AssureNote: Wiki 記法ベースの GSN オーサリングツール

松村 哲郎† 志田 駿介† 倉光 君郎‡

†横浜国立大学大学院 〒240-8501 神奈川県横浜市保土ヶ谷区常盤台 79-1

†横浜国立大学 〒240-8501 神奈川県横浜市保土ヶ谷区常盤台 79-1

E-mail: †{matsumura.t.lab, shunsuke.sida, kkuramitsu}@gmail.com

あらまし 本論文では，Wiki 記法ベースの GSN エディタ AssureNote の提案を行う．GSN は，Assurance Cases のビジュアル表記法として標準化が進み，オーサリングツールが開発されている．しかし，従来のオーサリングツールは，図形エディタをベースにしており，エディタを利用出来ない環境では Assurance Cases の再利用性がえられない．我々は，Wiki 記法ベースの表記法を提案し，データ再利用性とグラフィカル編集の両立を実現するツールを開発した．

キーワード Wiki，Assurance Cases，GSN

1. はじめに

本論文では，Assurance Cases の代表的なビジュアライズ方である GSN の，Wiki 記法ベースの編集法を採用した編集ツールである AssureNote を提案する．Assurance Cases はシステムの安全性やディペンダビリティ要求を議論し，システムのステークホルダ間で合意を形成するための技術文章である．GSN は，Assurance Cases の代表的なビジュアル記法であり，GSN を用いて Assurance Cases を記述するための GSN オーサリングツールの開発が行われている．しかし，従来のオーサリングツールは Eclipse の図形描画プラグインを利用するなど，図形エディタをベースとして開発されており，Assurance Cases のデータ再利用性がえられない．例えば，こうした GSN オーサリングツールではツール上のユーザインターフェースを介した編集のみを想定しており，保存形式である SACM (Structured Assurance Case Metamodel) や ARM (Argument Meta Model)を直接編集することは難しい．

そこで我々は，Wiki 記法ベースの表記法を採用し，データの再利用性とグラフィカル編集の両立を実現する GSN オーサリングツールである AssureNote を提案する．AssureNote は WGSN (Wiki-Style GSN)のエディタを GSN 編集のインターフェースとして採用している．WGSN はテキストベースの統一的な GSN 編集のために設計された表記法であるが，変数の定義やノードの参照など，GSN の記述に十分な機能を備えている．WGSN の記法を用いることで，木構造で表される GSN を，AssureNote 上のテキストエリア上から編集することが可能となり，複雑なユーザインターフェースを介さない GSN の編集が実現される．

本論文の構成は以下の通りである．第 2 章では GSN の説明を行い，第 3 章では関連研究について説明する．第 4 章では AssureNote の設計について述べ，第 5 章では WGSN について説明する．第 6 章では実装を述べる．第 7 章では AssureNote を使った実証実験を行う．最後に，第 8 章で本論文を総括する．

2. GSN (Goal Structuring Notation)

GSN の表記法の抜粋を図 1 に示す．GSN は主に Goal, Strategy, Context, Evidence の 4 種類のノードから成る木構造の表記法である．それぞれの意味と役割は以下の通りである．

- Goal: 安全性やセキュリティ，ディペンダビリティといった，システムが満たすべき性質が満たされていることを示すための主張である．
- Context: システムの状態や構成要素といった前提条件を示す際に用いられるノードである．
- Strategy: Goal をいくつかの SubGoal に分割するための議論構造を表す．例えば，「システムの構成要素ごとに考える」という Strategy は，「システムは安全である」という Goal を，「サブシステム A は安全である」，「サブシステム B は安全である」，「サブシステムの相互作用は安全である」に分割する．
- Evidence: Goal が成立することを示すための証跡である．例えば，ソフトウェアテストの結果や合意書などが Evidence として用いられる．Evidence に支持されていない Goal は下部にひし形の Undeveloped エンティティが付与され，Undeveloped Goal として区別される．

3. 関連研究

本章では，関連研究及び GSN オーサリングツールの開発動向について説明する．

Assurance Cases の基となる Safety Cases はシステムが十分に安全であることが正当な証跡によって示された，構造化された議論を表す文章と定義されており[6]，例えば原子力産業においてはプラントでの怪我や生命の危機を引き起こすものや故障の分析と，それらが引き起こされる確率が十分に低いことを示すために用いられている．この他にも特に人命に関わるシステムへの適用に感心が高まってきている[5][7].

