Testen van proces-communicatie en synchronisatie van ITP LoadBalancer software via model-checking

Marko van Eekelen[1], Stefan ten Hoedt[2], René Schreurs[2], Yaroslav S. Usenko[3]

[1] LaQuSo Nijmegen, Nijmeegs Instituut voor Informatica en Informatiekunde,
Radboud Universiteit Nijmegen, Toernooiveld 1, 6525ED Nijmegen
[2] Aia Software BV, Kerkenbos 10-129, 6546 BJ, Nijmegen
[3] LaQuSo Eindhoven, Faculteit Wiskunde en Informatica,
Technische Universiteit Eindhoven, Den Dolech 2, 5600MB, Eindhoven.

Aia Software BV [1] voert wereldwijd het product ITP. Het ITP Document Platform stelt organisaties in staat in een schaalbare omgeving gepersonaliseerde bedrijfskritische documenten te maken. Deze applicatie heeft een LoadBalancer, een proces-kernel waarin diverse servers en clients met elkaar communiceren en taken verdelen en uitvoeren. Dit systeem komt af en toe in een ongewenste toestand terecht. LaQuSo [2] heeft onderzocht in hoeverre de proces-communicatie en synchronisatie van deze LoadBalancer gemodelleerd en geanalyseerd kon worden ten einde tot een advies te komen hoe ongewenste situaties kunnen worden voorkomen.

Projectuitvoering

Het project is in de volgende fasen uitgevoerd:

- In overleg met twee medewerkers van Aia Software (Stefan ten Hoedt en René Schreurs) is een idee verkregen van de structuur en het gedrag van de software in het algemeen en van de relevante onderdelen voor modellering in het bijzonder.
- Het relevante gedeelte is gemodelleerd in mCRL2 [3] . Uitgegaan is van de C code van het relevante gedeelte. Vrij dicht tegen de C code aan, is een model gemaakt van de sessielaag van het LoadBalancer protocol. Zowel van de hierboven gelegen applicatielaag als van de hieronder gelegen TCP-socketlaag is een abstractie gemodelleerd.
- De code en het model zijn door de LaQuSo-modelleur en de Aia-ontwerper in een aantal sessies zorgvuldig gereviewed teneinde de overeenkomsten zo groot mogelijk te maken. Dit heeft geleid tot een aantal aanpassingen in het model. Tevens heeft dit een aantal vragen

over de code opgeleverd en een aantal concrete wenselijke eigenschappen die voor analyse in aanmerking kwamen.
- Het model is geanalyseerd met behulp van modelchecking technieken van de mCRL2 Toolset op deadlock-vrijheid en een aantal andere samen met de klant opgestelde starvation en consistentie eigenschappen. Dit bracht 5 problemen in de C code aan het licht. Deze problemen zijn door Aia Software geaccepteerd en in de applicatie verbeterd.
- In het model zijn de verbeteringen aangebracht. Met het resulterende model zijn de gewenste eigenschappen allemaal volledig geverifieerd (zij het voor een beperkte configuratie). Daarnaast is aan Aia Software advies uitgebracht over verbetering van de software door generatie van een groot codegedeelte via een statetransitietabelmethode.

Resultaten en Conclusies

De analyse heeft geleid tot een aantal relevante verbeteringen van probleemsituaties. Verder is er advies uitgebracht ter verbetering van de opzet van een deel van de code. De gewenste eigenschappen zijn daadwerkelijk geverifieerd. Naar verwachting zullen in de toekomst probleemsituaties in de software dus minder vaak voorkomen. Het vertrouwen in de software is hiermee gegroeid.

mCRL2 kon gebruikt worden in deze bedrijfsmatige setting van een server voor documentproductie. Een gedeelte van de operatingsysteem services (sockets, lock, events, etc.) kon hiermee gemodelleerd worden. Helaas kon verificatie slechts voor een beperkte configuratie gebeuren. Het zou wenselijk zijn als de mCRL2 toolset verbeterd kon worden zodat een grotere configuratie geverifieerd kan worden. Ook een geautomatiseerde controle op conformance tussen code en model zou interessant kunnen zijn.

AIA heeft nu een werkend reference model van een cruciaal onderdeel van de LoadBalancer software. In principe kunnen ze nu zelf wijzigingen van de code naar het model brengen en in het model de eigenschappen daarvan analyseren.

Het project was succesvol. De LoadBalancer code is verbeterd dankzij de modelchecking resultaten. Gewenste eigenschappen zijn geverifieerd. Modellering was relatief tijdrovend aangezien dit een belangrijke reverse engineering component bevat. Dit kan verbeterd worden als het bedrijf eerst een statetransitiemodel maakt. Hiermee kan ook de code – model conformance beter gegarandeerd worden via

gedeeltelijke code generatie. Een mogelijk vervolgproject zou een certificatietraject kunnen zijn waarbij meerdere eigenschappen uitgebreider geverifieerd worden.

Links

[1] Aia Software BV, http://www.aia-itp.com
[2] Laboratory for Quality Software (LaQuSo), http://www.laquso.com
[3] mCRL2 Toolset, http://www.mcrl2.org

Dr. M.C.J.D. van Eekelen (Marko) received his master's degree in mathematics and his Ph.D. degree in computer science (promotor Prof. H.P. Barendregt, co-promotor Dr. M.J. Plasmeijer) from Radboud University Nijmegen. He is an associate professor at the Security of Systems Department in the Institute for Computing and Information Science in the Nijmegen Faculty of Science. Furthermore, he is the director of the Nijmegen part of the Laboratory for Quality Software (LaQuSo), which is a collaboration of Technical University of Eindhoven and Radboud University Nijmegen. His research interests are in software certification, formal methods in software construction, software analysis, software verification and validation, and in industrial applications of these. He recently received an NWO grant for the Amortized Heap space consumption Analysis project.

Drs. S. ten Hoedt (Stefan) received his master's degree in computer science from the Radboud University in Nijmegen in 1995. Since 1996 he works for Aia Software as a consultant and software engineer.

Drs. R. Schreurs (René) received his master's degree in computer science from the Radboud University in Nijmegen in 1994. He started working for Aia Software in 1993; initially as a consultant and later as a software engineer. Since 2003 he is a Manager Product Development.

Dr. Y.S. Usenko (Yarick) received his master's degree in computer science from Taras Shevchenko University of Kyiv in 1997, and his Ph.D. degree in computer science from Technical University of Eindhoven in 2002. In 1996-1997 he was a fellow at the United Nations University / International Institute for Software Technology (UNU/IIST) in Macau. In 1998-2002 he worked at the Center for Mathematics and Computer Science (CWI), as a project researcher. In 2002-2004 he worked at the Formal Methods and Tools Group, University of Twente as a PostDoc.Since 2004 he works at Laboratory for Quality Software (LaQuSo), Technical University of Eindhoven, as a researcher and software engineer. His research interests are in formal methods in software engineering, software verification and validation methods, process algebra, and in industrial applications of these. He has participated and taken part in organization of several European and Dutch research projects.