

大規模アプリケーション向けハイパーメディアシステム (2)

- プレゼンテーション制御方式 -

3T-2

友納 正裕 杉淵 史子[†] 高野 元 元木 誠[‡] 中島 震
NEC C&C 研究所 NEC 情報システムズ[†] NEC 第二製造業 SI 事業部[‡]

1 はじめに

本稿では、大規模アプリケーションを考慮したハイパーメディアシステム [1] におけるプレゼンテーション制御部について説明する。

本システムでは、大規模データに対応するため、ノードリンクおよびメディアをデータベースで管理し、スキーマに基づいて AP のプロトタイピングを行いながら、順次データを追加して大規模化していくという AP 作成を可能としている。この際、プレゼンテーション制御部は、ノードリンクと GUI 間の対応づけを行い、ナビゲーション制御とシナリオ実行制御を司る。次章以降、プレゼンテーション制御部の要件とその実現方式を述べる。

2 プレゼンテーション方式の概要

2.1 構成要素

プレゼンテーション制御部の構成要素として、以下のものがある。

- パート
動画や静止画などのメディアを表示するカード類や、ボタンやリストボックスなどのコントロール部品からなる。
- プレゼンタ
パートを包含するフレームウィンドウである。
- シナリオ
プレゼンタやパートのレイアウトや動作を記述する。

プレゼンテーション制御部は、実行時にシナリオを読み込み、プレゼンタやパートを生成し、その後、ユーザ入力にしたがってナビゲーションを行ったり、シナリオ記述に基づいて自動ナビゲーションを行ったりする。

2.2 スキーマとの対応

ノードリンク DB 部では、スキーマの導入により、大量のノードリンクデータを組織的に登録・管理できるようにしている。このため、プレゼンテーション制御部

では、スキーマに沿ってノードを表示する機構が必要になる。

この機構として、プレゼンタやパートにノード格納用のスロットを導入する。スロット構成とスキーマを対応づけることにより、スロットに割り当てるノードの型¹が決まる。図 1 を例に、この仕組みを説明する。図 1 では集合ノード S1 ~ S3、T1 ~ T2 およびその間の太いリンクがスキーマである。

1. スロット
プレゼンタやパートはノードを格納するためのスロットを持つ。たとえば、図 1 で、プレゼンタ 1 の W、パート 1 の X などがスロットである。パートはスロットに割り当てられたノードの内容を表示できる。
2. スキーマとの対応
プレゼンタおよびパートのスロットをスキーマに対応づける。各スロットには、スキーマ中の型が対応し、その型に属するノードが割り当てられる。図 1 では、たとえばスロット X には集合ノード S1 が型として対応し、その要素ノード a1 ~ a3 が X に割り当てられる。

3 ナビゲーション制御

3.1 遷移先プレゼンタの選択

本システムでは、プレゼンテーション制御部とノードリンク DB 部が分離しているため、プレゼンタとノードの対応づけが必要になる。すなわち、WWW など従来の多くのハイパーメディアでは、ノードにプレゼンテーション記述が付随しているため、その記述にしたがってノードをブラウザに表示すればよかったのに対して、本モデルでは、ノードをどのプレゼンタやパートで表示するかを決定する必要がある。

これに対処するために、遷移先プレゼンタ表を導入する。遷移先プレゼンタ表は、遷移先ノードが属する型とそれを表示すべきプレゼンタを対応づける。ナビゲーションの際には、プレゼンテーション制御部が、ポイントされたホットスポットに対応づけられたリンクをたどって遷移先ノードを取得し、遷移先プレゼンタ表を検索して、遷移先ノードに対応したプレゼンタを求める。遷移先ノードの型として、集合ノードおよびノード属性

