

University of Groningen

Building Product Populations with Software Components

Ommering, Robbert Christiaan van

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2004

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Ommering, R. C. V. (2004). *Building Product Populations with Software Components*. s.n.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Stellingen

behorend bij het proefschrift

**Building Product Populations
with Software Components**

van

Rob van Ommering

1. Architectuur wordt vaak gedefinieerd als ‘de componenten en hun onderlinge structuur in een systeem’. Een betere definitie is ‘de specificatie van een ontwerp’.

Dit proefschrift, hoofdstuk 2.

2. ‘Late compile-time binding’ is niet een contradictio in terminis, maar juist een geschikte combinatie van ‘late binding’ en ‘static binding’.

Dit proefschrift, hoofdstuk 3.

3. Een uit componenten opgebouwde productfamilie moet bestand zijn tegen het onafhankelijk evolueren van de componenten.

Dit proefschrift, hoofdstuk 4.

4. Hoe groter de variatie in producten is, des te meer is compositie en niet variatie de geschikte realisatietechniek.

Dit proefschrift, hoofdstuk 5.

5. Compositie is des te beter mogelijk naarmate van meer contextafhankelijkheden variatiepunten gemaakt zijn.

Dit proefschrift, hoofdstuk 5.

6. Een ‘configuration management system’ is niet de meest geschikte oplossing voor het beheren van configuraties in een product familie.

Dit proefschrift, hoofdstuk 6.

7. Het opzetten van een productielijn voor software vergt niet alleen een goede technische oplossing, maar ook een zakelijke drijfveer en een aangepast ontwikkelproces en -organisatie.

Dit proefschrift, hoofdstuk 7.

8. Gedistribueerde besturing functioneert vaak beter dan hiërarchische besturing.
Dit proefschrift, hoofdstuk 8.
9. Wie het probleem niet ziet, die snapt ook de oplossing niet.
10. Gelijk krijgen is niet moeilijk; gelijk hebben is de kunst.
11. Een jaar onderzoek is zinloos als het ook een jaar kost om het uit te leggen.
12. Niet het weer maar het weerbericht is de grotere veroorzaker van verkeersdrukte.
13. Auto's waar je net niet tegenaan botst hebben op de verwachte plaats van impact vaak al een deuk.
14. Dertien stellingen zijn genoeg voor een proefschrift.

Deze stellingen worden verdedigbaar geacht en zijn als zodanig goedgekeurd door Prof. dr. ir. J. Bosch.

1. Architecture is often defined as ‘the components of a system and their interrelations’. A better definition is ‘the specification of a design’.
This thesis, chapter 2.
2. ‘Late compile-time binding’ is not a contradiction, but rather a suitable combination of ‘late binding’ and ‘static binding’.
This thesis, chapter 3.
3. A product family that is constructed from components should be able to deal with the independent evolution of components.
This thesis, chapter 3.
4. The larger the variation in products is, the more is composition rather than variation the right technique to realize the product family.
This thesis, chapter 4.
5. Composition becomes better possible when more and more context dependencies are turned into variation points.
This thesis, chapter 5.
6. A configuration management system is not the most appropriate solution to manage configurations in a product family.
This thesis, chapter 6.
7. Setting up a product line does not only require a proper technical solution, but also a business driver and a tuned development process and organization.
This thesis, chapter 7.
8. Distributed control often functions better than hierarchical control.
This thesis, chapter 8.
9. If you do not see the problem, you will not understand the solution.
10. To *be* right is more important than to be *judged* right.
11. It should not take a year to explain a year of research.
12. Not the weather but the weather forecast is the greater cause of traffic jams.
13. Cars with which you have just avoided a collision often already have a dent at the predicted point of impact.
14. Thirteen statements are enough for a PhD thesis.