GSN の図形描画を行うためのエディタは複数存在する．Adelard LLP が開発する ASCE[4]は GSN や GSN の議論構造の基となる CAE の編集を行うためのツールである．商用利用が進んでいるものの，有料のツールであり，また Windows のみでしかサポートされていない．Eclipse のプラグインとして開発された D-Case Editor[3]は ASCE と同等の機能を持っており，近年ソースコードが公開されたオープンソースソフトウェアである．

この他にも NASA が開発した GSN のオーサリングツールとして，CertWare[1]や，GSN の議論構造をパターンとして再利用し，自動的に展開する機能を持つ AdvoCATE[2][8]がある．

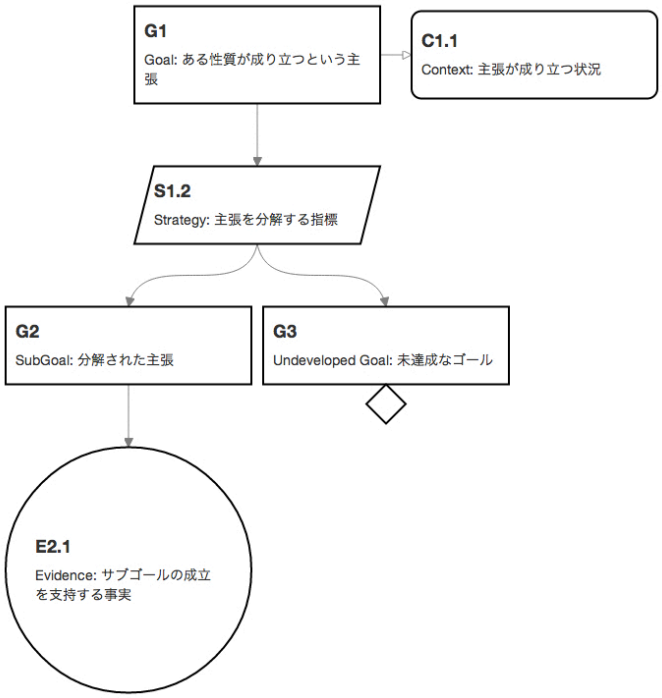

図 1 Goal Structuring Notation（GSN）

4. AssureNote の設計

本章では，AssureNote の設計について説明する．

4.1. ユーザインターフェースと編集機能

AssureNote のユーザインターフェースは主に，（1）サイドメニュー，（2）メニューバー，（3）コマンドラインの 3 つを用いて行う．サイドメニューは AssureNote の編集画面の左上部に常に表示されるメニュー群を差し，WGSN ファイルの読み込みや保存を行うメニューが表示される．メニューバーは図 2 に示すとおり，ノードを右クリックすることで表示される項目群を指す．コマンドラインは，コマンドをタイプすることで動作する機能である．

それぞれ，WGSN の保存などの常に有効な操作と，ノード毎に行う操作，補助的な操作として分類される．これらのユーザインターフェースは，後述のプラグイン機構を用いることで拡張することが可能である．

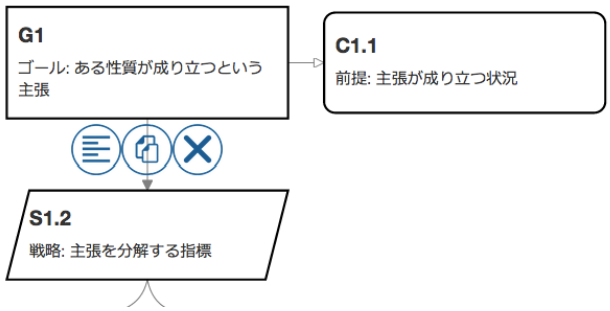

図 2 メニューバーの UI

4.2. プラグイン機構

AssureNote は，WGSN が編集機能の中心として採用され，複雑なユーザインターフェースの表示箇所を限定する設計を行っており，基本機能はシンプルな設計となっている．我々は機能の拡張を行うための，プラグイン機構を採用した．プラグインはファイル単位で定義されており，AssureNote のソースツリー内の特定の箇所に配置することで有効化される．AssureNote をブラウザで読み込む際にエントリポイントが呼び出され，プラグインの初期化と AssureNote への登録を行う．

