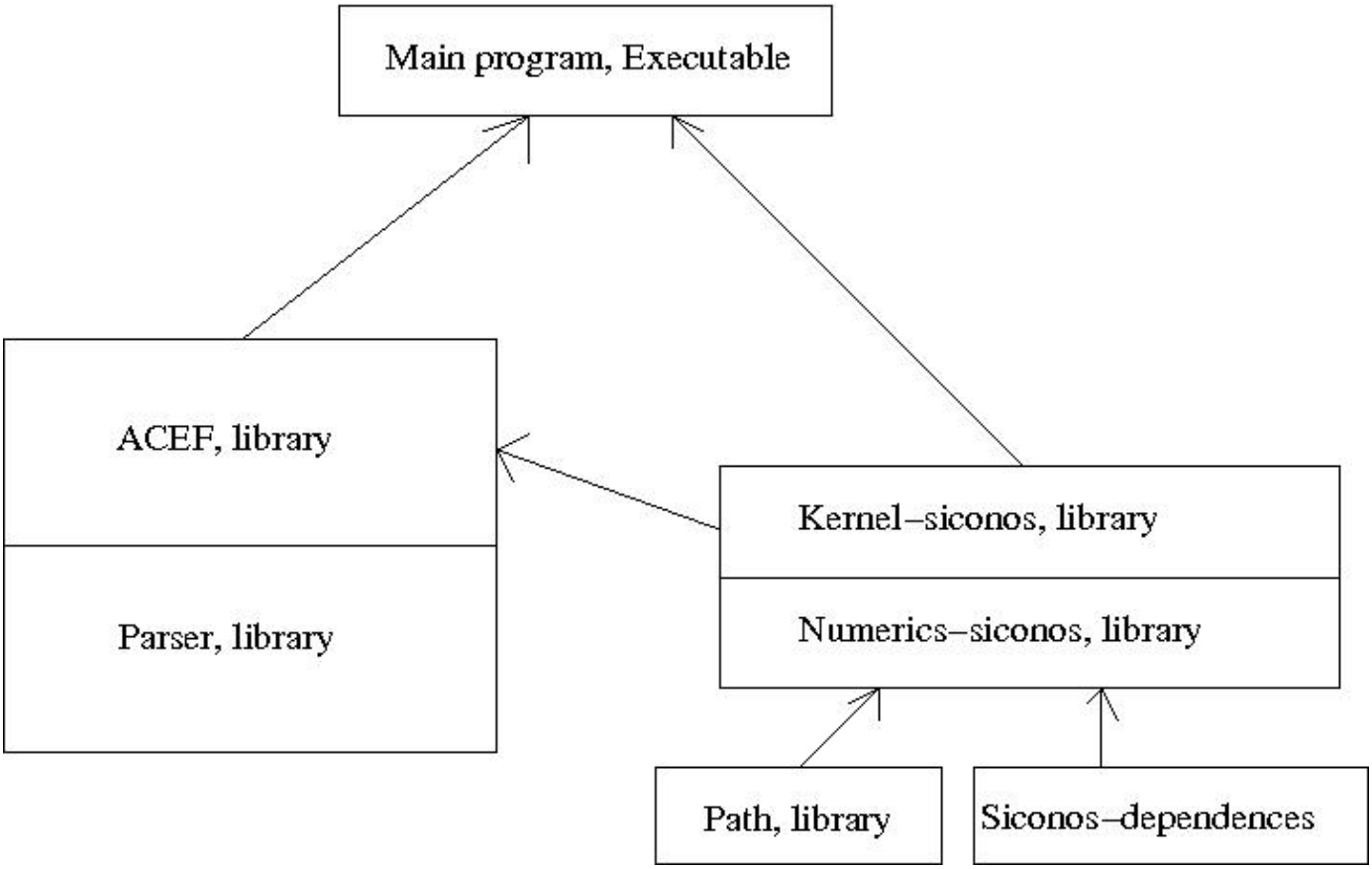
2 Architectural Design

2.1 Project information

Project:	NS-ACEF
Related Documents:	Functionalities Detailed Design and spefication implementation
Last update:	January 16, 2008

2.2 Introduction

The following chart presents the global software architecture



2.3 Parser library

The parser is an external library. It exports an API used by NS-ACEF.

If the input file format change, it will impact only the parser.

2.4 An executable

It implements two functionalities:

- Automatic Circuit Equation Formulation.
- Simulator or transient analysis.

2.5 SICONOS/NUMERICS

Numerics is a library used to solved the MLCP instances.

