プログラム実行状況図形表示におけるリスト構造レイアウト方式

6R-8

高橋健司 下村隆夫 磯田定宏

NTTソフトウェア研究所

1．はじめに

プログラム実行状況の図形表示可能なビジュアルデバッガＶＩＰＳ[1]における，リスト型データの図形表示方式について報告する．

大量のデータを効率良く表示するためには，①視認性が高く，表示スペースの小さいリストレイアウト方式，②注目する部分だけの選択的な表示方式，が必要である．

2．木構造のレイアウト方式の比較

リスト型データを木形式にレイアウトする方式を視認性および表示面積の観点から比較する．木構造のレイアウト方式として，リーフ等間隔方式，最密・親ノード中心配置方式，レイヤ等分割方式の３方式を選択し比較する．ただし，表示密度の比較では，同一レイヤ間の最小単位およびレイヤ間の距離を一定とする．

リーフ等間隔方式

全レイヤの葉ノード間の距離が最小単位となるように配置する．親ノードは，子ノードの中心に配置する（図１）．

- 視認性
  視認性が高い．ノードの親子関係を直感的に把握できる．
- 表示密度
  表示密度は３方式の中でもっとも低い．

最密・親ノード中心配置方式

親ノードが子ノードの中心にあるという条件の下で表示密度最大となるように配置する（図２）．

- 視認性
  リーフ等間隔方式と同様に，視認性が高い．
- 表示密度
  リーフ等間隔方式より高く，レイヤ等分割方式より低い．

レイヤ等分割方式

ノード数最大レイヤにおいて，ノードを最小間隔で配置する．他のレイヤでは，ノード数最大レイヤの幅を等分して，各ノードを配置する（図３）．

- 視認性
  視認性が低い．親ノードが必ずしも子ノードの中心にないため，ノード間の親子関係が直感的に把握できない．
- 表示密度
  ３方式の中でもっとも表示密度が高い．

デバッグ作業におけるリスト型データの表示では，限られた画面上により多くの情報を，いかに把握しやすい形式で表現するかが重要である．したがって，最密・親ノード中心配置方式がもっとも適している．

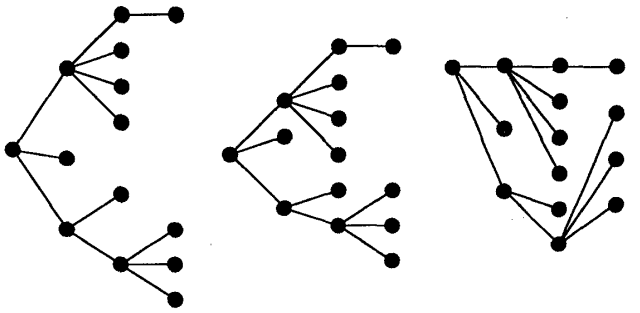


図1 リーフ等間隔方式　図2 最密・親ノード中心配置方式　図3 レイヤ等分割方式

3．ＶＩＰＳにおけるリスト型データ表示方式

ＶＩＰＳでは，①全体／部分表示方式，②選択表示方式により，大量のリスト型データの中から注目したい部分を効率良く表示することができる．

全体／部分表示方式

全体表示と部分表示の２つの形式でリスト型データを表示する[2]．

全体表示では，リスト型データのマクロ構造をユーザ指定のルートノードを原点とする木形式で表示する．木構造のレイアウトでは，２.の最密・親ノード中心配置方式を用いる．

部分表示では，データの一部分を詳細に表示する．ユーザは，マクロ構造上でノードを指定することにより，そのノードを始点とした部分を詳細に表示できる（図４）．

全体／部分表示方式により，注目したい部分の選択および表示を容易化する．

選択表示方式

ＶＩＰＳでは，部分表示において，複数の要素から構成されているノードを矩形の組合せで表現する．各ノードの中で注目したい要素については，選択的に表示できる．選択表示方式には，以下がある（図５）．

Visualization Method for List-Type Data in a Visual Debugger,VIPS
Kenji Takahashi, Takao Shimomura, Sadahiro Isoda
NTT Software Laboratories

● 選択表示

各ノードの中の注目したい要素を指定して，表示する．選択する要素は，各ノード個別に指定する以外に，同じ構造を持つ全てのノードについて，選択表示する要素を一括して指定することもできる．

