



用户手册

SSCC 基于场景规约的服务验证工具

1 引言	3
1.1 编写目的.....	3
1.2 背景.....	3
1.3 定义.....	3
1.4 参考资料.....	3
2 用途	4
2.1 功能.....	4
2.2 性能.....	4
2.2.1 精度.....	4
2.2.2 灵活性.....	4
3 运行环境	5
3.1 硬设备.....	5
3.2 支持软件.....	5
4 使用过程	5
4.1 安装与初始化.....	5
4.2 输入.....	5
4.2.1 输入数据的现实背景.....	5
4.2.2 输入格式.....	5
4.2.3 输入举例.....	6
4.3 输出.....	9
4.3.1 输出数据的现实背景.....	9
4.3.2 输出格式.....	9
4.3.3 输出举例.....	9

1 引言

1.1 编写目的

本手册是 SSCC 基于场景规约的服务验证工具的用户手册，旨在帮助该工具的用户正确使用工具所提供的功能。

1.2 背景

Web 服务技术在开发和执行分布于网络上的各种业务流程的领域中起着核心作用，对 Web 服务行为和组合的验证是在 Web 服务相关研究领域日益引起关注的一个论题。因为 Web 服务是由消息驱动的，而且服务可被看作一个构件，所以我们重点关注服务之间的消息交互顺序，并且针对基于场景的规约（Scenario-based Specification）来验证消息交互顺序的一致性。基于场景的规约提供了一种直观的、可视化的方法来描述系统需求，在系统的规约和设计方面起着日益重要的作用，基于场景的规约是由 UML 活动图（UML Sequence Diagrams）来描述的。SSCC 重点关注消息交互顺序这一性质。作为由 Web 服务标准族的成员，WS-BPEL 被用于 Web 服务的行为描述和组合。SSCC 针对基于场景的规约来验证消息交互顺序的一致性，并使用 Petri 网作为 WS-BPEL 消息流转换的中间模型。为了减小状态空间、提高验证效率，SSCC 还对输入的 WS-BPEL 以及 Petri 网模型中与消息交互无关的活动和元素进行了简化。

SSCC 分两步实现了上述基于场景规约的验证：首先，将 WS-BPEL 消息流自动转换为 Petri 网模型，并且在转换过程中对 WS-BPEL 源码和 Petri 网模型先后进行简化；随后，使用自动验证工具针对 UML 活动图规约和 Petri 网模型来验证 Web 服务的存在或强制一致性。

SSCC 由南京大学计算机科学与技术系软件工程研究组独立研制。它可以读入 WS-BPEL、WSDL 以及 UML 顺序图文件，进行分析，并生成化简的 WS-BPEL 代码以及转化得到的 Petri 网的 PNML 表示，验证的结果也保存在文件中。

1.3 定义

WS-BPEL: Web Services Business Process Execution Language (Web 服务-业务流程执行语言)

UML: Unified Modeling Language

PNML: Petri Net Markup Language

1.4 参考资料

- LI Xuandong, HU Jun, BU Lei, ZHAO Jianhua and ZHENG Guoliang. Consistency Checking of Concurrent Models for Scenario-based Specifications. In Proc. of 12th International SDL Forum. 2005. 298–312
- YANG Lu, LIU Xi, LI Xuandong. Scenario-based Verification of Web Services Communication and Composition. Technical Report, Software Engineering Group, Nanjing

2 用途

2.1 功能

此版本的 SSCC 的主要功能包括：

- WS-BPEL 以及 WSDL 文件的读入与解析：读取符合 WS-BPEL v2.0 以及 WSDL 标准的文件，并对其中的元素进行解析；
- UML 顺序图读入和解析：读取由 Rational Rose 生成的 UML 模型文件并且显示其中的 UML 顺序图，解析 UML 顺序图中的各种元素；
- WS-BPEL 的化简：为减小验证状态空间、提高验证效率，对 WS-BPEL 中与正常流程的消息交互无关的元素进行删除或合并，对服务流程进行化简；
- WS-BPEL 到 Petri 网模型的转换；
- Petri 网模型的化简：为减小验证状态空间、提高验证效率，对转化生成的 Petri 网模型进行化简；
- 基于场景规约的验证：对化简之后的 Petri 网模型针对 UML 顺序图规约进行验证，验证策略可由用户进行选择；
- 结果保存：对化简后的 WS-BPEL、转化得到的 Petri 网模型（以 PNML 格式）以及验证结果进行保存，方便用户查阅。