3 Detailed Design

See doxygen documentation of the code.

4 Functionalities

4.1 Project Information

Project:	NS-ACEF
Last update:	January 16, 2008
Related Documents:	Feature details

4.2 Features by Functional Area

4.2.1 Parser

It consists in storing and interpreting a Netlist.

- F-1.001 Store a Netlist.
- F-1.002 Sub-circuit management.
- F-1.003 Transient analysis.

4.2.2 Automatic Equation Circuit Formulation

- F-2.000 Linear, circuit RLC.

4.2.2.1 Piecewise linear formulation:

- F-2.001 Diode.
- F-2.002 Ideal transistor MOS.
- F-2.003 Comparator.
- F-2.004 Diode, Transistor, Comparator with parameters.
- F-2.005 Automatic picewise linear approximation and L.C.P. formulation.
- F-2.006 Multigrid algorithm: use a more simple piecewise linear approximation to accelerate the MLCP. resolution.
- F-2.007 Forecast active set: use a local piecewise linear approximation to solve a smaller MLCP.

4.2.2.2 Non linear formulation

- F-2.008 Non linear circuit: R,L or C non constant.
- F-2.009 Non linear complementarity formulation.

4.2.3 Simulation, analysis

- F-3.000 Time discretization.
- F-3.001 Transient analysis.
- F-3.002 Non constant source.
- F-3.003 DC analysis.

4.2.4 Mixed Simulator

The goal is to mix Newton methods and LCP. Newton methods are used for the smooth components and the LCP formulation for the non smooth components.

- F-4.000 Gradient method.
- F-4.001 Quasi-Newton methods.

- F-4.002 Write a mixed solver.

4.2.5 SICONOS integration

- F-5.000 MLCP solver.

4.2.6 Perspectives

- F-6.000 Specific electrical heuristics.

4.3 Project Information

Project:	NS-ACEF
Last update:	January 16, 2008
Related Documents:	Functionalities

4.4 Features

4.4.1 F-1.001: Store a Netlist

Priority:	Essential
Functional area(s):	Parser
Release:	1
Description:	The parser store a Netlist. There is an API to explore the circuit topology.
Notes and Questions:	

4.4.2 F-1.002: Sub-circuit management

Priority:	Essential
Functional area(s):	Parser
Release:	2
Description:	Sub circuit is a special key in a Netlist. It consists in managing it when the sub circuit is a comparator.
Notes and Questions:	A comparator is a non smooth component modeled with complementarity conditions.

4.4.3 F-1.003: Transient analysis.

Priority:	Secondary importance
Functional area(s):	Parser
Release:	2
Description:	A Netlist has a special key word to define the transient analysis parameters. It consists in reading and exporting these parameters.
Notes and Questions:	In the release 1, this information is written in an other file.

4.4.4 F-2.000: Linear, circuit RLC

Priority:	Essential
Functional area(s):	NS-ACEF
Release:	1
Description:	In this case the circuit is composed only with resistors, capacitors and inductors. It consists in developing an automatic circuit equation formulation. The result is a linear system.
Notes and Questions:	

4.4.5 F-2.001: Diodes

Priority:	Essential
Functional area(s):	NS-ACEF
Release:	1
Description:	A diode is a non smooth component, modeled with complementarity conditions. The diode model is an ideal diode. The automatic circuit equation formulation leads to a Mixed Linear Complementarity Problem(MLCP).
Notes and Questions:	There is no parameter in the model.

4.4.6 F-2.002: Ideal transistor MOS

Priority:	Essential
Functional area(s):	NS-ACEF
Release:	1
Description:	A transistor is a non smooth component, piecewise linear ,modeled with complementarity conditions. Automatic circuit equation formulation leads to a Mixed Linear Complementarity Problem(MLCP).
Notes and Questions:	There is no parameter in the model.

4.4.7 F-2.003: Comparator

Priority:	Essential
Functional area(s):	NS-ACEF
Release:	2
Description:	A Comparator is a non smooth component, piecewise linear ,modeled with complementarity conditions.
Notes and Questions:	There are no parameters in the model. It requires F-1002.

4.4.8
F-2.004: Diode, transistor, comparator with parameters

Priority:	Secondary importance
Functional area(s):	NS-ACEF
Release:	3
Description:	It consists to customize the component model with parameters(Saturation current, zero biais,...). The goal is to be nearer to the physical behaviour.
Notes and Questions:	Parameters list must be defined for each component.

