

Specifying Generic Depictions of Language Constructs for 3D Visual Languages

Jan Wolter

University of Paderborn, Germany

VL/HCC, San Jose, CA, USA

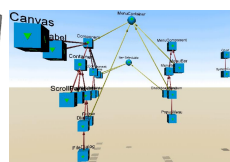
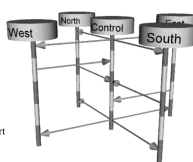
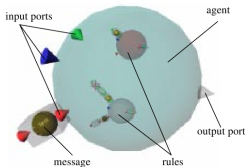
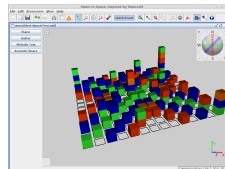
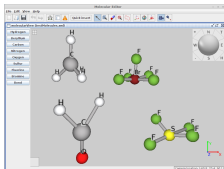
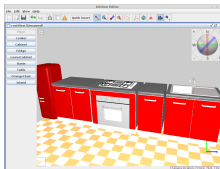
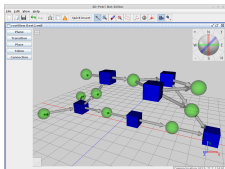
September 18, 2013



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

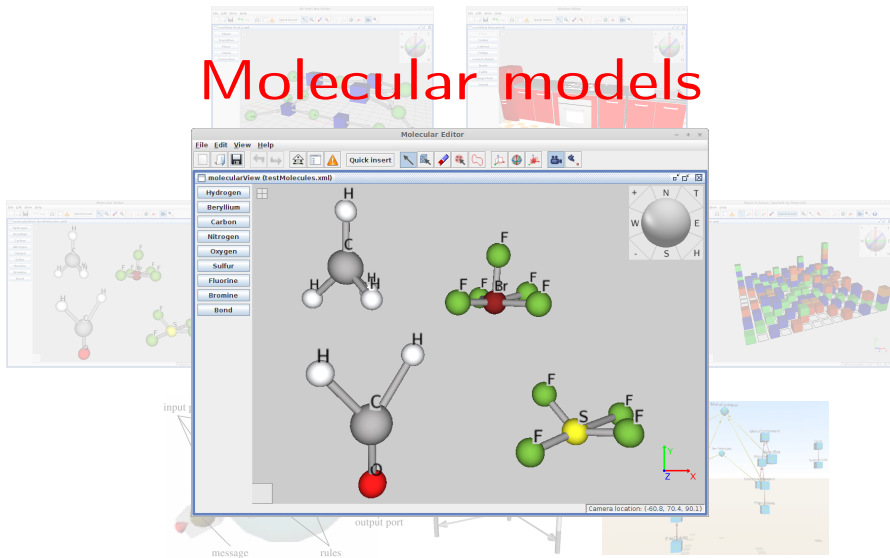


