M-52

階層的作業空間モデルにおける セキュリティー保持のための制約

Constraints for securities in a hierarchical workspace model

村上隆治 Takaharu Murakami
横田裕介 Yusuke Yokota
上林彌彦 Yahiko Kambayashi

1．はじめに

協調作業は役割が様々に異なる人同士で行うことがよくあり、我々はそうした作業を支援できるシステムを開発している。役割に対応した作業空間を定義し、作業空間ごとにユーザの権限を管理する。また作業空間は木構造を構成しそれぞれの関連を示している。それぞれの作業空間では参照できる資料が指定される。各環境では親環境で加えられたカスタマイズに加え、更に独自のカスタマイズを行うことができ、利用者の役割に応じたビューを提供する。図1は企画部と開発部を有するプロジェクトにおける企画資料の利用の例を示しており、各部で独自のカスタマイズを加えて利用している。

このシステムでは各作業空間で権限の管理を行うため、共有資料の利用者に対するセキュリティーを保持する役割も担っていることになる。我々はこの作業空間を環境と呼び、環境ごとの設定を XML 文書で表現した。しかし各環境の設定が自由に変更できると、ひとつの要素の変更によりシステムのモデル設定に支障を生じる場合がある。これは共有資料に対するセキュリティーの立場から見ても望ましいものではない。こうした不都合を避けるためには環境の設定要素に対してなんらかの制約を加えることが必要となる。本稿では環境の設定を変更する時に定められるべき制約条件について示す。

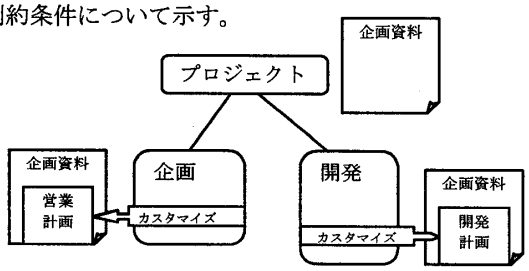

図1：システムの概要

2．作業空間ごとに管理される情報

各環境の情報は図2に示したような XML 文書によって環境ごとに管理されている。その内容について簡単に説明する。まずその環境の名前が示されている(env)。その次にその環境を利用できるユーザ名のリストが示される(usrs)。そして manager の欄に示されたユーザ名はその環境の管理者であり、この環境 XML 文書自体を書き換える権限を持つ。その次の parent、children はそれぞれこの環境の親環境、子環境を示す。最後の docs にこの環境で参照できる資料が並ぶ。それぞれの資料の欄にはこの環境独自にその資料に対して行なったカスタマイズの内容を記した文書が記述されている。これらの文書にはそれを書き換える権限を持つユーザ名が記される。

```
<?xml version="1.0" encoding="shift_jis"?>
<env name="envC">
<usrs><usr name="User1" />
        <usr name="User2" />
</usrs>
<manager name="User1" />
<parent env="envA" />
<children><child env="envD" />
          <child env="envE" />
</children>
<docs>
  <doc id="document1" top="true" leaf="false">
    <style-lim sheet="lim001.xsl">
      <writer name="User1" />
    </style-lim>
    <style-add sheet="add001.xsl">
      <writer name="User1" />
      <writer name="User2" />
    </style-add>
  </doc>
</docs>
</env>
```

図2：環境 XML 文書の例

ここでは資料のカスタマイズの内容を2つの文書に分けて管理している。これは上位の環境におけるその文書の情報の中でも、その環境の人に見せないようにしたいものがある場合にそれらの情報を削除するための操作と、それに対して新たに何らかの情報を追加するための操作ではそれを行なうことのできる人に違いをつけるべきであるとの考えによるものである。それぞれを制限カスタマイズ(style-lim で指定)と、追加カスタマイズ(style-add で指定)と呼ぶことにする。

また、各々の文書には top 属性と leaf 属性が付けられている。まず top 属性はその文書がこの環境の親環境から継承されたものであるかどうかを示す。この属性が false であればそれは親環境でも参照されていることを示す。この場合には先祖の環境でこの資料に対してカスタマイズが加えられている可能性があることを示す。もし true であればこの資料は親環境では参照されていない。次に leaf 属性については、それが false であればこの資料がこの環境の子環境でも参照できるようにすることを許可していることを示す。この属性が true であればこの資料はこの環境の子環境では参照することができない。

このような環境 XML 文書の項目それぞれの取りうる値には、どのような制約が必要であるかについて次節でまとめる。

3．作業空間に対する制約

管理者の権限：

京都大学大学院情報学研究科

まず他の環境と関係なく、1 つの環境 XML 文書の中で成り立つべき制約を考える。環境の管理者は、その環境の環境 XML 文書自体を書き換える権限をもつ人である。環境の管理者にはその環境で参照可能な資料は一般のユーザと同じように利用できる権限があるということである。ゆえに管理者はユーザの一員である。