● ポインタ表示制御

各ノード間のポインタによるリンク状態を全ポインタ表示／限定ポインタ表示モードの２つの形式で表示する．

全ポインタ表示モードでは，リンクされているノードを全て表示する．

限定ポインタ表示モードでは，各ノードのポインタ要素のうち選択表示されているものについてのみリンクされているノードを表示する．

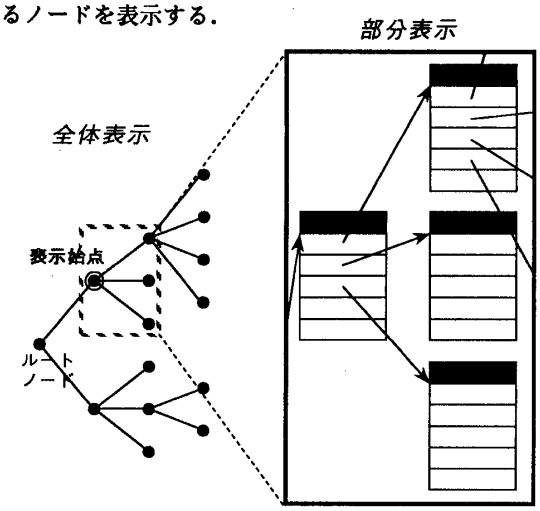
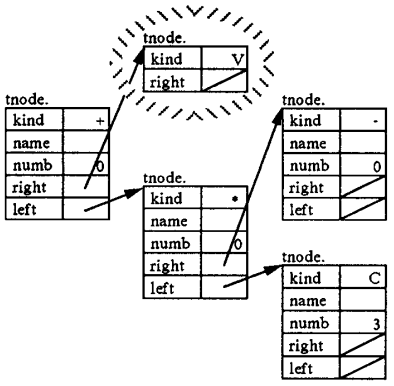


図４．リスト型データの全体／部分表示



全ポインタ表示モード　　限定ポインタ表示モード

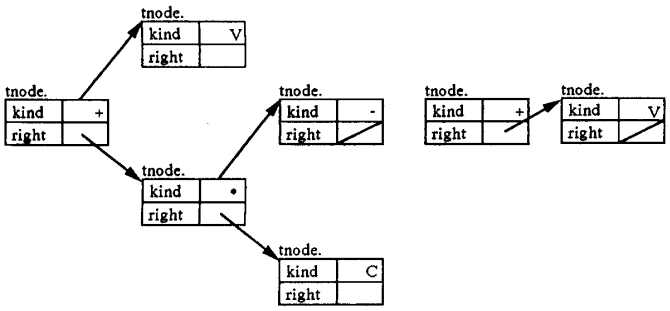


図５．選択表示

４．レイアウトアルゴリズム

リスト構造全体は，最密・親ノード中心配置方式に基づき全体表示エリア上でレイアウトする．

部分表示では，まず全体表示エリア上で部分表示する範囲を決定する．つぎに，決定した範囲を部分表示エリアのスケールに換算して表示する．

レイアウトアルゴリズムを以下に示す（図６）．

木構造のレイアウトアルゴリズム

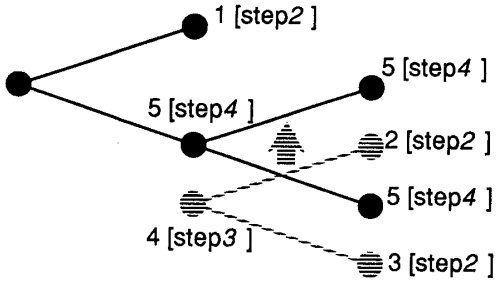
step１　depth-firstの帰りがけ順で各ノードを配置していく．

step２　リーフ・ノードは，配置済みの全てのノードよりも１単位下に配置する．

step３　親ノードを子ノードの中心に配置する．

step４　親ノードをルートとする部分木のいずれかのノードと，配置済みのノードとの距離が最小単位となるまで，上方に移動する．

step５　step１，２，３，４を木全体が完成するまで繰り返す．



凡例：配置の順番[４章のレイアウトアルゴリズムのstep番号]

図6. 木構造レイアウトアルゴリズム

５．おわりに

プログラム実行状況の図形表示可能なビジュアルデバッガＶＩＰＳにおける，リスト型データを表示する方式について報告した．本方式により，大量のリスト型データの中から注目したい部分を効率良く表示することができる．

今後，本方式を計算機上で実現し，有効性を評価する．また，表示アルゴリズムの高速化についても検討していきたい．

［参考文献］

[1]S.Isoda, T.Shimomura, and Y.Ono, "VIPS: A Visual Debugger", IEEE SOFTWARE, Vol.4, No.3, May 1987, pp.8-19.

[2]高橋，下村，森下，磯田，"既存ﾃﾞﾊﾞｯｸﾞﾂｰﾙを利用した高速なﾋﾞｼﾞｭｱﾙﾃﾞﾊﾞｯｸﾞ方式"，情報処理学会第38回全国大会予稿集，2M-3，March 1989，pp.1235-1236.