Three-dimensional visual languages



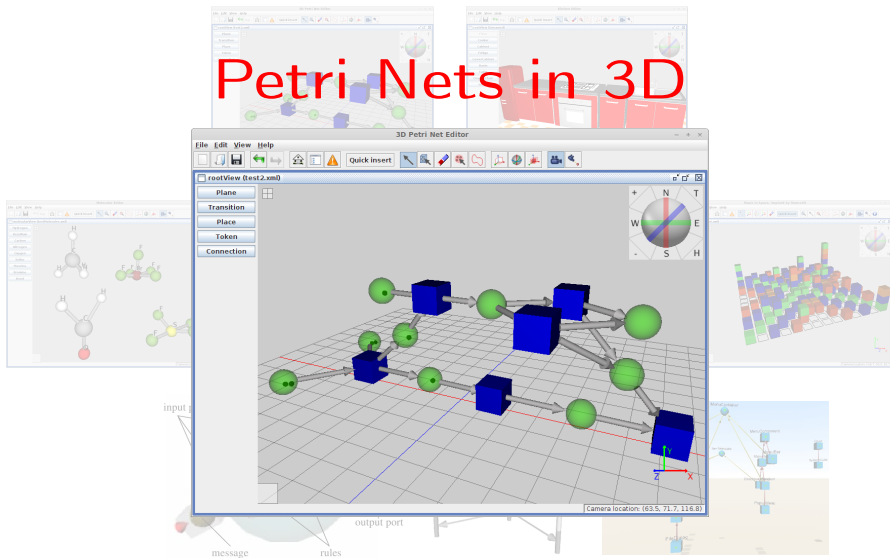
Three-dimensional visual languages

Molecular models

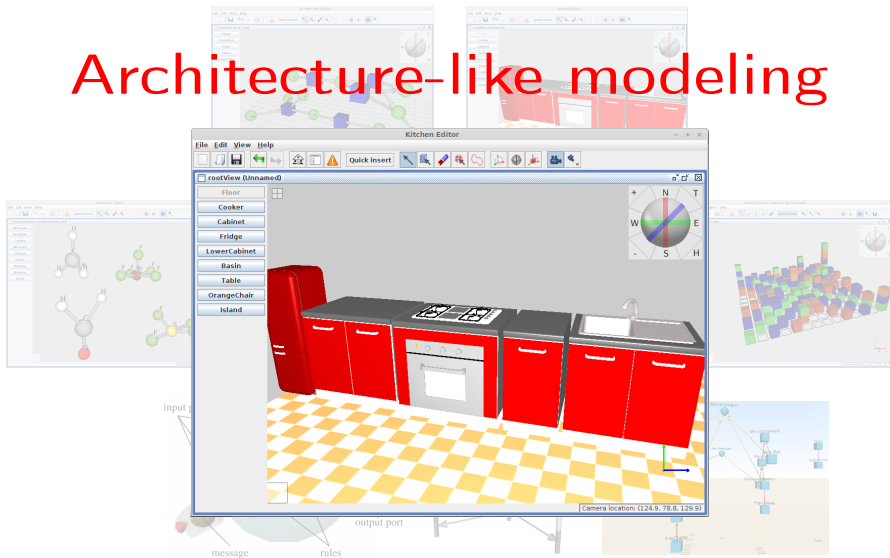


Three-dimensional visual languages

Petri Nets in 3D

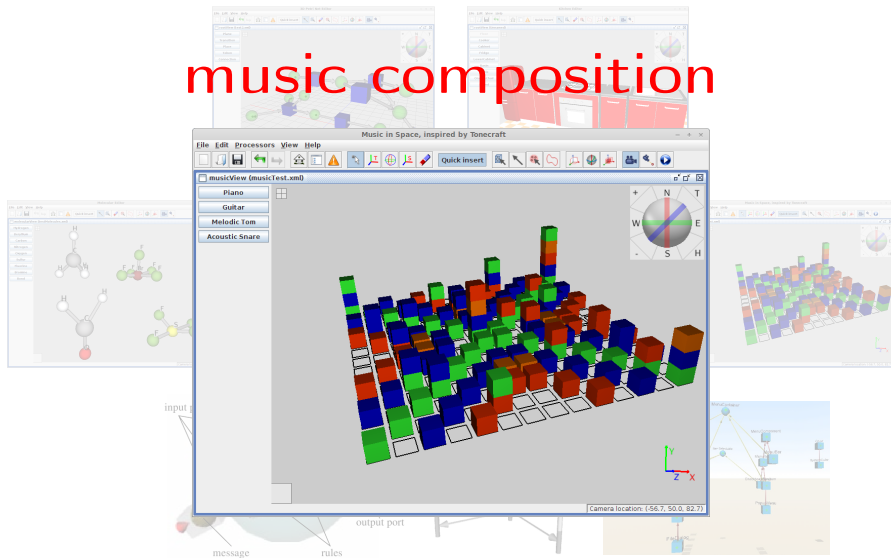


Architecture-like modeling

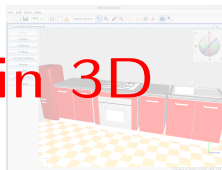
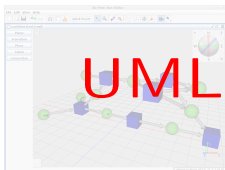


Three-dimensional visual languages

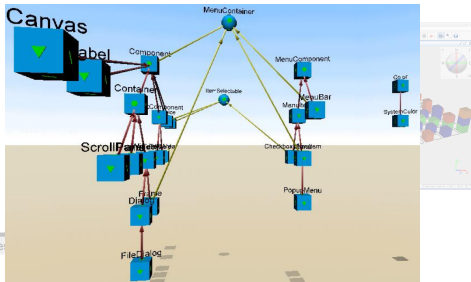
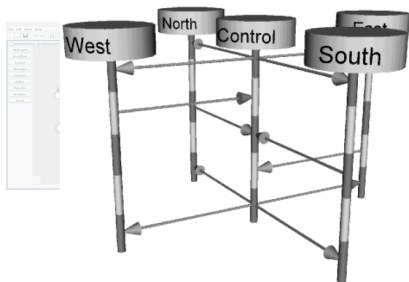
music composition



