

# Stellingen

behorende bij het proefschrift

## Towards Continuous Knowledge Engineering

door Klaas Schilstra

1. De continuous knowledge engineering aanpak voorgesteld in dit proefschrift is een instantie van de aanpak voorgesteld door Rodney Brooks (1991, pg. 139): “We must incrementally build up the capabilities of intelligent systems, having complete systems at each step of the way and thus automatically ensure that the pieces and their interfaces are valid. At each step we should build complete intelligent systems that we let loose in the real world with real sensing and real action. Anything less provides a candidate with which we can delude ourselves.” [dit proefschrift]
2. De wetenschap kan een even krachtige metafoor zijn voor de constructie van systemen als ingenieurschap. In die gevallen waar de beperkingen en eisen niet volledig bekend zijn zelfs krachtiger. Volgens de gesitueerde mantra is dit altijd het geval, omdat de beperkingen en eisen in de context ingebed zijn. [dit proefschrift]
3. De expert is de beste kennis ontwikkelaar. [dit proefschrift]
4. Er is geen compleet kennissysteem. [dit proefschrift]
5. De realiteit van elke ontwikkelmethodiek wordt in grote mate bepaald door de mogelijkheden van gereedschappen om haar praktisch bruikbaar te maken. [dit proefschrift]
6. Het is onwaarschijnlijk dat kennissysteem ontwikkeling nieuwe inzichten oplevert, tenzij de ontwikkelomgeving gemakkelijk genoeg is en uitnodigt tot experimentatie. [dit proefschrift]
7. Elke technologie die voldoende los staat van de wetenschappelijke methode is niet te onderscheiden van ritueel. Elke wetenschappelijke inspanning die voldoende los staat van praktische toepassing is niet te onderscheiden van religie.
8. Soms is een simulatie de werkelijkheid.
9. Spellen zijn geen kinderspel voor kunstmatige intelligentie.
10. Naar mate de verschillen tussen kunstmatige en menselijke intelligentie verdwijnen, zal de ophef die over die verschillen gemaakt wordt ongetwijfeld sterk aanzwellen.
11. Het is nutteloos om deze stelling te vertalen in het Nederlands.
12. Herbruikbaarheid van software componenten wordt versterkt door het aantal aannames die gemaakt worden over de omgeving te verminderen. Ergo, des te inhoudslozer de component, des te beter mogelijkheden voor hergebruik.

# Thesis Propositions

for

## Towards Continuous Knowledge Engineering

by Klaas Schilstra

1. The continuous knowledge engineering approach proposed in this thesis is an instance of the approach proposed by Rodney Brooks (1991, pg. 139): “We must incrementally build up the capabilities of intelligent systems, having complete systems at each step of the way and thus automatically ensure that the pieces and their interfaces are valid. At each step we should build complete intelligent systems that we let loose in the real world with real sensing and real action. Anything less provides a candidate with which we can delude ourselves.” [this thesis]
2. Science can be just as powerful a metaphor to use for the construction of systems as engineering. In those cases where the constraints and requirements are not known fully in advance, even more powerful. The situated mantra would exclaim that one never knows the constraints and requirements, as they're embedded in the context. [this thesis]
3. The expert is the best knowledge engineer. [this thesis]
4. There is no final knowledge system. [this thesis]
5. The reality of any method of development is determined to an overwhelming extent to the ability of tools to support its viability and practicality. [this thesis]
6. Knowledge engineering is unlikely to yield new insights, unless the development environment is convenient to use and inviting to experiment. [this thesis]
7. Any technology sufficiently detached from scientific method is indistinguishable from ritual. Any scientific endeavour sufficiently detached from practical application is indistinguishable from religion.
8. Sometimes the simulation is actually the real thing.
9. Games are not a toy domain for artificial intelligence.
10. As the differences between artificial and human intelligence diminish, the fuss that will be made about those differences will undoubtedly increase.
11. It is useless to translate this proposition into Dutch.
12. Diminishing the number of assumptions that a software component makes on the environment strengthens its reusability. Ergo, as a component has less substance, its reusability increases.