

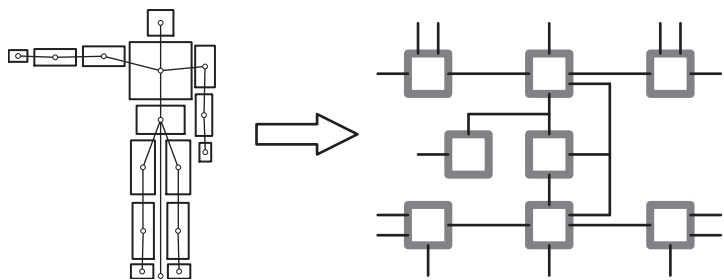
マリオネットと制約

マリオネット：人形劇に登場する糸操り人形

多くのマリオネットは10本に満たない少ない糸で操作できるようになっているが、各パーツの間にある**力学的、運動学的な相互作用**によって、まるで**生きているかのような表情豊かな動作**が可能である。

各パーツをつなぐ関係を制約としてモデル化

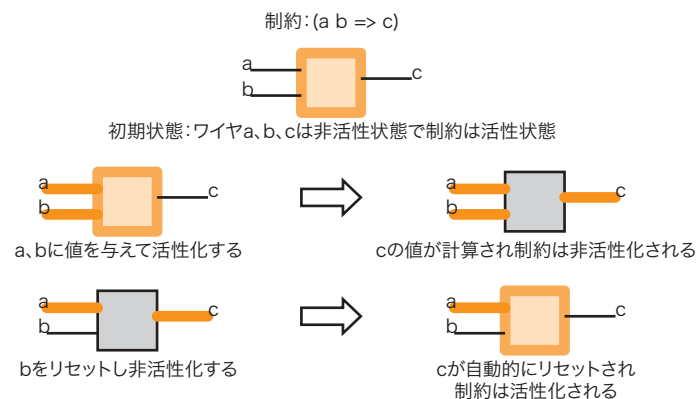
制約システムを使うことで、操作に必要なパラメタを少なく抑えられるため、ゲームのようなインタラクティブな操作環境でも、**簡単な操作で多彩な動作**が実現できると考えられる。



あえて木構造を捨てる

MEPHISTOでは従来の木構造を捨て、全てのノードをフラットに扱い、ノード間の関係は**制約システム**を使って記述する。そのモデルを特徴づける各種の方向ベクトルや、その他の**モデル固有のパラメタ**も含ませることができる。

性能重視の“安直な”制約システム



制約による計算は全て一方向的におこなわれる

制約システムの構成要素のそれぞれについて、**活性、非活性の2種類の状態**を持たせ、活性状態の構成要素のみが計算に使われる。また、一度計算に使われた制約は非活性状態となり、再び活性状態に戻すまでは、計算に使われることがない。

これにより、計算の無限ループに陥ることを原理的に避けることができ、また、**高々制約の個数程度の計算**しかしないことが保証される。

制約としてのアニメーション

モデルの**状態を時刻と結びつける制約**を設定すれば、アニメーションが表現できる。

無限ストリーム ≈ 映画のフィルム

映画のフィルムには、一定の時間間隔ごとに映すべき静止画が焼かれている。この静止画の1枚1枚を**フレーム**と呼ぶ。

MEPHISTOでは、アニメーションをフレームの無限**ストリーム**で表現する。無限ストリームは**遅延評価**の機構を使って実現するが、概念的には**無限の長さを持つリスト**ととらえることができる。

ストリームの中身：描画手続き、デバイスやアプリケーションの状態、etc...

各フレームで描画手続きを実行することで、画面に図形が描画され、映像が作られる。全ての図形要素は手続きとして統一的に表現する。この手続きは**状態を持つクロージャ**であってよい。

アプリケーションや入力デバイスの状態もストリームとして表現すれば、**アプリケーションの性質や実行環境に依存した部分を隠蔽**することができ、より効率的なアプリケーションの開発ができる。