Three-dimensional visual languages



UML in 3D



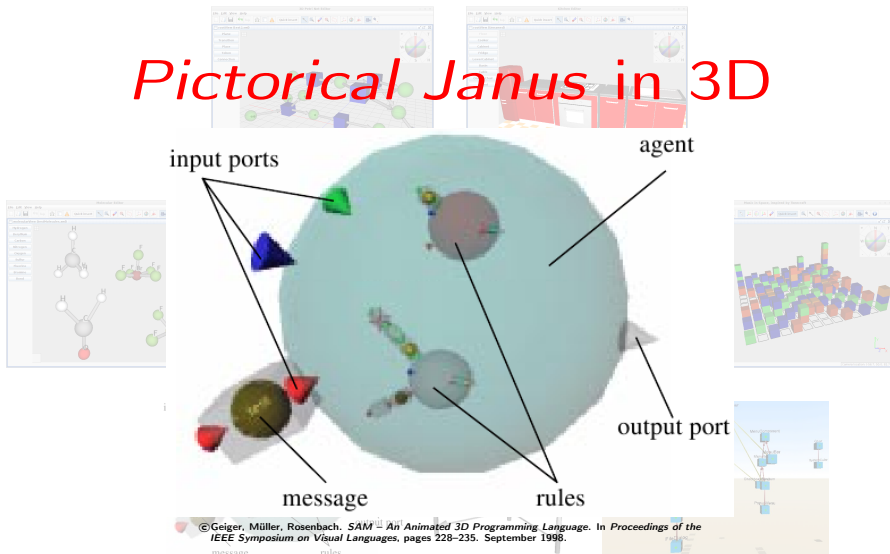
© Gogolla, Radfelder, Richters. Towards Three-Dimensional Representation and Animation of UML Diagrams. In *Proceedings of the International Conference on The Unified Modeling Language: Beyond the Standard*, pages 489–502. September 1999.

© Alfert, Engelen. Experiences In 3-Dimensional Visualization Of Java Class Relations. In *Journal of Integrated Design & Process Science*, 5(3), pages 91–106.



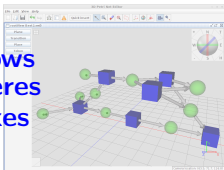
Three-dimensional visual languages

Pictorial Janus in 3D

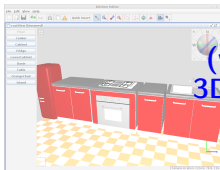


Three-dimensional visual languages

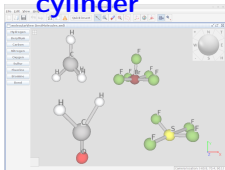
arrows
spheres
boxes



(various)
3D models

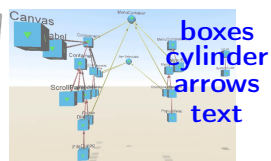
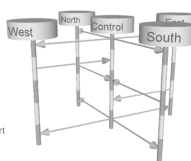
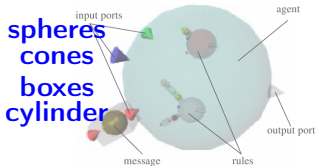
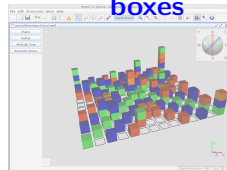