此版本 SSCC 的验证策略分类如下：

- I. 存在一致性验证
- II. 强制一致性验证
 - A. 向前强制一致性验证
 - B. 向后强制一致性验证
 - C. 双向强制一致性验证

2.2 性能

暂无要求

2.2.1 精度

对 WS-BPEL 的各种活动、结构都能够进行分析，对 WS-BPEL 流程中的消息交互均有考虑。

2.2.2 灵活性

本软件基于 Eclipse 平台用 Java 语言开发，可在多种操作系统平台上运行，可移植性高。

3 运行环境

3.1 支持硬件

硬件：CPU: Pentium(R) 2 及以上；内存: 128M 及以上；硬盘：4G 及以上

3.2 支持软件

Jasmine 支持在下列操作系统及平台上运行：

- Microsoft Windows 2000 全系列
- Microsoft Windows XP 全系列
- Microsoft Windows Vista 全系列
- Linux 全系列
- JDK1.5 以上版本

4 使用过程

4.1 安装与运行

SSCC 部署版本由一个 SSCC.jar 文件构成。不需要安装双击即可直接运行。

4.2 输入

本工具的输入主要包括 WS-BPEL、WSDL、1~3 个 UML 顺序图模型文件。配置数据输入在“输入举例”一节一并介绍。

4.2.1 输入数据的现实背景

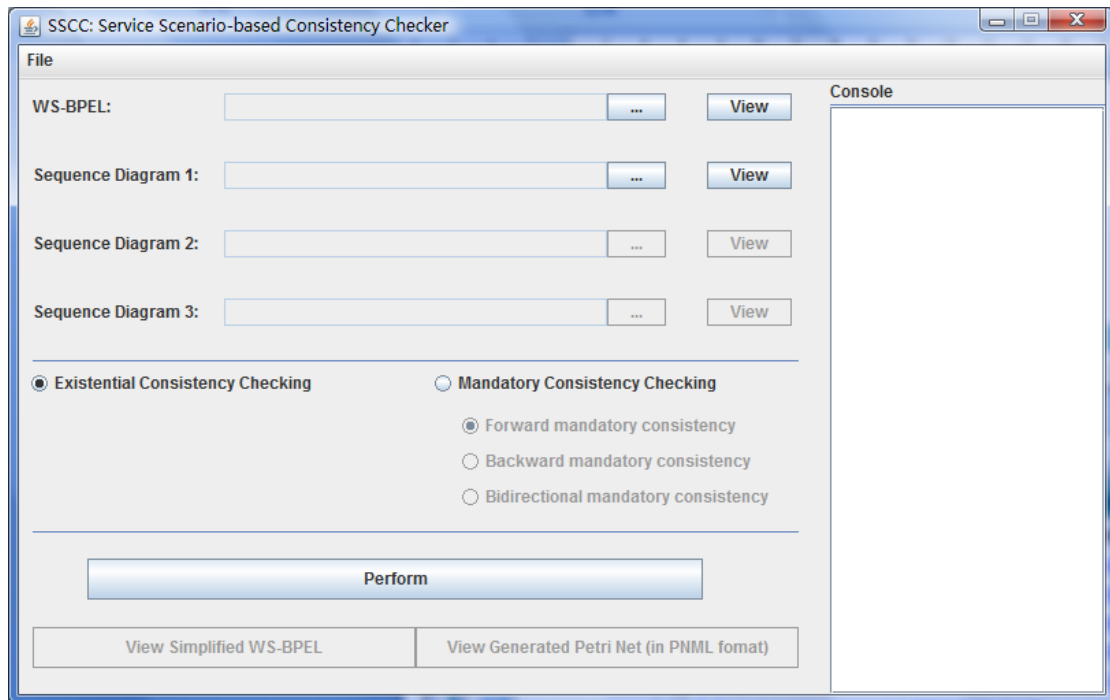
UML 顺序图反映了交互的服务之间的消息时间顺序关系，是服务交互行为的可视化描述。

