



Joni-Matti Määttä, Mikko Honkonen,
Tommi Korhonen

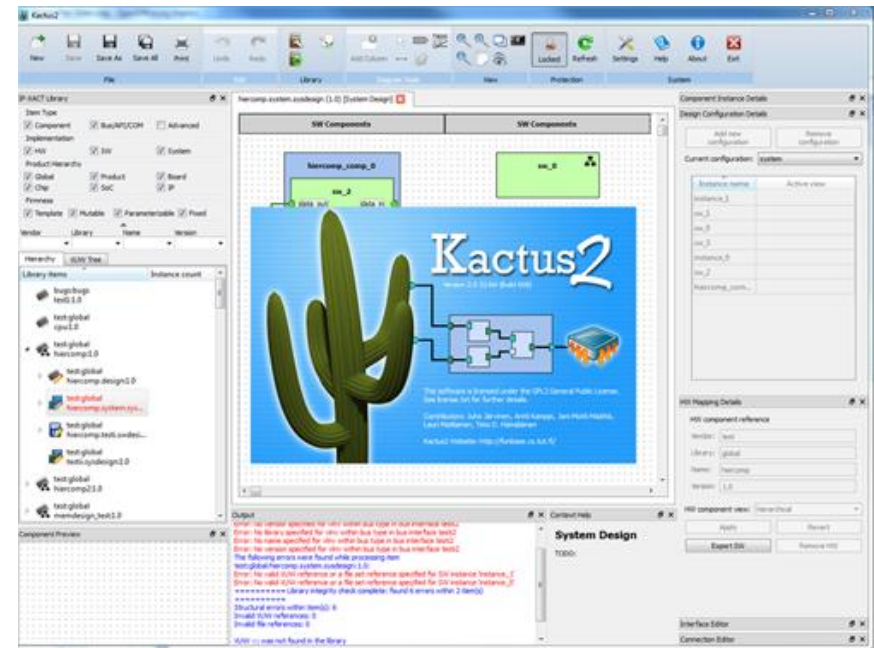
Dependency Analysis for Kactus2

Background

■ Client: Timo Hämäläinen

■ Kactus2

- Developed at TUT
- Open-source toolset to design embedded products, especially FPGA-based MP-SoCs
- Based on IP-XACT XML metadata and design methodology



Background 2

- Implemented in C++ and Qt
- Kactus2 goals
 - Easier IP reusability
 - Practical HW/SW abstraction for easier application SW development
 - The most user-friendly EDA tool
- Handling file dependencies is essential for good usability

Problem

- Both hardware and software projects include high amounts of files dependent on other files
 - Class instantiation in C++
 - Component instances in VHDL
 - Not only strict code dependencies – for example a hardware component might be dependent on a block specification document
- As project size increases, it becomes more and more difficult to keep track of these file dependencies
- What files might a change in this file have an effect on?

Dependency Analysis Tool: Overview

- Automates the file dependency tracking by scanning supported files and keeps track of file changes
- The user should be able to see with one quick glance if a change has been made in a file another file depends on
- As not all dependencies can be found automatically, user should also be able to specify dependencies manually
- Automatic scan support initially planned for C, C++ and VHDL, but the system should be modular so that it is easy to implement additional file types

Dependency Analysis Tool: Features

- Choosing and Managing Source Directories
- Running Dependency Analysis
- Manual Connections
- Editing Existing Connections
- Filtering Connections
- External Dependencies

Dependency Analysis Tool: UI

	Status	Path	↕	Dependencies	Referenced by
-	●	source/common/		●	
	●	source/common/utls.c		●	
	●	source/common/utls.h		●	
-	●	source/program/		●	
	●	source/common/main.c		●	
-	●	documentation/		●	
	●	documentation/architecture.uml		●	
-		External		●	
		string.h		●	

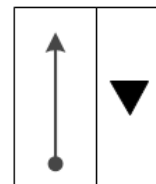
<
>

Dependency information

Description

The main routine uses functions from the utility module.

Direction



Dependency Analysis Tool: Requirements

- Must seamlessly integrate with Kactus2
- Must be able to handle thousands of files
 - Without breaking
 - Within reasonable time
- Must be easy to use

Project Group

■ 3 group members

- Joni-Matti: project manager, SW architect and developer
 - 6th year student at TUT
 - Original member of the Kactus2 team
 - Over 10 years of experience in C++
- Mikko: Test lead and development tasks
 - 5th year student at TUT
 - Studies in both hardware and software
 - A good amount of experience in C++
- Tommi: UI specialist and SW developer
 - 5th year student at TUT
 - Studies in both hardware and software

Schedule

- Estimated total time: 619 hours
- Schedule for 2012
- Planning and writing specification documents
- Schedule for 2013
- SW development and testing

