

推論による情報漏えい防止のための オブジェクト関係の視覚化

木下研究室

鈴木 遼 (200602840)

1 はじめに

現在、今まで行われてきた情報漏えいの対策というのは、簡単に言うなら秘密情報がそのまま丸ごと本来情報を得ることが出来ない利用者に流出することを防ぐのを目的としている。だが、ネットワークの拡大などに伴い、一つ一つの情報がそれぞれ秘密情報として扱う必要がなくともその情報が複数集まることにより推論によって、本来は得ることが出来ない秘密情報を利用者に流出してしまう可能性が出てきた。このことから、どのような情報が複数集まることによって、こういった情報が推論によって得られるかを考慮する必要性が発生した。

そこで本研究では、情報（オブジェクト）同士の関係を考慮するためのデータ構造を提案するまた、それを視覚化することで、どの情報からどの情報が推論できるかを容易に把握することにより想定外の推論による情報漏えいを防ぐためのツールを実装することを目的とする。

2 オブジェクト間の構造

どのような情報が集まることによって、どのような情報が推論によって得られるかをグラフ構造を用いて表現する。これにより、オブジェクト間の関係から通常得られる情報と推論により得ることが出来る情報との関係を視覚化し易くすることが可能である。

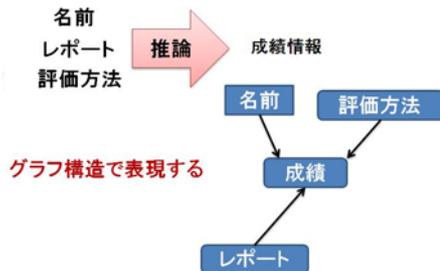


図 1: グラフ構造の例

図 1 は名前、評価方法、レポートの 3 つのから成績情報が推論によって分かるというオブジェクト間の関係をグラフ構造として表現している。

3 オブジェクト間の視覚化

オブジェクト間の関係を視覚化する際に問題になる項目を以下に挙げる。

- オブジェクト間の関係を視覚化する。
- 秘密としておきたいオブジェクトを設定することができる。

これに対して以下のシステムを提案する。

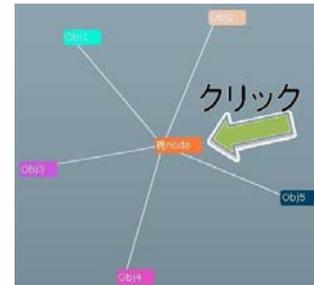


図 2: 視覚化の例 1 クリック前



図 3: 視覚化の例 2 クリック後

例えば、上の図 2 では中心にあるオブジェクト情報を秘密情報としておきたいとき、中心にあるオブジェクトの情報をクリックした時、どのオブジェクトから中心のオブジェクトが推論されて情報が漏れてしまうかを教えてくれる。このときにシステム全体のプログラムは Adobe Flex3 を使用。また、オブジェクト全体の情報を読み込む際に使用するファイルの形式は XML を使用する。

下に載せるのはそのプログラムの一部です。

```
public var graph:Graph = new Graph();
public var abc:UIComponent = new UIComponent();
public function sample(event:Event):void{
    addChild(abc); abc.addChild(graph);}
var a1:Node = graph.createNode(250,250,{text:member.main});
var s1:Node = graph.createNode(100+Math.floor(Math.random()*100),
100+Math.floor(Math.random()*100),{text:"利用者"});
```

4 まとめ

本編では、オブジェクト間の関係を視覚化し簡単な推論が行えるシステムを提案した。だが、推論するための情報や、オブジェクト間の関係などは予め与えられていることを前提として行われている。

今後の課題としては、推論規則をオブジェクトとしてとらえた場合の処理の方法や、オブジェクトの関係を外部のファイルとして入力できるようにする。