4.2.2 输入格式

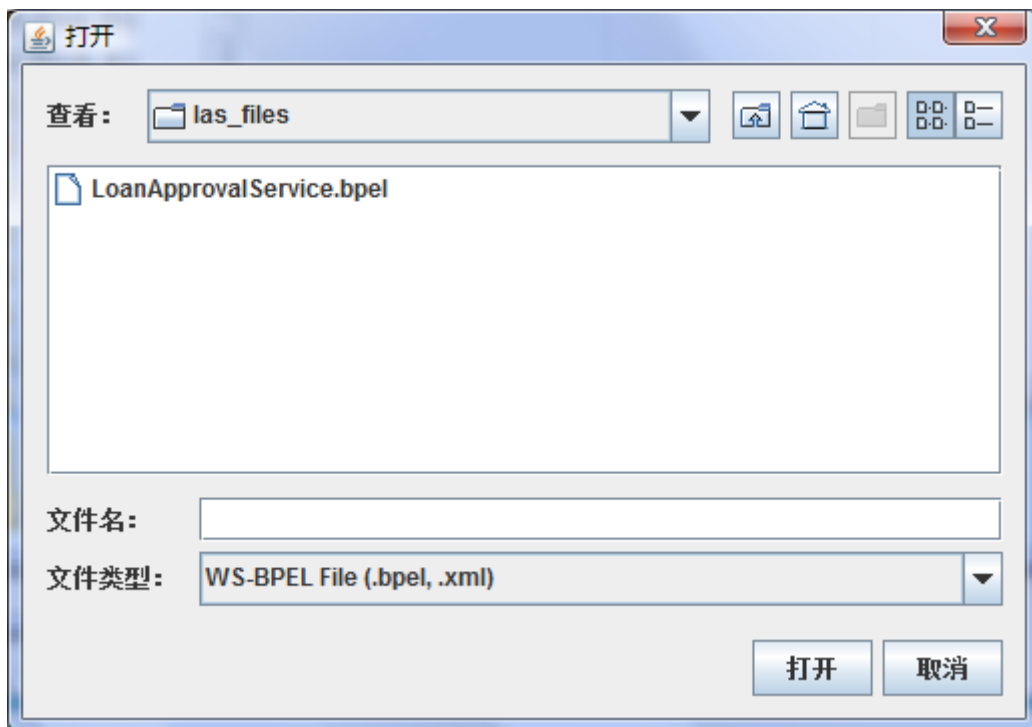
WS-BPEL 和 UML 顺序图模型文件均要求以文件位置的方式提供（WSDL 位置通过 WS-BPEL 中导入位置查找）。