Schedule 2012

ID	Task name	2012-11 (63)				2012-12 (59)				Hours per task 2012
		45	46	47	48	49	50	51	52	
T1	Functional specification	12	12	12		12	12			60
T2	Technical specification					10	10	0		20
T3.1	Dependency analysis algorithm interface									0
T3.2	C++ dependency analysis algorithm									0
T3.3	C++ hash calculation									0
T3.4	Dependency parser settings									0
T3.5	VHDL dependency analysis algorithm									0
T3.6	VHDL hash calculation									0
T4.1	Basic table-style widget									0
T4.2	Dependency graph visualization									0
T4.3	Manual dependencies									0
T4.4	Locking dependencies									0
T4.5	Filter views									0
T4.6	Dependency change visualization									0
T5.1	Folder and file scanning									0
T5.2	Saving graph data in XML									0
T6	Testing									0
T7.1	Group meetings	3	3	3		3	3	3		18
T7.2	Client meetings		3			3		3		9
T7.3	Assistant meetings				3					3
T7.4	Project plan presentation	2	4	6						12
T7.5	Final presentation									0
T7.6	Final report									0
	Hours per week	0	17	22	24	18	25	16	0	122
	Deliverables						D1			
	Meetings				A1					



Schedule 2013

ID	Task name	2013-01 (182)					2013-02 (159)				2013-03 (103)				2013-04 (53)			Hours per task 2013
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
T1	Functional specification																	0
T2	Technical specification	18	12															30
T3.1	Dependency analysis algorithm interface		3	3														6
T3.2	C++ dependency analysis algorithm		4	4	4	4	4											20
T3.3	C++ hash calculation		4	4														8
T3.4	Dependency parser settings		2	2	2	2	2											10
T3.5	VHDL dependency analysis algorithm						5	5	5	5	5							25
T3.6	VHDL hash calculation						1	1	2	2	2							8
T4.1	Basic table-style widget		10	10	10	10	10											50
T4.2	Dependency graph visualization				10	10	10	10	10	10	10							70
T4.3	Manual dependencies						3	3	3	3	3							15
T4.4	Locking dependencies						2	2	2	2	2							10
T4.5	Filter views										3	2	2	3				10
T4.6	Dependency change visualization										3	4	4	4				15
T5.1	Folder and file scanning		2	1	2													5
T5.2	Saving graph data in XML				6	6	6	7										25
T6	Testing		4	4	4	4	6	6	6	8	8	10	10	10				80
T7.1	Group meetings		3	3	3	3	3	3	3	3	3	3	3	3	3	3		42
T7.2	Client meetings			3		3		3		3		3		3		3		21
T7.3	Assistant meetings				3										4			7
T7.4	Project plan presentation																	0
T7.5	Final presentation														3	6	6	15
T7.6	Final report														5	10	10	25
	Hours per week	18	44	34	44	42	52	40	31	36	39	22	19	23	15	22	16	497
	Deliverables		D2													C1	FR	
	Meetings				A2										A3			



Budget (Imaginary)

- No special equipment required
 - Salary costs the only major cost in the project
- Any tools used also available for free or with free licensing for students
- Some additional costs from travel (meetings at the university) and materials such as paper
- Funding for the project would come from TUT

Personnel costs	
Working hours	619
Salary per hour	14 €
Salaries for project	8 666 €
Indirect employee costs (40%)	3 466 €
Personnel costs total	12 146 €
Other costs	
Travel (VAT excluded)	50 €
Equipment	0 €
Licences	0 €
Consumables and material	50 €
Subcontracting	0 €
Other costs	0 €
Other costs total	100 €
Total project expenses	12 246 €
Profit	0 €
Price of the project	12 246 €
Price of the project (vat 23% inc)	15 063 €

Project Management

- Weekly meetings
 - Review of what has been done on the past week
 - Decisions what will be done on the next week
 - Open questions are addressed
- Client meetings once in every two weeks
 - Especially important when making functional specification
- Redmine
 - Free project management software
 - Centralized place for tasks, issues and time management

SWOT

■ Strengths

- The project topic is quite simple and does not require too much special skills
- All members have experience in C++
- Joni-Matti has experience in other projects and project management

■ Weaknesses

- Two members of the project group have only basic experience in Qt

SWOT 2

■ Opportunities

- All team members learn new skills in project management and Qt development
- If Kactus2 becomes popular, the dependency analysis tool will be well-known and used by many

■ Threats

- Joni-Matti has multiple ongoing projects at the moment

Conclusion

- The project has a strict scope
 - Good chances to get everything done in schedule
- All members have enough skills to accomplish tasks
 - Large-scale learning not required during the project
- The project results are immediately taken into use in Kactus2
 - Direct benefits for the community