Communicating Process Architectures 2008
P.H. Welch et al. (Eds.)
IOS Press, 2008

Designing Animation Facilities for gCSP

Hans T.J. VAN DER STEEN, Marcel A. GROOTHUIS and Jan F. BROENINK

Control Engineering, Faculty EEMCS, University of Twente,
P.O. Box 217 7500 AE Enschede, The Netherlands.

{T.T.J.vanderSteen, M.A.Groothuis, J.F.Broenink} @utwente.nl

Abstract. To improve feedback on how concurrent CSP-based programs run, the graphical CSP design tool (gCSP [3,2]) has been extended with animation facilities. The state of processes, constructs, and channel ends are indicated with colours both in the gCSP diagrams and in the composition tree (hierarchical tree showing the structure of the total program). Furthermore, the contents of the channels are also shown. In the Fringe session, we will present and demonstrate this prototype animation facility, being the result of the MSc project of Hans van der Steen [5], and ask for feedback.

Keywords. graphical CSP tools, IDE, code generation.

Outline

The CTC++ run time library [1,4] has been augmented, such that it generates the status information gCSP needs. The content of the ready queue of the scheduler is also made available.

The animation is controlled from within the gCSP design tool. Breakpoints and the animation speed can be set. Choosing the animation speed between about 0.2 and 1 s (i.e. time duration between two state changes of processes, constructs or channel ends) allows the user to follow the behaviour of the program. The execution history (the stream of status events coming from the running CSP program) is shown in a log window. The stream of status events can be filtered, to focus on those parts of the program one is interested in. The contents of the channels and of the ready queue are shown in a separate log window.

Tests were performed on the practical use of animation, the execution behavior of gCSP models and the C++ code generator of gCSP. Using a prototype version in our MSc class on real-time software development showed that this animation helps the students' understanding of concurrency. At least, significantly fewer questions were asked during the lab exercises.

References

[1] G.H. Hilderink, A.W.P. Bakkers, and J.F. Broenink. A Distributed Real-Time Java System Based on CSP. In *The third IEEE International Symposium on Object-Oriented Real-Time Distributed Computing ISORC 2000*, pages 400–407. IEEE, Newport Beach, CA, 2000.

[2] D.S. Jovanovic. *Designing dependable process-oriented software, a CSP approach*. PhD thesis, University of Twente, Enschede, NL, 2006.

[3] Dusko S. Jovanovic, Bojan Orlic, Geert K. Liet, and Jan F. Broenink. gCSP: A Graphical Tool for Designing CSP systems. In Ian East, Jeremy Martin, Peter H. Welch, David Duce, and Mark Green, editors, *Communicating Process Architectures 2004*, pages 233–251. IOS press, Oxford, UK, 2004.

[4] Bojan Orlic and Jan F. Broenink. Redesign of the C++ Communicating Threads library for embedded control systems. In Frank Karelse, editor, *5th PROGRESS Symposium on Embedded Systems*, pages 141–156. STW, Nieuwegein, NL, 2004.

[5] T.T.J. van der Steen. Design of animation and debug facilities for gCSP. MSc Thesis 020CE2008, University of Twente, 2008.