Stellingen

behorende bij het proefschrift

Rule-based Compilation of Data-parallel Programs

Leo Breebaart

1. Een op herschrijfregels gebaseerde, voor de gebruiker programmeerbare compiler opent deuren die bij het gebruik van conventionelere 'black box' vertaaltechnieken gesloten blijven.
   [dit proefschrift]

2. Voor efficiënte herschrijfregels is de volledige en directe toegang tot de ontleedboom onontbeerlijk.
   [dit proefschrift]

3. David Gries' programmavertalingsregel: "Stel nooit uit tot run time wat tijdens de compilatie gedaan kan worden" [*Compiler Construction for Digital Computers*, 1971] is algemener toepasbaar: "Genereer nooit dynamisch wat van te voren bepaald kan worden". Deze regel zou bijvoorbeeld ook bij het bouwen van websites in acht genomen moeten worden.

4. De immer toenemende complexiteit van software-systemen zal alleen beheersbaar blijven als de implementaties van deze systemen zelf, meer dan tot nu toe, door software gegenereerd worden.

5. Omdat gebruikers programmeertalen en -systemen terecht niet op hun potentie, maar op hun eerste implementaties beoordelen, is de kwaliteit van deze implementaties van cruciaal belang.

6. De object-gebaseerde aspecten van C++ zitten te vaak de template-gebaseerde aspecten in de weg, en vice versa.

7. Voor de kwaliteit en onderhoudbaarheid van software is het actief ontwikkelen en gebruiken van regressie-testen nog belangrijker dan het schrijven van documentatie.

8. Fabrikanten van consumentenelectronica, PDAs en zelfs bureaucomputers onderschatten de mate waarin de consument zich bindt aan de *software* die op deze apparaten draait, in plaats van aan de apparaten zelf.

9. Linux is alleen maar gratis als je tijd niets waard is.
   [*Jamie Zawinski*, `http://www.jwz.org/`, 1998]

Deze stellingen worden verdedigbaar geacht en zijn als zodanig goedgekeurd door de promotor, Prof. dr. ir. H.J. Sips.

Propositions

accompanying the thesis

Rule-based Compilation of Data-parallel Programs

Leo Breebaart

1. A rule-based, user-programmable compiler opens doors that remain closed when more conventional 'black box' compilation techniques are used.
   [this thesis]

2. Full and direct access to the parse tree is indispensable for efficient rewrite rules.
   [this thesis]

3. David Gries' program translation maxim: "Never put off until run time what you can do at compile time" [*Compiler Construction for Digital Computers*, 1971] is more broadly applicable: "Never generate dynamically what you can determine beforehand". This rule should for example also be heeded when building websites.

4. The ever-increasing complexity of software systems will only remain manageable if the implementations of these systems themselves will, to a larger extent than is currently the case, be generated by software.

5. As users rightly judge programming languages and systems not by their potential, but by their first implementations, the quality of these implementations is crucially important.

6. The object-based aspects of C++ are too often in the way of the template-based aspects, and vice versa.

7. For the quality and maintainability of software the active use and development of regression tests is even more important than writing documentation.

8. Manufacturers of consumer electronics, PDAs, and even desktop computers underestimate the extent to which the consumer bonds with the *software* that runs on these devices, rather than with the devices themselves.

9. Linux is only free if your time has no value.
   [*Jamie Zawinski*, `http://www.jwz.org/`, 1998]

These propositions are considered defendable and as such have been approved by the supervisor, Prof. dr. ir. H.J. Sips.