spheres
cylinder

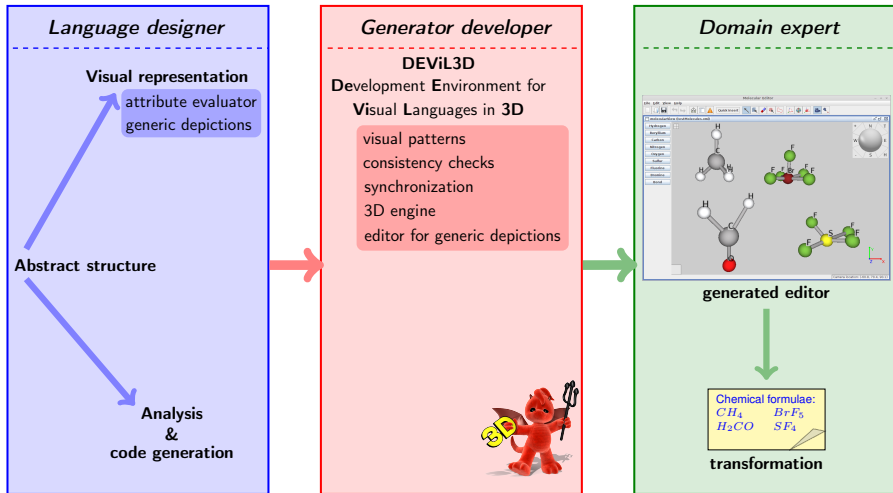


3D programs are
composed of objects
with different
3D shapes

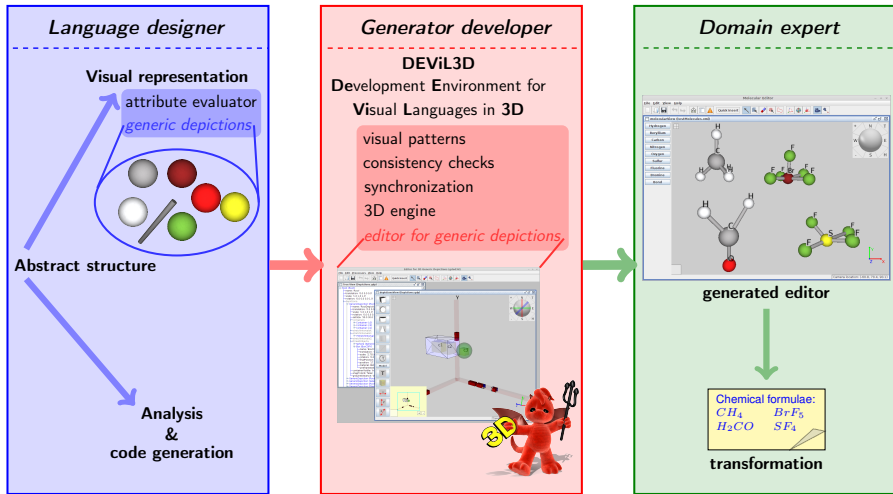
boxes



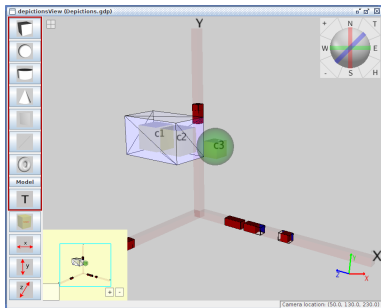
DEViL3D – Generator framework for 3D languages



DEViL3D – Generator framework for 3D languages



Generic Depictions

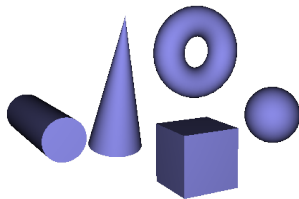


$$D = (\mathcal{P}, \mathcal{R}, \mathcal{C}, \mathcal{I})$$

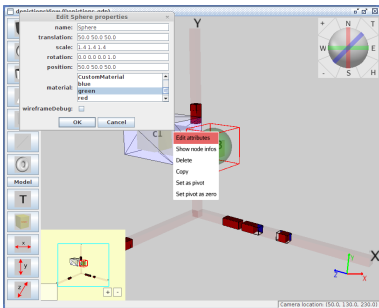
graphical primitives \mathcal{P}

$\mathcal{P} = \text{Box} \dot{\cup} \text{Sphere} \dot{\cup} \text{Cone} \dot{\cup}$
 $\text{Cylinder} \dot{\cup} \text{Arrow} \dot{\cup} \text{Line} \dot{\cup}$
 $\text{Quad} \dot{\cup} \text{Torus} \dot{\cup} \text{3DModel} \dot{\cup} \text{Text}$
representation properties \mathcal{R}
containers \mathcal{C}
stretch intervals \mathcal{I}