5. WGSN (Wiki-Style GSN)

本説では Wiki スタイルの GSN 表記法である WGSN の設計について説明する．図 3 は，WGSN の表記法の定義のうち，Goal に関する箇所を抜粋したものである．

```
WGSN    ::= GOAL
          | CONTEXT
          | STRATEGY EVIDENCE
GOAL ::= GLABEL eol { DESCRIPTION } eol
       | GOAL_CHILDREN eol
GLABEL ::= ASTERISKS “G:” SYMBOL
         | ASTERISKS “G”
GOAL_CHILDREN ::= EVIDENCES
                | STRATEGIES
                | CONTEXT EVIDENCES
                | CONTEXT STRATEGIES
DESCRIPTION ::= TAG | free characters
TAG ::= KEY “::” VALUE
KEY ::= SYMBOL
VALUE::= SYMBOL | NUMBER
ASTERISKS ::= “*” ASTERISKS | “”
SYNBOL  ::= letter { letter | digit }
```

図 3 WGSN の表記法

5.1. WGSN への要求

第 1 章でも述べたとおり，我々は WGSN を用いた GSN をテキストベースの編集を実現することで，データの再利用性を高めることを目的としている．そのためには，WGSN には GSN を記述するに足る十分な表現力が必要となる．我々は，WGSN に要求される記述力を，以下のように定義した．

- GSN の基本構造を記述することができる能力．GSN のノードを定義し，ノード間の親子関係をテキストベースで表す必要がある．
- 既存の GSN エディタがサポートする機能を WGSN のみで表現することができる能力．AssureNote における GSN の編集を，WGSN を中心に行うためには，既存のエディタが GUI を用いて行っていた操作をテキストベースで表現しなければならない．

特に後者の要求に対して，WGSN を機械的に解釈するためのタグと呼ばれる記法を定義することで，既存の GSN エディタが持つ機能と同等のものをサポートする設計を行った．

5.2. 基本構造

WGSN は Wiki 表記を基にして開発されたものである．それぞれのノードはアスタリスク「*」から始まるラベル行を先頭とし，それ以降の自由記述が可能な行とで構成される．Goal, Context, Strategy, Evidence のいずれかのノードを示すためのラベルは，G, C, S, E のように各ノードのイニシャルを用いる．Undeveloped Goal は WGSN の記法上は定義されておらず，AssureNote などの GSN オーサリング上で，Evidence の存在しない Goal に対して自動的に Undeveloped エンティティを付与する設計となっている．ラベルの直後にコロン「:」から開始されるシンボルを記述することが可能だが，これはノードを一意に表す際に用いられる識別子として扱われる．

5.3. ノードの親子関係

ノードの親子関係は以下のルールに基づき，アスタリスクの数によって決定される．

- N 個のアスタリスクを持つ Goal は N-1 個のアスタリスクを持つ直前の Strategy の子
- N 個のアスタリスクを持 Strategy と Evidence は N-1 個のアスタリスクを持つ直前の Goal の子
- N 個のアスタリスクを持つ Context は，N 個のアスタリスクを持つ直前のノードの子
- 最上位の Goal (TopGoal) はアスタリスク一つ

図 4 は，WGSN の記述例である．この例では，*G:TopGoal* が最上位の Goal であり，*C:Context* と *S* のラベルを持つ二つのノードがその子となっている．*G:SubGoal* のアスタリスクの数は 2 つとなっているが，これは *G:SubGoal* が *S* の子であることを表す．

5.4. タグ

ラベル行以降はノードの本文となる．自由記述が可能な箇所だが，WGSN ではタグと呼ばれる記法を用いることでパラメータの定義を行うことができる．図 4 の例では，*C:Context* においてタグの定義がなされており，「システム」というパラメータに対して「Web 教育システム」という値がセットされている．タグの記法を用いることで，より構造化された GSN を記述することが可能である．この例では「Web 教育システム」に用いられている WGSN を，「改良された Web 教育システム」へと容易に修正することができる．