カスタマイズ実行者：

また資料に対する追加カスタマイズを行なう際には、そのカスタマイズの結果が参照できることは重要である。つまり追加カスタマイズを行なう人に必要な権限はその環境のユーザの権限をみたしているということになる。

一方制限カスタマイズについては、親環境におけるその資料の内容に対してその環境のユーザに見せたくない内容を削除するといった意図がある。制限カスタマイズが文書の内容に対するセキュリティーの意味を持っていることを考えれば、このため制限カスタマイズを行なう権限を持つユーザはむしろその環境の親環境のユーザにあるべきである。

資料の継承に対するセキュリティー：

資料の内容を子環境でも継承して利用できるというモデルにおいても、その内容の一部には子孫の環境では見せたくない情報もある。このような情報に対してセキュリティーを保持するためには、環境を新たに作りだす場合や、環境 XML 文書を書き換える際にある程度の制約を考える必要がある。

セキュリティー保持の目的としては、それぞれの環境で参照出来る資料の選択にもある程度の制約を設定する必要がある。上位の環境で参照されており、そのいずれかの環境で継承を拒否する設定が行なわれている（leaf 属性に false が設定されている）ような資料を考える。階層構造のひとつのパス上にある環境どうしには何らかの関連があると考えられる。それゆえ子孫の環境においてはその資料を参照できないと定めても何ら不自然なことは無いだろう。

しかしこの制約によりある時点では参照可能であったはずの資料が、上位環境の環境 XML 文書の変更によって参照できない資料になるといったこともある。この際には影響を受ける各環境 XML 文書を自動的に変更する、あるいは変更を各環境の管理者に指示するといったことが必要となる。資料に対する参照権限が変化することを考えれば、影響が及ぶ環境に対する通知や環境 XML 文書の変更は、その原因となる上位環境における変更が生じたときに行なわれるべきであるだろう。この変更を簡単に行なうためにも各環境において参照が許可されない共有資料のリストが得られることはセキュリティー管理の観点から重要である。

以上のように環境の情報についての制約については、それぞれの環境の設定を変更する際に参照され、変更や設定が行なわれるたびにその値の妥当性を検証することが必要となる。そのためこれらの制約はなんらかの形式で記述され、それぞれの環境で保持しておくのが妥当であると考えてよいだろう。そのためにはこの制約の記述を図 1 に示したような環境 XML 文書に加える必要がある。この資料の参照制約は、環境の階層構造に従ってルート環境から順に leaf 属性が true となる文書をリストアップして行くことによって得られる。

4．環境作成時の制約例

ある環境 A に対して、その環境の子環境 B を新たに生成する場合を例にとり、どのような制約が参照されるかについての具体例を以下に示す。

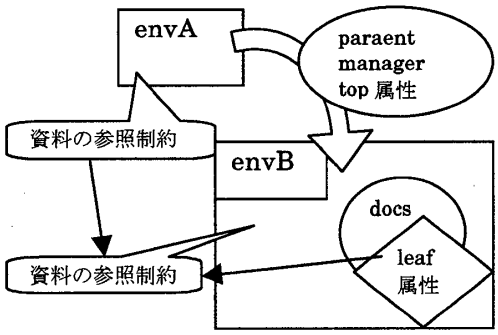

図 3 ：制約参照の例

まず新たに作成された環境 B の環境 XML 文書では、その親環境 A の環境 XML の値等により必然的に親環境、子環境の名前が定まる。環境 B の管理者は親環境である環境 A のユーザの中から選ばれる。参照できる資料の設定においては、親環境 A で設定された資料の参照制約に無いものの中で設定できる。これらの資料に対する属性は、top 属性については環境 A でその資料が参照されているかどうかで決まり、leaf 属性は任意に設定できる。その環境 B で指定する資料の参照制約を指定する際には、親環境 A で指定されている資料の参照制約を元にして環境 B で指定する参照制約を作成する。具体的には環境 B で参照されている資料のうちで leaf 属性が true であるものについてその資料を資料の参照制約リストに加えることになる。環境 B で参照できる資料の追加カスタマイズを行う権限を持つ人は環境 B のユーザでなければならず、制約カスタマイズを行う権限をもつユーザは親環境 A のユーザから指定される。

5．　まとめ

我々の開発している協調作業支援システムでは作業空間のモデルに階層構造を導入している。この場合共有資料に対するセキュリティーを考えるためには作業空間の上下関係を考慮した制約の設定が必要となる。本稿ではこの作業空間のモデルにおいて、システムで共有される資料に対するセキュリティーを保持するために必要となる制約について述べた。今後、システムに必要な制約のシステムを更に深く考慮した上で導入していきたい。こうした制約を考慮した上で環境の情報の更新が可能なユーザインタフェースの作成も重要である。

参考文献

[1] M. Tanigaki, Y. Yokota, Y. Kambayashi, An XML-based Work Space and Document Model for Flexible Cooperative Work, Proc. the IASTED International Conference APPLIED INFORMATICS, 2002, pp.431-436