- graphical primitives determine the shape(s) of a language construct



Generic Depictions



$$D = (\mathcal{P}, \mathcal{R}, \mathcal{C}, \mathcal{I})$$

graphical primitives \mathcal{P}

$\mathcal{P} = \text{Box} \dot{\cup} \text{Sphere} \dot{\cup} \text{Cone} \dot{\cup}$

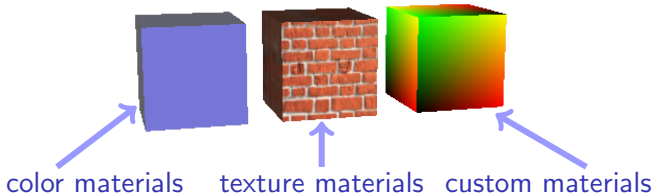
$\text{Cylinder} \dot{\cup} \text{Arrow} \dot{\cup} \text{Line} \dot{\cup}$

$\text{Quad} \dot{\cup} \text{Torus} \dot{\cup} \text{3DModel} \dot{\cup} \text{Text}$

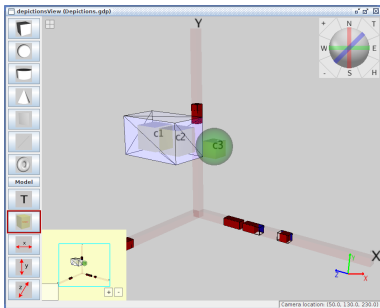
representation properties \mathcal{R}

containers \mathcal{C}

stretch intervals \mathcal{I}



Generic Depictions



$$\mathcal{D} = (\mathcal{P}, \mathcal{R}, \mathcal{C}, \mathcal{I})$$

graphical primitives \mathcal{P}

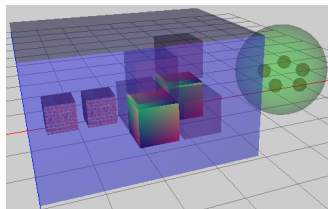
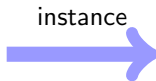
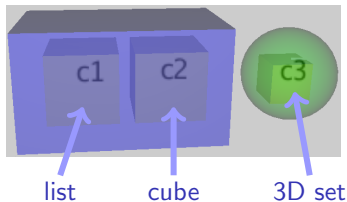
$\mathcal{P} = \text{Box} \dot{\cup} \text{Sphere} \dot{\cup} \text{Cone} \dot{\cup}$
 $\text{Cylinder} \dot{\cup} \text{Arrow} \dot{\cup} \text{Line} \dot{\cup}$
 $\text{Quad} \dot{\cup} \text{Torus} \dot{\cup} \text{3DModel} \dot{\cup} \text{Text}$

representation properties \mathcal{R}

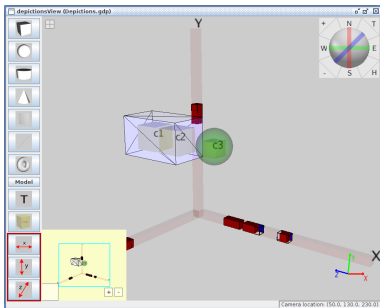
containers \mathcal{C}

stretch intervals \mathcal{I}

- containers are responsible to embed nested constructs
- **visual patterns** define layout of nested constructs



Generic Depictions



$$D = (\mathcal{P}, \mathcal{R}, \mathcal{C}, \mathcal{I})$$

graphical primitives \mathcal{P}

$\mathcal{P} = \text{Box} \dot{\cup} \text{Sphere} \dot{\cup} \text{Cone} \dot{\cup}$
 $\text{Cylinder} \dot{\cup} \text{Arrow} \dot{\cup} \text{Line} \dot{\cup}$

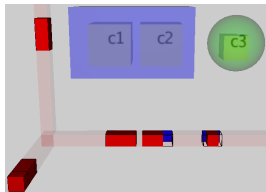
$\text{Quad} \dot{\cup} \text{Torus} \dot{\cup} \text{3DModel} \dot{\cup} \text{Text}$

representation properties \mathcal{R}

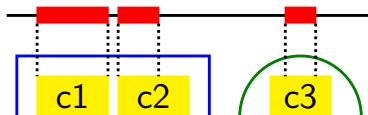
containers \mathcal{C}

stretch intervals \mathcal{I}

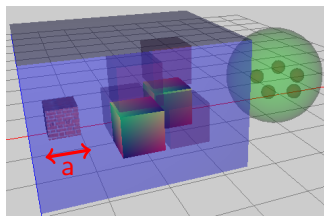
- when the size of nested objects exceeds the container's size, stretch intervals determine which part of a container grows