また，タグは AssureNote 上で WGSN をレンダリングする際の補助として用いられる他，Evidence として実行時のシステムの状態を与えるためのモニタリング機構などに用いられる．これらの機能は後述のプラグイン機構を用いて拡張することが可能である．

```
*G:TopGoal
ゴール: [システム]はディペンダブルである

*C:Context
前提: [G:TopGoal]が成り立つ状況
システム:: Web教育システム

*S 戦略: 主張を分解する指標

** G:SubGoal
サブゴール: 分解された主張

** E
証拠: [システム]における
サブゴールの成立を支持する事実

** C 反証: 証拠に対する反例

** G 未達成なサブゴール
```

図 4 WGSN の例

5.5. ラベルとタグの参照

シンボルによって一意に表されたラベルやタグを参照するには角括弧 [] を用いる．図 4 の例では，*G:TopGoal* において[システム]と書かれている箇所や，*C:Context* で[G:TopGoal]とラベルを参照している箇所で使用されている．これらの記法を用いて記述された WGSN を AssureNote 上で閲覧すると，AssureNote が自動的に割り振った値に置き換えて表示される．

定義されたラベルは WGSN 中のあらゆる箇所から参照することが可能である．他方，タグにはスコープが定義されており，スコープの外からこれを参照することは出来ない．タグのスコープの以下の通りに定義される．

- 定義されたタグはその子ノードや孫ノードのみで参照することができる
- 例外として，Context で定義されたタグのスコープは，Context の親と同一である
- タグが重複して定義されている場合は，最も近い親のものが優先される

Context は Goal や Strategy の前提条件を記述するためのノードであるため，例外的なスコープを持つ．図 4 の例では，「システム」を定義する *C:Context* は子を持たないが，その親である *G:TopGoal* と同じスコープを持つためこれを *G:TopGoal* や *E* で参照することができる．

6. AssureNote の実装

AssureNote は Web ブラウザ上で動作する Web アプリケーションとして実装されており，JavaScript 生成言語である TypeScirpt が使用されている．静的な HTML と JavaScript ファイルによって構成され，サーバサイドとのやりとりは無い．このため，ユーザは AssureNote のファイル群を事前に入手し，オフライン環境で使用することが可能である．図 5 は AssureNote の編集画面を示している．AssureNote は WGSN を主な編集法としているため，Eclipse をベースとした多くの GSN オーサリングツールと比較して，ボタン等のユーザインターフェースが少ないのが特徴である．

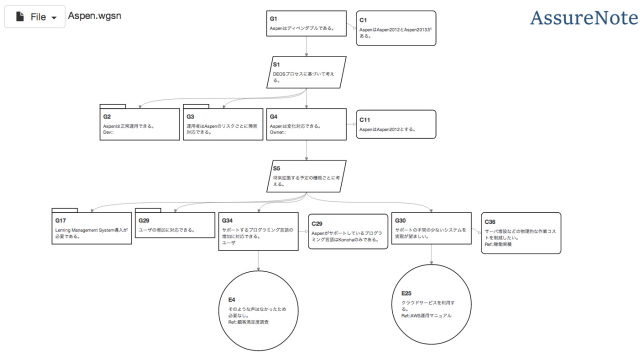

図 5 AssureNote の編集画面

6.1. WGSN パーサの実装

WGSN パーサは，第 5 章で定義される WGSN の構文解析などを行う機構であり，主に以下の二つの役割を持つ．

1. 構文解析：WGSN の構文解析を行い，抽象構文木（AST）を生成する機能．なお，WGSN の AST はそのまま AssureNote の内部表現として用いられる．
2. WGSN 生成：任意の AST から WGSN を生成する機能．AssureNote 上で編集中の GSN をエディタへと出力する際や，WGSN 形式で保存する際に用いられる．

AssureNote は，任意のノードの部分木のみを編集する機能を持つため，WGSN パーサは GSN 中の任意の箇所からパース・WGSN 生成が行える必要がある．そのため，WGSN 生成を行う際にはノードの親子関係を表すアスタリスクの数を必ず 1 から始まるようになっている．このため，*TopGoal* 以外のノードから WGSN 生成を行った際，生成された WGSN の見かけ上は WGSN 生成の基点となるノードが *TopGoal* となる．なお，Strategy や Context，Evidence を基点として WGSN

