CNJ: A Visual Programming Environment For Constraint Nets

Fengguang Song and Alan K. Mackworth
Laboratory for Computational Intelligence
Department of Computer Science
University of British Columbia
IRIS ROPAR Project

Introduction

• Constraint Nets is a semantic model for hybrid dynamic systems. Constraint Nets in Java (CNJ) is an environment to support CN modeling, simulation, and 3D animation.
• CNJ is useful for a wide variety of purposes, ranging from intelligent agent systems and real-time embedded systems to integrated hybrid systems.
• It supports hybrid time structures: discrete, continuous, and event-based.
• XML-based Constraint Nets Markup Language (CNML) is defined as the standard interchange format.

Objectives

• Design an interactive tool for CN programming.
• Study component-based technologies to implement control system modeling and simulation.

Constraint Nets

• CN model is an online dataflow-like distributed programming language with a standard graphical representation.
• A constraint net is a triple \( CN = \langle Lc, Td, Cn \rangle \), where \( Lc \) is a finite set of locations, \( Td \) is a finite set of transductions, and \( Cn \) is a finite set of connections between locations.
• The constraint-based agent (CBA) approach applies to artificial agents, also equally to embedded devices, pure software agents and natural animate agents, e.g.,

Class Hierarchy

Conclusions

1. CNJ uses the technologies of component-based software and JavaBeans to successfully implement a practical real-time visual programming environment.
2. The reusability of JavaBeans makes building CN models easier, reducing users time and effort.
3. CNML makes translation to and from other tools feasible and convenient.
4. Development plans include an extension to allow for graphical specification and verification tools.

Acknowledgements

Many thanks go to Yu Zhang (IBM) and Ying Zhang (Xerox PARC). Alan Mackworth is a Canada Research Chair in Artificial Intelligence. Funding support is also provided by IRIS and NSERC.