4.4.9
F-2.005: Automatic piecewise linear approximation and L.C.P. formulation.

Priority:	Essential
Functional area(s):	NS-ACEF
Release:	Not plan
Description:	The goal is to develop a module to build a piecewise linear approximation for any non-smooth component, and write the L.C.P. formulation for this geometry.
Notes and Questions:	A feasibility study must be done.

4.4.10
F-2.006: Multigrid algorithm: Use a more simple piecewise linear approximation to accelerate the M.L.C.P. resolution.

Priority:	Secondary importance
Functional area(s):	NS-ACEF
Release:	3
Description:	The dimension of the L.C.P. formulation for a transistor is 10. These 10 parameters described 5 linear parts in the piecewise linear description. The goal is to use a very simple piecewise linear description, with only two parts, to get a smaller M.L.C.P.. We solve this smaller formulation, and the result is projected to solve the initial M.L.C.P.
Notes and Questions:	A feasibility study must be done.

4.4.11
F-2.007: Forecast active set : Use a local piecewise linear approximation to solve a smaller M.L.C.P..

Priority:	Essential
Functional area(s):	ACEF
Release:	2
Description:	It consists in looking only around the current physical values to get a smaller M.L.C.P..

Notes and Questions:	A feasibility study must be done.

4.4.12 F-2.008: Non linear circuit: R,L or C non constant.

Priority:	Essential
Functional area(s):	NS-ACEF
Release:	Not plan
Description:	It consists in updating the table equations when R, L or C are not constant.
Notes and Questions:	A time discretization for R, L, C must be done.

4.4.13 F-2.009: Non linear complementarity formulation.

Priority:	Secondary importance
Functional area(s):	NS-ACEF
Release:	Not plan
Description:	
Notes and Questions:	

4.4.14 F-3.000: Time discretization

Priority:	Essential
Functional area(s):	Simulator
Release:	1
Description:	It consists in developing a Moreau time discretization.
Notes and Questions:	

4.4.15 F-3.001: Transient analysis

Priority:	Essential
Functional area(s):	Simulator
Release:	1
Description:	It consists in doing the transient analysis.
Notes and Questions:	During each time step, SICONOS is called to solve a MLCP.

4.4.16 F-3.002: Non constant sources

Priority:	Essential
Functional area(s):	Simulator
Release:	1
Description:	

	Some currents and voltages source are not constant (RAMP, SIN,...). It consists in managing it for the transient analysis.
Notes and Questions:	

4.4.17 F-3.003: DC analysis

Priority:	Essential
Functional area(s):	Simulator
Release:	2
Description:	The DC analysis (Direct Current) determines the operating point of a circuit. There are no currents flowing through capacitors and zero voltages across inductors.
Notes and Questions:	

4.4.18 F-4.000: Gradient method

Priority:	Essential
Functional area(s):	Mixed Simulator
Release:	3
Description:	It consists in building the Jacobi matrix from the linear components.
Notes and Questions:	

4.4.19 F-4.001: Quasi-Newton methods

Priority:	Essential
Functional area(s):	Mixed Simulator
Release:	3
Description:	It consists in building a Hessian matrix approximation.
Notes and Questions:	

4.4.20 F-4.002: Write a mixed solver

Priority:	Essential
Functional area(s):	Mixed Simulator
Release:	3
Description:	It consists in writing an iterator algorithm using both Newton-Raphson and MLCP formulation. The non smooth components are described with a LCP formulation. The mixed solver makes a Newton-Raphson step for the smooth components and solves the LCP formulation. It is possible to formulate a step like a MLCP.
Notes and Questions:	

4.4.21 F-5.000: MLCP solver

Priority:	Essential
Functional area(s):	SICONOS
Release:	1
Description:	It consists in developing MLCP solver in SICONOS.
Notes and Questions:	

4.4.22 F-6.000: Specific electrical heuristics

Priority:	Must be defined
Functional area(s):	NS-ACEF
Release:	Not plan
Description:	It consists in adding electrical knowledges to help the MLCP solver. For example, to give the list of possible electrical modes is very efficient to accelerate the simulation.
Notes and Questions:	

5 Specification

5.1 Project Information

Project:	NS-ACEF
Internal Release Number:	1.0
Last update:	January 16, 2008
Related Documents:	Global architecture Functionalities

5.2 Parser

5.2.1 Parser Implementation

To develop a Netlist parser without using any existing code is a very big and long work.

A quicker solution is to use the parser of NGSPICE. It consists in extracting the parser and building a library. NGSPICE is under BSD license.(Ngspice legal issues).

Programming language is C.