生成を行う事も可能だが，その場合もアスタリスクは 1 から開始される．

6.2. 編集機構

AssureNote の編集機構はそれぞれのノードを右クリックすることで表示されるメニューバーから行う．メニューバーの中には WGSN による編集を行うためのエディタを開くメニューが存在し，これを編集のために主に用いるが，ノードの削除など一部の操作については UI から行う．

6.3. WGSN を用いた編集機構

図 6 は AssureNote 上で WGSN を編集する際に用いられる全画面エディタを表している．全画面エディタは各ノードのメニューバーから開くことができるが，メニューバーを表示されるノードに応じて全画面エディタ上の WGSN は変化する．最上位の Goal である *TopGoal* から全画面エディタを表示させた際は WGSN の全体をエディタ上から編集することが出来るが，それ以外の任意のノードの場合は，その部分木に当たる箇所のみが WGSN 生成器を用いてエディタ上に表示される．

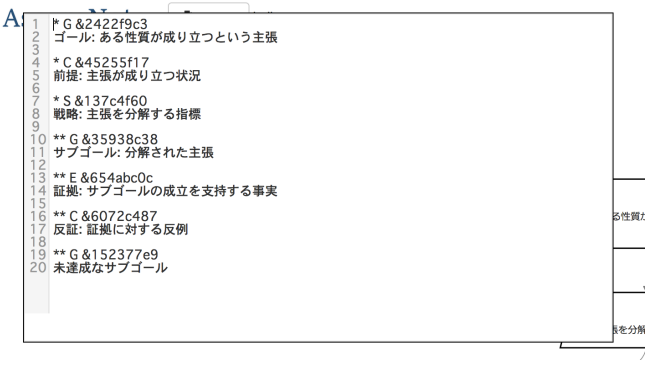

図 6 全画面エディタ

6.4. 補助的な編集機構

前述の通り，一部の編集機能はメニューバーから直接呼び出すためのボタンが備わっている．ただし，これらの機能は WGSN を用いたテキストベースの編集の欠点を補うことを目的としている．

ノードの削除を行うためのメニューはその代表である．WGSN を用いたノードの削除を行うためには，削除したいノードの部分木に当たる箇所を全て選択し，削除しなければならず，操作が煩雑になってしまう．そこで，右クリックでメニューバーを表示したノードの部分木を削除する機能は，WGSN を用いた編集機能とは別途用意した．

6.5. プラグイン機構の実装

AssureNote のプラグインの分類は以下に述べる通りである．プラグインの実行は，レンダリングの開始時点や，メニューバーの選択時などに設定されるフックポイントから行われる．

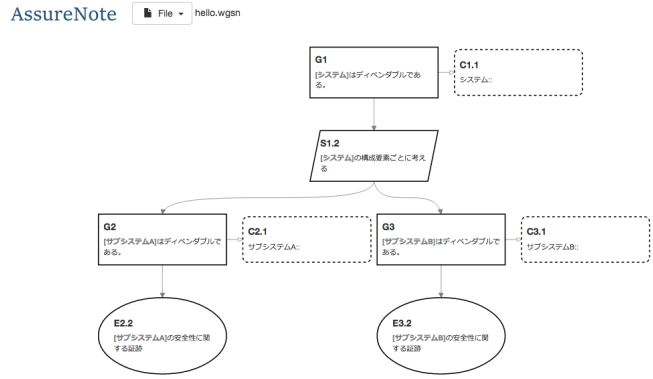

図 7 プラグインによるレンダリング

6.5.1. UI の拡張プラグイン

AssureNote 上のサイドバーやメニューバーに項目を追加する．メニューに表示される画像や項目名を指定し，項目が選択された際の処理を記述する．

6.5.2. コマンドライン拡張プラグイン

AssureNote で補助的な操作を行うためのコマンドラインで使用するコマンドは，組み込みの物も含めて全てがプラグインとして実装されている．コマンドライン拡張プラグインでは，コマンド名と，パースされたコマンド引数を基に処理を行う関数を定義する．プラグインとして定義される機能の抜粋を以下に示す．なお，これらのコマンドには引数が必要となるものも存在するが，省略して説明を行う．