4.2.3 输入举例

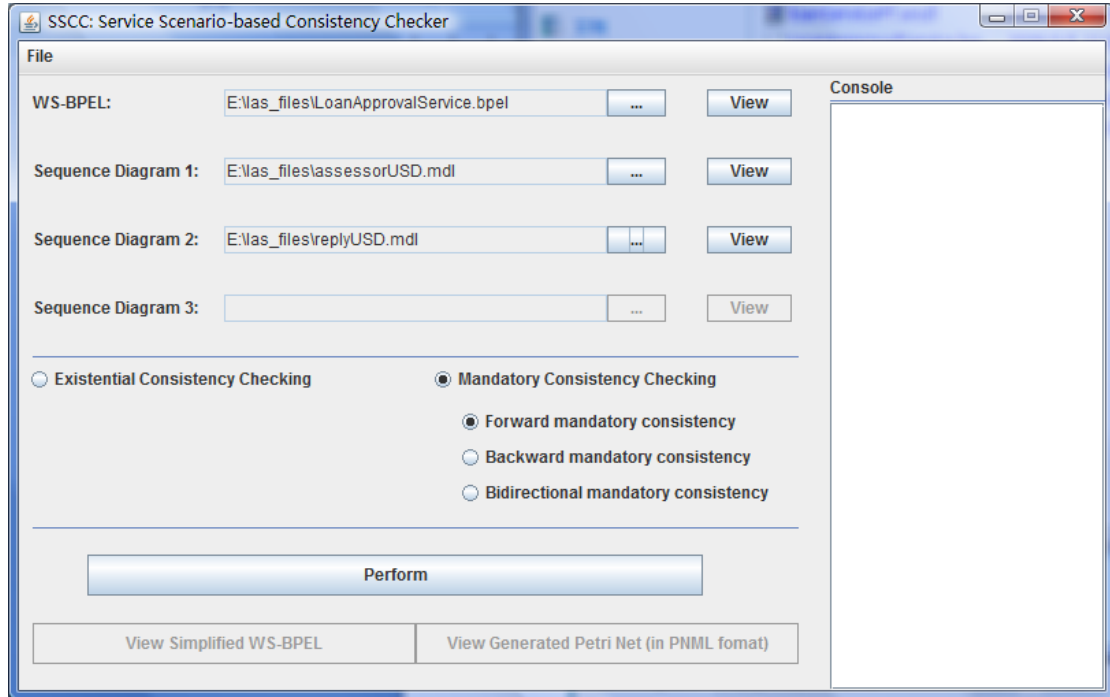
启动 SSCC 软件：



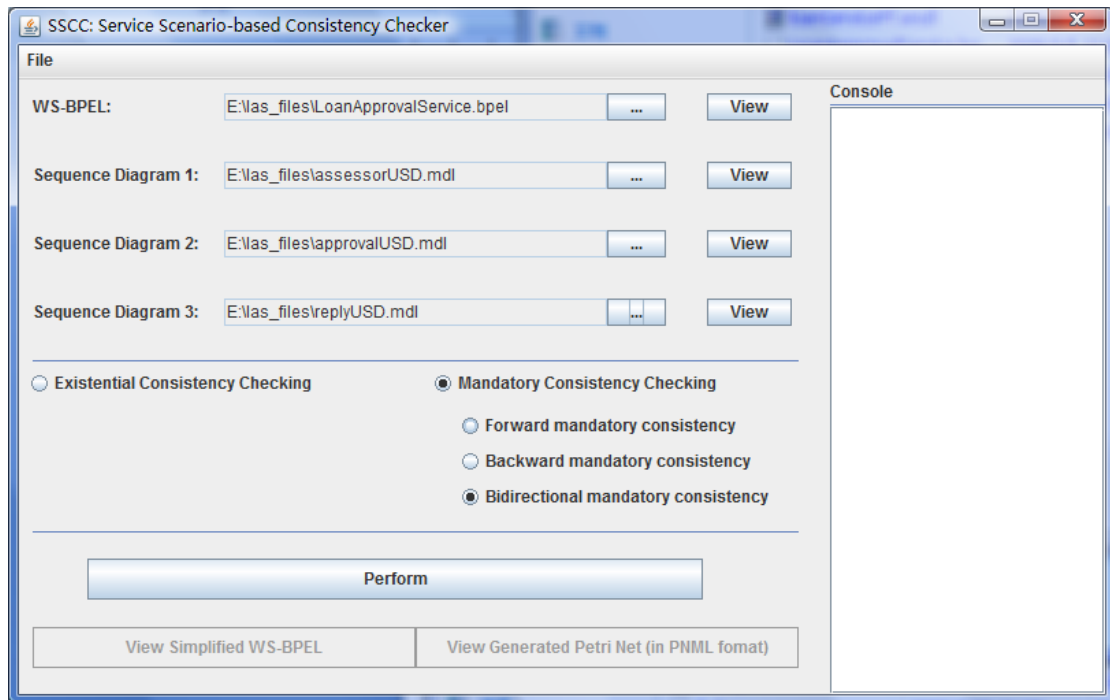
首先选择 WS-BPEL 文件的位置，点击“WS-BPEL”标签行右边的“...”按钮，打开文件选择对话框。选择文件，并打开。如下所示：