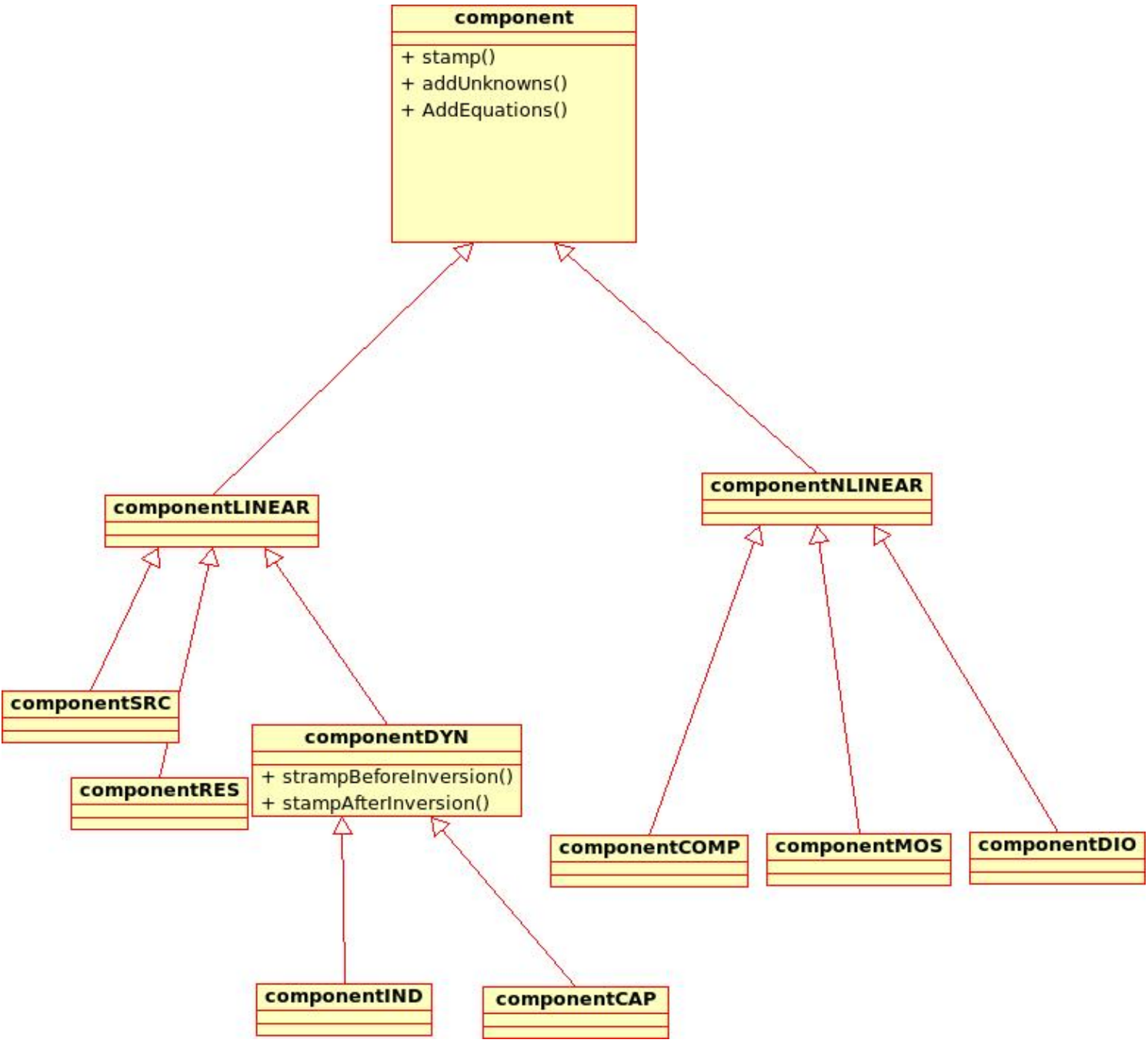
5.2.2 Parser API

- int readFile(char **file*) : Load a Netlist.
  - input : file name must be load.
  - return value : 1 if succes, 0 if echec.
- int initComponentList(char **type*) : Initialize an electrical component list of type *type* (RESITOR, CAPACITOR, INDUCTORS,...).
  - input : Component type.
  - return value : 1 if succes, 0 if echec.
- int nextComponent(void **data*) : Get information about a component.
  - output *data*: contains all informations about the electrical component.
  - return value : 0 if it is the last component, else 1.
- int getNbElementsOfType(char **type*)
  - return value : Component number of type *type*.
- void printCircuit() : Print Components list. Useful to test the parser.
- int getTransInfo(void **data*) : Get information about transient analysis.
  - output *data*: contains all informations about transient analysis.
  - return value : 1 if succes, else 0.