- help：ヘルプの参照
- new：TopGoal のみの新しい GSN を作成する
- open：ファイルを開く
- save：ファイルを保存する
- save-as-svg：SVG 形式でファイルを保存する
- set-scale：GSN の拡大率を変更する
- set-color：ノードを色付けする
- connect：複数人同時編集サーバへ接続する
- @：同時編集者へメッセージを送信する

6.5.3. レンダリングプラグイン

レンダリングプラグインは，画像の表示や特定のノードのスタイルを変更する際に用いられる．ただし，レンダリングプラグインは個々のノードのレンダリングを行うためのものであり，AssureNote のレイアウトの変更やノードの配置を変更することはできない．

WGSN のタグを解釈することで, レンダリングを行う. 例えば, TODO を表示するプラグインでは,「TODO」というパラメータが定義されているか, 空の値がセットされているパラメータが定義されているノードは枠が点線で表示される. 図 7 の例では, 3 つの *Context* のスタイルがレンダリングプラグインによって変更されている.

7. GSN の記述実験

本章では, AssureNote を用いて GSN を記述し, WGSN を用いることで GSN の記述に十分な表現が可能であることを確認する. ここで WGSN の表現力とは, 第 1 章で述べた通り, (1) 木構造の文書である GSN の基本的な構造を定義することが可能な点と, (2) 既存の GSN エディタが GUI を用いて行っていた操作が, WGSN を用いて表現可能な点である.

7.1. Web 教育システムの GSN

本記述実験では, 我々が開発を行っている Aspen と呼ばれる Web 教育システム (図 8) の Assurance Cases を AssureNote に組み込まれた WGSN を用いて記述する. そして, WGSN を用いた GSN の基本的な構造の記述が, 実際のシステムの Assurance Cases を記述する際にも十分に有用であることを確認する.

Aspen はプログラミング初学者向けの教育システムであり, ブラウザ上でプログラムを記述し, その結果を表示することでプログラミング学習を進めていく. Aspen はプログラムの編集と実行結果の表示を行う Aspen クライアントと, ユーザデータの管理やプログラムのコンパイルを行う Aspen サーバから構成される. Aspen サーバはロードバランサによって冗長化されており, 2 つのアプリケーションサーバが動作している.

本記述実験は Assurance Cases 記述経験が豊富な 3 名と, Aspen の設計・開発・運用に関わった技術者を中心として行われた. いずれのメンバも既存の GSN オーサリングツールを用いた Assurance Cases の作成経験はあるものの, AssureNote の使用経験は無い. なお, 本来 Assurance Cases の作成はシステムの設計段階から始める必要があるが, 今回は既に構築済みのシステムに対して, システムの設計から運用を担当した技術者への聞き取りの基, 当時のプロセスを追跡する形で行われた.

記述実験参加者は「Aspen はディペンダブルである」を TopGoal として設定し, Aspen のディペンダビリティを,「Aspen は正常に運用することができる」「運用者は Aspen のリスク毎に障害対応できる」「Aspen は要求の変化に対して対応することができる」の SubGoal に分割し, それぞれについて議論を行った. 作成された Assurance Cases は表 1 の通り, 250 ノードとなった. この Assurance Cases は 854 行の WGSN として編集が行われたものである.

既存の GSN オーサリングツールとは異なり, GSN の編集を, WGSN を用いて行うため, 記法の習得に不慣れな段階では編集に手間取る場合が見受けられた. しかし, WGSN に慣れてきた実験参加者は, WGSN の一部をコピー&ペーストすることで GSN の部分木をそのまま複製するといった操作手法を独自に見つけ出し, 用いていた. こうした機能は他の GSN オーサリングツールでも採用されている機能の一つである. 既存の GSN オーサリングツールでは GUI のメニューより一つ一つノードを作成し, 本文を記述する必要がある. 他方, WGSN を用いることで, Assurance Cases 作成者は複数のノードの追加や削除, 編集を行う事ができる.