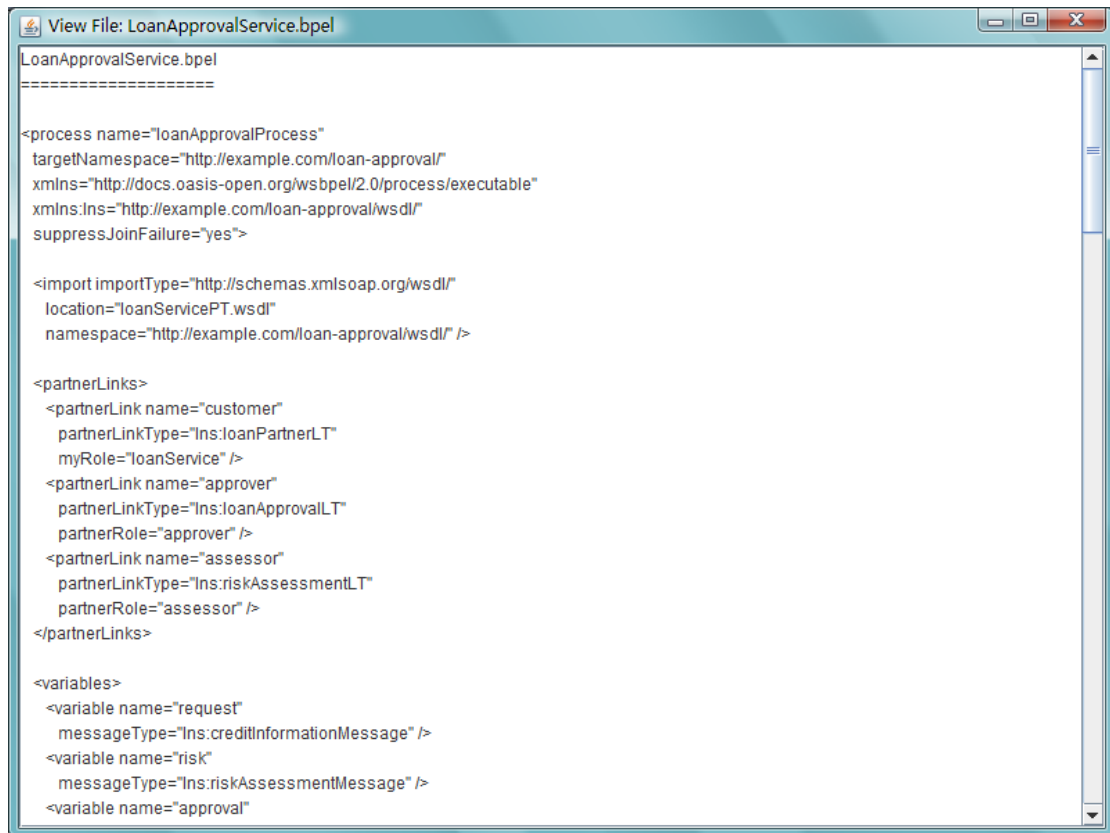
然后依次选择 UML 消息顺序图的位置。选择之后如下图所示：



然后选择拟采用的一致性验证的策略。如选择存在一致性验证，则选择“Existential Consistency Checking”单选按钮，如果选择强制一致性验证，则选择“Mandatory Consistency Checking”单选按钮。默认的策略是存在一致性验证。若选择了强制一致性验证策略，默认的子策略是向前强制一致性（即“Forward mandatory consistency”）。用户也可选择向后强制一致性（即“Backward mandatory consistency”）和双向强制一致性（即“Bidirectional mandatory consistency”）子策略。若选择向前/向后强制一致性子策略，还需选择第二个 UML 顺序图文件的位置；若选择了双向强制一致性子策略，就一共需要指名三个 UML 顺序图文件的位置。如下所示：