5.3 Automatic circuit equation formulation

It is a C++ module. Following the class diagram of mains class.

5.3.1 Components class

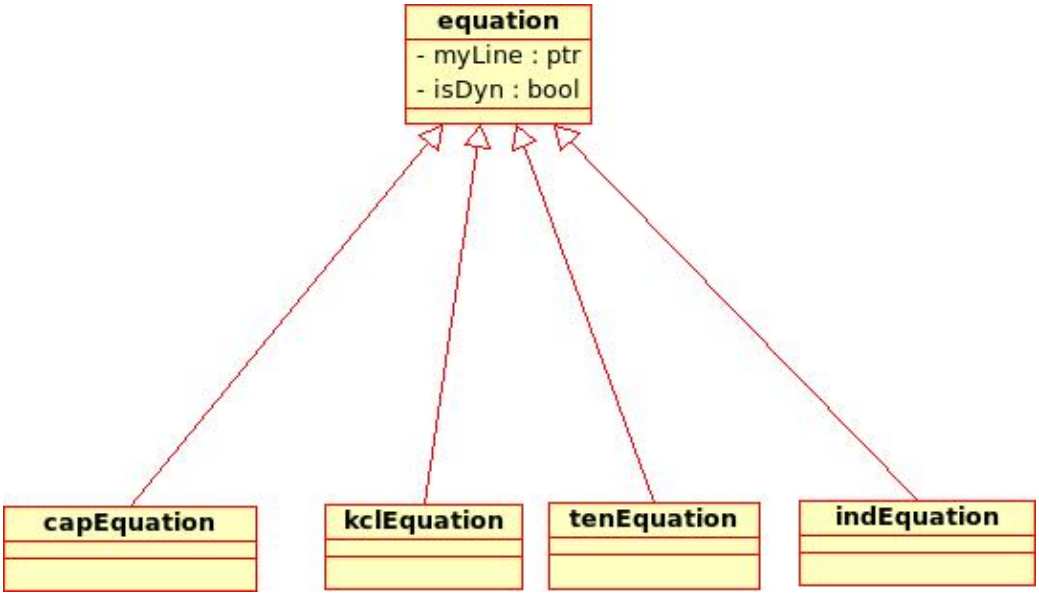


Each electrical component has a component instance in ACEF.

Component add its equations and unknowns. The main advantage is only that useful unknowns are added in the system. For example, current through a comparator is null, so this current does not appear in the system. Stamp method consists in writing component's contribution in the table equation.

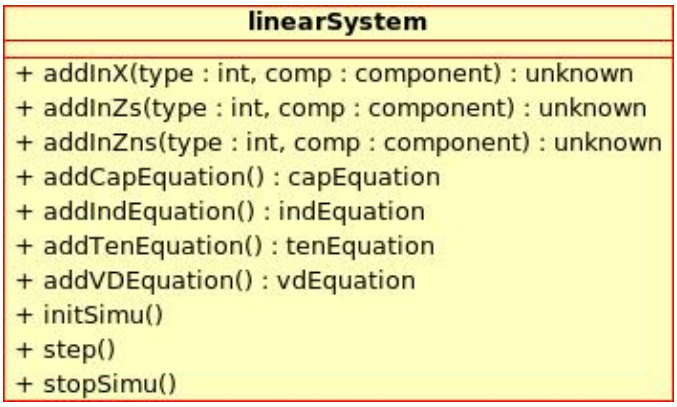
componentNLINEAR stamp methods is overloaded to fill the linear complementarity matrix formulation.

5.3.2 Equation class



The components build the equation instances. A equation is a line (member myLine) in the table equations. The component stamp method uses equation instances to fill the table equations.

5.3.3 Linear system class



It manages the memory allocation to build the matrix formulation. Component instances call method to add unknowns and equations in the system. It contains the MLCP formulation. There are also methods to run the simulation. This class calls the MLCP solver.

6 Release and delivery

6.1 Project Information

Project:	NS-ACEF
Last update:	January 16, 2008
Related Documents:	Functionalities

6.2 Release 1

It is planned for December 2007.

It contains the following functionalities :

- F-1.001 Store a Netlist.
- F-2.000 Linear, circuit RLC.
- F-2.001 Diode.
- F-2.002 Ideal transistor MOS.
- F-3.000 Time discretisation.
- F-3.001 Transient analysis.
- F-3.002 Non constante source.
- F-5.000 MLCP solver.