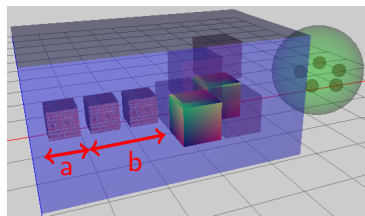
schematic sketch reduced to x-axis:



Stretch algorithm



insert two
list elements
with size b

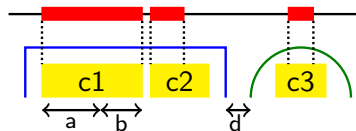
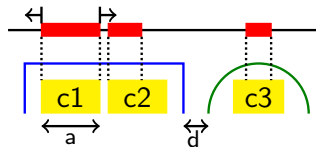


Container c_1

$c_1.\text{actualSize} = a$

$c_1.\text{preferredSize} = a+b$

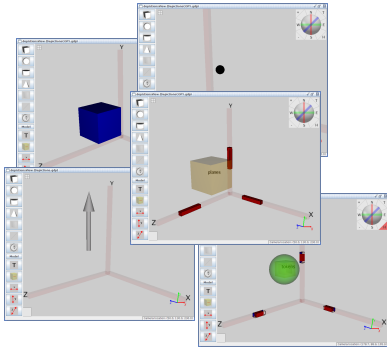
$c_1.\text{actualSize} = a+b$



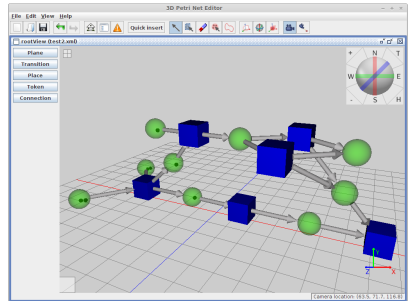
Range of application

generic depictions for **petri net editor**

generic depictions



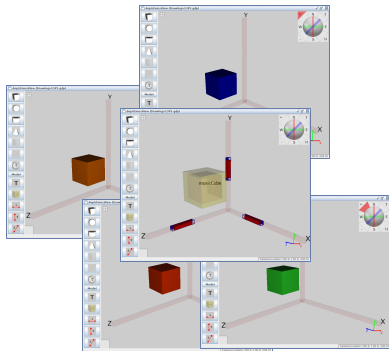
generated editor



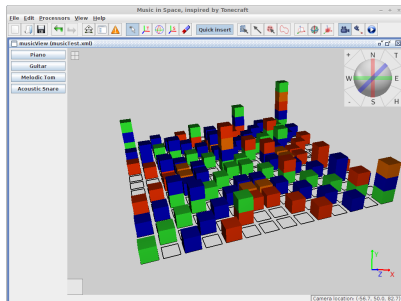
Range of application

generic depictions for music in space editor

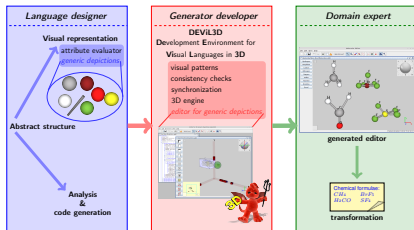
generic depictions



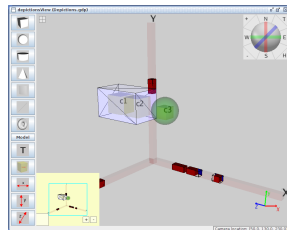
generated editor



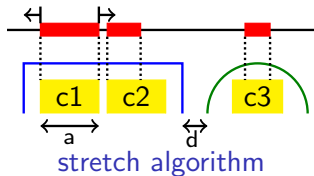
Questions?



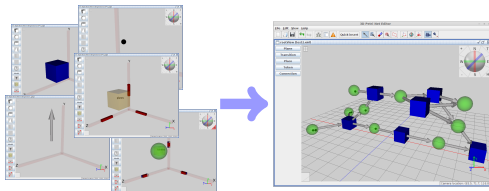
DEVi3D



generic depictions editor



stretch algorithm



application of depictions