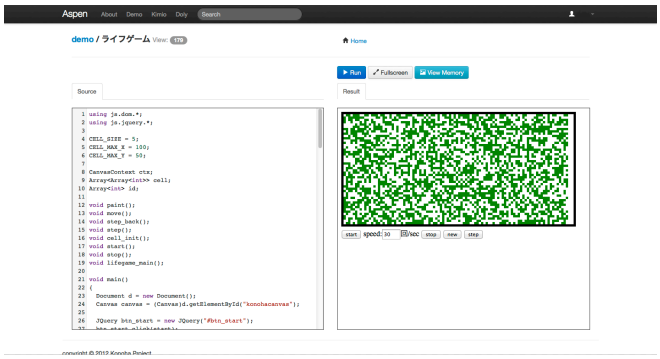

図 8 Aspen の操作画面

表 1 記述実験で作成された GSN の概要

ノード数	250 ノード
WGSN の行数	854 行

8. まとめ

我々は, Assurance Cases の代表的なビジュアル表記法である GSN のデータ再利用性とグラフィカル編集機能を両立させた GSN オーサリングツールである AssureNote の設計と実装を行った. AssureNote は GUI による機能の追加を最小限に抑え, プラグインによる機能拡張を行う設計がなされている. そのために, GSN をテキストベースで編集するための表記法である WGSN の設計を行い, WGSN のパーサを AssureNote に組み込んだ. WGSN を用いることで, GSN オーサリングツールの補助無しに十分な記述力を得る事が可能となり, AssureNote の機能拡張も容易に行う事が可能であった. また, GSN の基本構造の記述や, 既存の GSN オーサリングツールが持つ機能のサポートが AssureNote でも行えることが記述実験より確認された.

我々が行った AssureNote の記述実験では, WGSN の

記述力や機能性関して AssureNote 使用者の感覚的な変化にのみ注目しているため，同一の Assurance Cases の作成コストの比較など,定量的な評価が必要である．現在の AssureNote はサーバとのやりとりなしにブラウザ上で動作するアプリケーションとなっている．現在実装が行われており，試験的に導入されている複数人による GSN の同時編集機能やサーバへの GSN アップロード機能などのサポートを行い，実用的なソフトウェアを目指す予定である．

AssureNote はオープンソースで開発が行われているソフトウェアであり,下記の URL から自由にダウンロードし，使用する事が可能である．

https://github.com/AssureNote/AssureNote

また，これらの入手可能なファイルは下記の URL に設置されており，ブラウザ上からアクセスすることで AssureNote を使用することができる．

http://www.ubicg.ynu.ac.jp/AssureNote/

謝辞 本研究は，JST/CREST「実用化を目指した組込みシステム用ディペンタブル・オペレーティングシステム」領域の研究課題「実行時の安全性を確保する SecurityWeaver と P-SCRIPT」の一部として行われた．

参 考 文 献

[1] Matthew R. Barry. 2011. CertWare: A workbench for safety case production and analysis. In *Proceedings of the 2011 IEEE Aerospace Conference* (AERO '11). IEEE Computer Society, Washington, DC, USA, 1-10.

[2] Ewen Denney, Ganesh Pai, and Josef Pohl. 2012. AdvoCATE: an assurance case automation toolset. In *Proceedings of the 2012 international conference on Computer Safety, Reliability, and Security* (SAFECOMP'12), Frank Ortmeier and Peter Daniel (Eds.). Springer-Verlag, Berlin, Heidelberg, 8-21.

[3] Matsuno, Yutaka, Hiroki Takamura, and Yutaka Ishikawa. "A dependability case editor with pattern library." *High-Assurance Systems Engineering (HASE), 2010 IEEE 12th International Symposium on*. IEEE, 2010.

[4] LLP, A.: Adelard ASCE, http://www.adelard.com/

[5] Jee, Eunkyoung, Insup Lee, and Oleg Sokolsky. "Assurance cases in model-driven development of the pacemaker software." *Leveraging Applications of Formal Methods, Verification, and Validation*. Springer Berlin Heidelberg, 2010. 343-356.

[6] ine Menon, Cather, Richard Hawkins, and John McDermid. "Defence standard 00-56 issue 4: Towards evidence-based safety standards." *Safety-Critical Systems: Problems, Process and Practice*. Springer London, 2009. 223-243.

[7] Bloomfield, Robin, and Peter Bishop. "Safety and assurance cases: Past, present and possible future–an Adelard perspective." *Making Systems Safer*. Springer London, 2010. 51-67.

[8] Denney, Ewen, and Ganesh Pai. "A Formal Basis for Safety Case Patterns." *Computer Safety, Reliability, and Security*. Springer Berlin Heidelberg, 2013. 21-32.