6.3 Release 2

It is planned for June 2008.

It includes the functionalities of the Release 1 and the following :

- F-1.002 Sub-circuit management.
- F-1.003 Transient analysis.
- F-2.003 Comparator.
- F-3.003 DC analysis.
- F-2.007 Forecast active set: use a local piecewise linear approximation to solve a smaller M.L.C.P.

It includes also a study on the following points :

- F-2.006 Multigrid algorithm: use a more simple piecewise linear approximation to accelerate the M.L.C.P. resolution.
- F-2.008 Non linear circuit: R,L or C non constant.

6.4 Release 3

It is planned for Decemder 2008.

It includes the functionalities of the Release 2 and the following :

- F-4.000 Gradient method.
- F-4.001 Quasi-Newton methods.
- F-4.002 Write a mixed solver.
- F-2.004 Diode, Transistor, Comparator with parameters.

It includes also a study on the following points :

- F-6.000 Specific electrical heuristics.

7 Tests plan

7.1 Project Information

Project:	NS-ACEF
Internal Release Number:	1.0
Last update:	January 16, 2008
Related Documents:	Functionalities

7.2 Parser library tests

It consists in checking if Netlist is correctly interpreted. Indeed Netlist contains macro must be evaluated. The program test displays the interpreted Netlist.

7.3 Automatique circuit equation formulation

It consists in checking the table equation is correct. The program test displays the table equation and we have to check that physical law are correctly wrote.

7.4 Transient analysis

The test consists in comparing result with a SPICE simulation.

7.5 Automatic tests

A script runs the program on a list of Netlist. For each circuit file, a file is created. This file contains the result of the parser, the automatic equations formulation and the transient analysis.

8 Project organisation

8.1 Project Information

Project:	NS-ACEF
Internal Release Number:	1.0
Last update:	January 16, 2008

8.2 Organizational roles and responsibilities

Due to the number of participants in the software project development, the organization of the project is relatively simple. The NS-ACEF is led by Vincent ACARY.

The team for the design and the development are defined as follows :

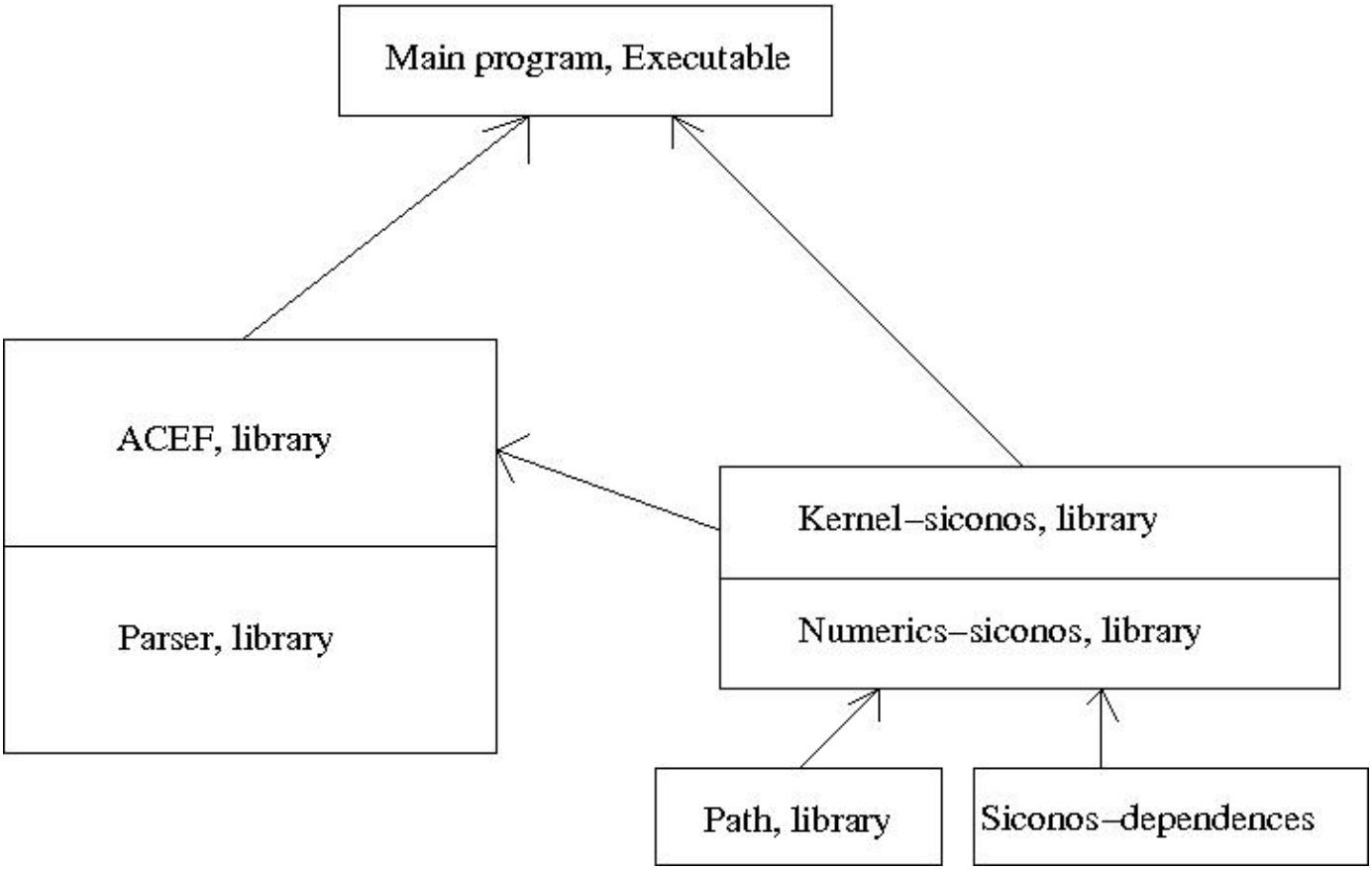
1. Team INRIA :
    - Vincent Acary (Team Leader) Vincent.Acary@inrialpes.fr
    - Roger Pissard-Gibollet (Software quality leader)
    - Franck Pérignon (SICONOS expert and programmer)
    - Olivier Bonnefon (designer, programmer, test engineer)
    - Pascal Denoyelle (designer, programmer, test engineer)

