

Outline		
<ul> <li>Complex Systems and High Performan Simulations</li> <li>Introduction to Modelica</li> <li>Overview of OpenModelica Environme</li> <li>Automatic fine-grained parallelization</li> <li>Explicit model-based parallelization</li> <li>Explicit parallel programming</li> </ul>	nt	
2 Peter Fritzson Parallel Programming Workshop Linköping 2006 June 13	MODELICA	pelab















Modelica – The Next Generation Modeling Language		
<b>Declarative language</b> Equations and mathematical functions allow acausal modeling,		
high level specification, increased correctness		
Multi-domain modeling		
Combine electrical, mechanical, thermodynamic, hydraulic, biological, control, event, real-time, etc		
Everything is a class Strongly typed object-oriented language with a general class concept, Java & MATLAB-like syntax		
Visual component programming Hierarchical system architecture capabilities		
Efficient, non-proprietary		
Efficiency comparable to C; advanced equation compilation, e.g. 300 000 equations, ~150 000 lines on standard PC		
10 Peter Fritzson Parallel Programming Workshop Linköping 2006 June 13	1	







Modelica Conferences	
<ul> <li>The 1<sup>st</sup> International Modelica conference October, 2000</li> </ul>	
<ul> <li>The 2<sup>nd</sup> International Modelica conference March 18- 19, 2002</li> </ul>	
<ul> <li>The 3<sup>rd</sup> International Modelica conference November 5-6, 2003 in Linköping, Sweden</li> </ul>	
<ul> <li>The 4<sup>th</sup> International Modelica conference March 6-7, 2005 in Hamburg, Germany</li> </ul>	
<ul> <li>The 5<sup>th</sup> International Modelica conference September 4-5, 2006 in Vienna, Austria</li> </ul>	
14 Peter Fritzson Parallel Programming Workshop Linköping 2006 June 13	



























































































## Implementation Issues

Spanning tree over all processors of the group •



60

Peter Fritzson Parallel Programming Workshop Linköping 2006 June 13

HODELICA pelab

![](_page_30_Figure_0.jpeg)

	Parallel Prefix Sums in NestStepModelica (1)		
fu i	Inction parPrefix "Compute prefix sums in parallel" nput Integer[:] arr1 ( mem="shared", distr="block" ); putput Integer[size(arr,1)] arr ( mem="shared", distr="block") := arr1;		
pr          	<pre>rotected parameter Integer p = nProcessors(); nteger Ndp = N div p; // Assume p divides N for simplicity nteger[Ndp] prefix; // Local prefix array nteger myPreSum; // Local prefix offset for this processor nteger sum ( mem="shared" ); // Shared, consistent at superstep boundary nteger i, j;</pre>		
algorithm // the parallel code comes here, see next slide end parPrefix;			
62	Peter Fritzson Parallel Programming Workshop Linköping 2006 June 13		

![](_page_31_Figure_0.jpeg)

![](_page_31_Figure_1.jpeg)

![](_page_32_Figure_0.jpeg)

![](_page_32_Figure_1.jpeg)

![](_page_33_Figure_0.jpeg)

![](_page_33_Picture_1.jpeg)