¹型の実現としては、集合ノードを利用する方法とノード属性を用いる方法があるが [1]、ここでは集合ノードの場合で説明する。

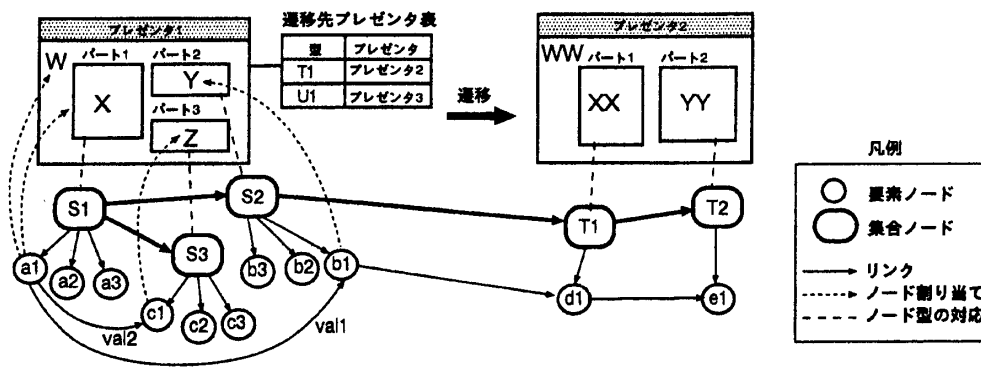


図 1: ナビゲーション方式

を使うことができる。遷移先プレゼンタ表には、スキーマの遷移構造に従わないノード型を登録することもできるので、アドホックなリンクを張りたい場合にも対応することができる。

3.2 ノード割り当て

遷移先プレゼンタが決定した後、各パートへのノード割り当ては、以下のように行う。

1. ヘッドノードの設定
ヘッドノードとは、プレゼンタのスロットに割り当てられるノードである。図1では、WやWWがヘッドノードである。
2. パートへのノードの設定
ヘッドノードを基準にして、パートに設定すべきノードを取得する。これらのノードは、ヘッドノードからリンクをたどる検索命令により取得する。この検索命令はシナリオに記述する。なお、ヘッドノードを基準にしない一般の検索式を記述することも可能である。

図1のプレゼンタ1に関するシナリオ例を以下に示す。行7のSet文でパート1にプレゼンタ1のヘッドノード(\$selfで指定)を設定している。行8では、ヘッドノードから出るリンクの中で属性atrの値がval1であるものをたどって得られるノードをパート2に設定している。

割り当てられたノードの内容は、次章で述べるシナリオ実行制御により画面に表示される。

```

1 Presenter プレゼンタ1
2 {
3   Part   パート1 ; # 動画カード
4   Part   パート2 ; # 静止画カード
5   Part   パート3 ; # 静止画カード
6
7   Set   パート1 $self;
8   Set   パート2 GetNodeByLinkAttr $self atr val1;
9   Set   パート3 GetNodeByLinkAttr $self atr val2;
10 }

```

4 シナリオ実行制御

パートに設定されたノードを表示する際、そのタイミングを指定することができる。たとえば、「パート1の(ノードの)表示が終了したらパート2を表示する」といった指定ができる。一般に、「On イベント Then アクション」というように、実行したいアクションをイベント駆動で記述する。

これにより、メディア表示順序の柔軟な指定や、イベントによりボタンをプッシュして自動的にナビゲーションを行うといったことをシナリオに記述することができ、魅力的なAPの作成が可能になる。

5 まとめ

大量データからなるハイパーメディアAP作成に適したプレゼンテーション制御方式について述べた。この方式は、一定のノードリンクデータに対してプレゼンテーションを種々変更したい場合や、一定のプレゼンテーションに対して時々刻々ノードリンクデータが追加されるようなAPに対して特に有効である。また、実装においては、オブジェクト指向フレームワークとして拡張性に富むように設計している[2]。シナリオ記述を支援するツールの開発が今後の課題である。

参考文献

- [1] 高野、他、「大規模アプリケーション向けハイパーメディアシステム(1)、(3)、(4)」、第53回情報処理学会、1996。
- [2] 友納、中島、「ハイパーメディア向けオブジェクト指向フレームワークの構築」、情報処理学会 00 シンポジウム'96, pp.175-182, 1996。