所有指定了位置的文件，均可通过点击右边的“View”按钮来查看。



完成上述输入，即可点击“Perform”按钮开始执行验证。

4.3 输出

程序运行结束后，产生的化简 WS-BPEL 代码、Petri 网模型的 PNML 文件以及验证结果，都保存在输入的 WS-BPEL 文件所在的文件夹下。

4.3.1 输出数据

化简 WS-BPEL 代码

Petri 网模型的 PNML

控制台输出（包括验证用时）

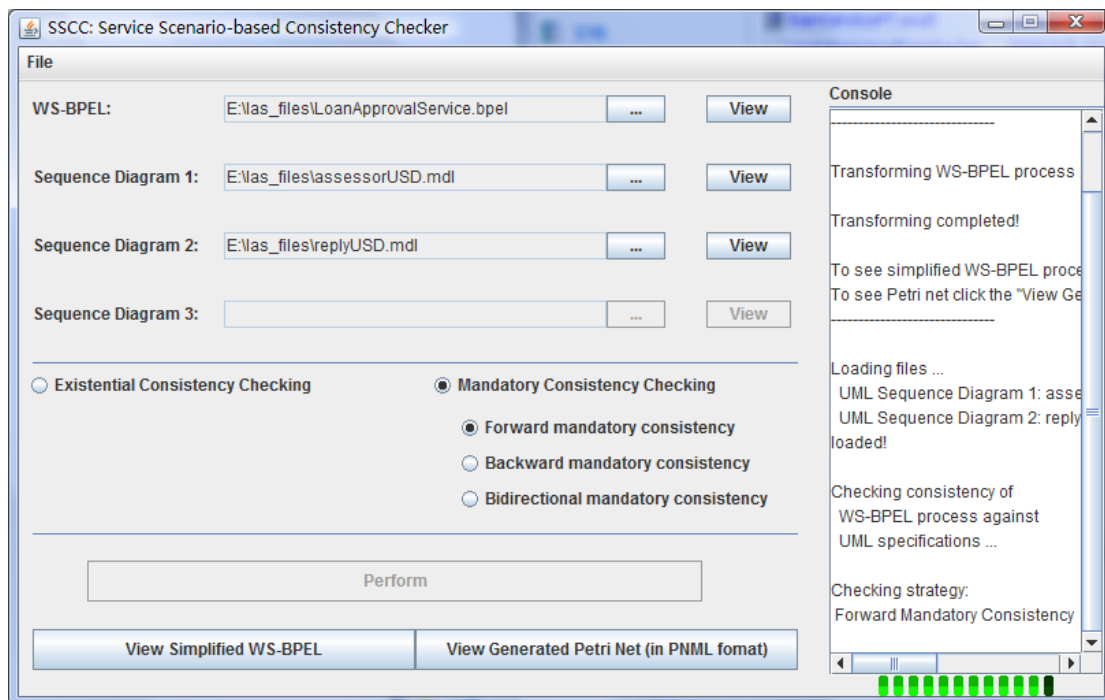
验证结果：满足条件的路径文件，满足条件的转换集合文件

4.3.2 输出格式

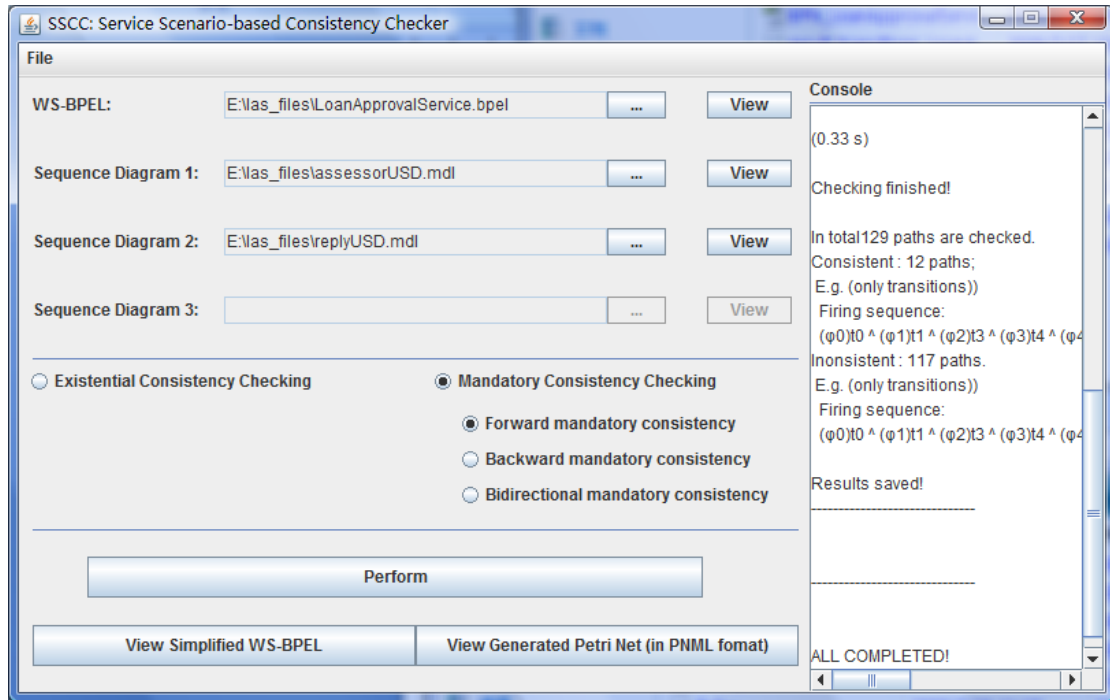
以上各输出的表示格式在“输出举例”一节详细介绍。

4.3.3 输出举例

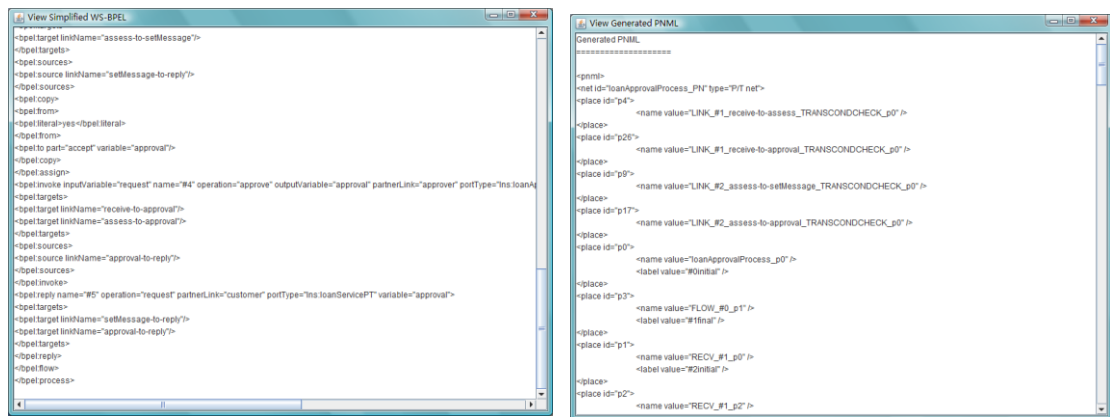
程序运行过程如下图所示：



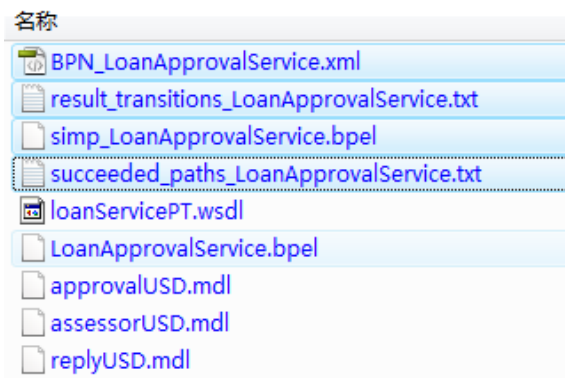
程序运行结束后，出控制台输出信息外，输出文件保存在输入的 WS-BPEL 文件所在的文件夹下。其中，控制台输出如下（可以看到这次验证仅用时 0.33 秒）：



可通过分别点击界面上的“View Simplified WS-BPEL”按钮和“View Generated Petri Net (in PNML format)”按钮，分别浏览产生的化简过后的 WS-BPEL 代码以及 Petri 网模型的 PNML 文件。



产生的验证结果则保存在输入 WS-BPEL 所在的文件夹中。



所有成功路径信息保存在 `succeeded_path_LoanApprovalService.txt` 中，如下所示：