9 Software delivery 1, July 2008

9.1 Project Information

Project:	NS-ACEF
Internal Release Number:	1.0
Last update:	January 16, 2008
Related Documents:	Functionalities

9.2 Software installation



9.2.1 Requisite

ACEF requires the siconos platform, and optionelly, Path (http://pages.cs.wisc.edu/~ferris/path.html), an external library used by siconos.

This delivery version is built with the pathFerris demo library. So the path library must be copied somewhere in the LD_LIBRARY_PATH, example:
cp ...libpath46.so /usr/lib/
ACEF needs the boost graph module:
sudo apt-get install libboost-graph1.33.1
sudo apt-get install libboost-graph-dev
Siconos needs a library a DefaultPlugin.so, it must copie in the LD_LIBRARY_PATH, example:
ln -s /usr/share/siconos-kernel/DefaultPlugin.so /usr/lib/DefaultPlugin.so

9.2.2 Installation of acef and parser libraries

These libraries are installed with the command:
If a previous version has been installed, remove the previous version of siconos.
Now, the packages can be installed:
->**sudo dpkg -i siconos-spiceparser-3.0.0-r124--i386-linux-debian-etch.deb**
->**sudo dpkg -i siconos-numerics-3.0.0-r1262--i386-linux-debian-etch.deb**
->**sudo dpkg -i siconos-kernel-3.0.0-r1262--i386-linux-debian-etch.deb**
->**sudo dpkg -i siconos-acef-3.0.0-r124--i386-linux-debian-etch.deb**

9.2.3 Main program compilation

The source code mainSiconos.cpp must be compiled on your computer. The file *script* contains an example to compile the main program.
The generated executable is noselect, usage:
-> **noselecs InverterChain10.cir DIRECT_PATH**

9.3 Functionalities

This delivery contains all Functionalities of the release 1.

9.3.1 Software input: a Netlist

The input of the main program is like a spice Netlist. The NGSPICE User Manual contains all the informations about the netlist format (http://ngspice.sourceforge.net/docs.html).
The following figure presents the circuit elements managed in ACEF.