```
Successful paths:
12 paths

0: Path:
Marking sequence:
(u0)μ: { p0 }
(u1)μ: { p1, p24, p25, p26, p27 }
(u2)μ: { p24, p25, p26, p27, p2, p4, p23 }
(u3)μ: { p25, p26, p27, p2, p23, p5 }
(u4)μ: { p25, p26, p27, p2, p23, p6, p21 }
(u5)μ: { p25, p26, p27, p2, p23, p21, p7, p8, p15 }
(u6)μ: { p25, p27, p2, p21, p7, p8, p16 }
(u7)μ: { p27, p2, p21, p7, p16, p9 }
(u8)μ: { p27, p2, p21, p7, p9, p19, p18 }
(u9)μ: { p27, p2, p21, p7, p19, p18, p11, p10 }
(u10)μ: { p2, p21, p7, p18, p10, p12 }
Firing sequence:
(φ0)t0 ^ (φ1)t1 ^ (φ2)t3 ^ (φ3)t4 ^ (φ4)t5 ^ (φ5)t12 ^ (φ6)t6 ^ (φ7)t15 ^ (φ8)t11 ^ (φ9)t8 ^ (φ10)t9

1: Path:
Marking sequence:
(u0)μ: { p0 }
(u1)μ: { p1, p24, p25, p26, p27 }
(u2)μ: { p24, p25, p26, p27, p2, p4, p23 }
(u3)μ: { p25, p26, p27, p2, p23, p5 }
(u4)μ: { p25, p26, p27, p2, p23, p6, p21 }
(u5)μ: { p25, p26, p27, p2, p23, p21, p7, p8, p15 }
(u6)μ: { p25, p27, p2, p21, p7, p8, p16 }
(u7)μ: { p25, p27, p2, p21, p7, p8, p19, p18 }
(u8)μ: { p27, p2, p21, p7, p19, p18, p9 }
(u9)μ: { p27, p2, p21, p7, p19, p18, p11, p10 }
(u10)μ: { p2, p21, p7, p18, p10, p12 }
Firing sequence:
```

为实现该场景，必须触发的转换和可能被触发的转换的信息保存在 `result_transitions_LoanApprovalService.txt` 中。其内容如下所示：

```
All Possible Transitions:
Transitions: {t0, t1, t3, t4, t5, t12, t6, t15, t11, t8, t9, t7}

Must-be-fired Transitions:
Transitions: {t0, t1, t3, t4, t5, t12, t6, t15, t8, t9}
```

`simp_LoanApprovalService.bpel` 保存了化简之后的 WS-BPEL 代码，`BPN_LoanApprovalService.xml` 即 Petri 网模型的 PNML 文件。