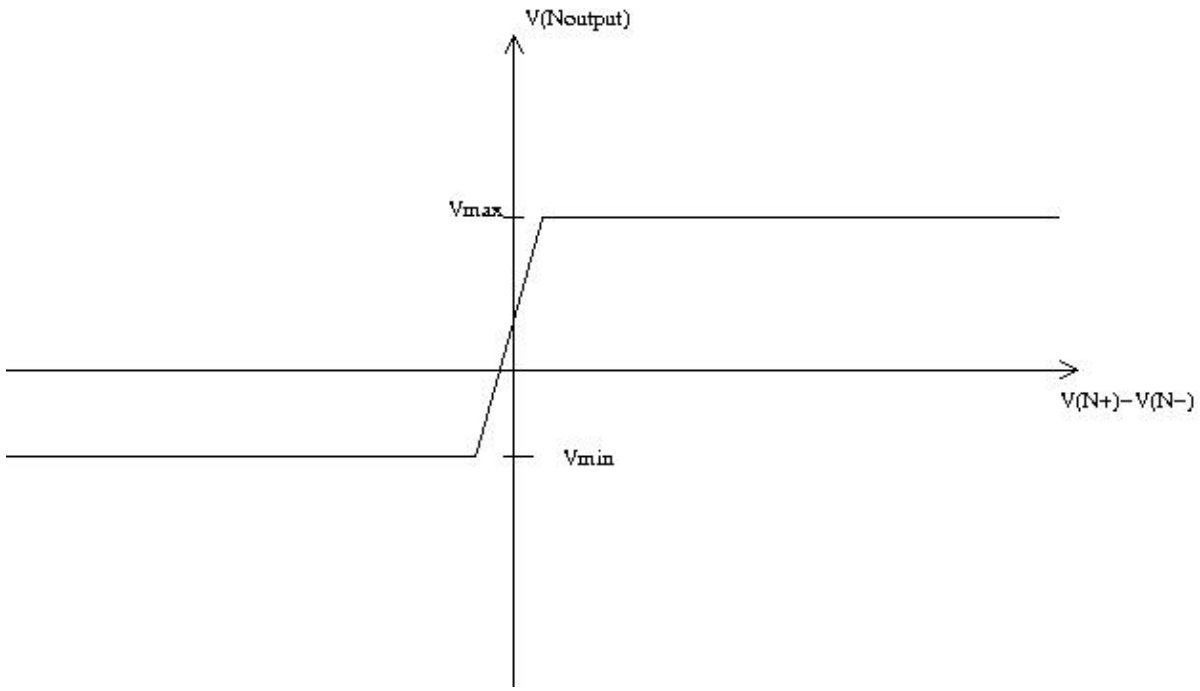
Name	code	example	note
Resistors	R	R1 0 1 10K	It is a constant value.
Capacitors	C	C1 0 1 10pf	It is a constant value. Model are ignored.
Inductors	L	L1 n0 n1 15mh	It is a constant value. Model are ignored.
Independent voltage and current	PULSE, SIN, EXP	PULSE(0 1.8 0 0.1e-3)	See Spice documentation for more information.
Dependent voltage and current	G, E, F, H, B	G1 2 0 5 0 0.1MMHO	See Spice documentation for more information.
Diode model	D	D1 0 3 DIOMODEL	ACEF uses an ideal model. Model are ignored.
Transistor PMOS/NMOS	M	M106 c1 h2 0 0 mosP_Sah	ACEF uses a piecewise linear model.
Comparator	.comp	.comp1 6 5 7 Vmin=0 Vmax=3 Vepsilon=0.1	ACEF uses a piecewise linear model.

9.3.1.1 About the transistor model

Example : **.model mosP_Sah PMOS LEVEL=1 KP=3.24e-5 VT0=0.6**
A model transistor is a necessary information. It specifies witch kind of transistor is used.
LEVEL=1 is a necessary option for the parser, but is not used in ACEF.
VT0 is the zero-bias voltage.
KP is the transconductance parameter.
The other parameters of the model are ignored.
ACEF use a piecewise linear model for the transistor. The number of hyperplan is a parameter available in the main program.

9.3.1.2 The comparator component

The **.comp** is a specific component of ACEF. It describes a comparator. The syntax is :
.comp N+ N- Noutput Vmin=0 Vmax=3 Vepsilon=0.1
if V(N+) - V(N-) > Vepsilon/2 then V(Noutput) must be Vmax.
if V(N+) - V(N-) < -Vepsilon/2 then V(Noutput) must be Vmin.
else V(Noutput) is an interpoled value.



9.3.2 Transient Analysis

The **.tran** line contains the time interval of the simulation, Example :
.tran 1ns 100ns specified a simulation from 0 to 100ns with a fixed time stepping of 1ns.

9.3.3 Output

The **.input** line defines the output of the simulation, Example:
.print tran V(4) V(23,27). The output could be any voltage or any tension. This software version does not display currents.
The result is written in a file *circuit.sim*.

9.3.4 Initial conditions

The **.ic** line defines the initial conditions, Example:
.ic V(10)=1
This line forces the initial node voltages.