ESTRELLA

IST-2004-027655

European project for Standardized Transparent Representations in order to Extend LegaL Accessibility
Specific Targeted Research or Innovation Project

Specific Targeted Research Project
Information Society Technologies

Deliverable N°: 4.2

Refined translators to and from LKIF for each of the knowledge formats of the participating vendors

Version:	FINAL 1.0 HALEY
Due date of Deliverable:	September 30, 2008
Actual submission date:	September 30, 2008
Start date of Project:	1 January 2006
Duration:	30 months
Project Coordinator:	Universiteit van Amsterdam (NL)
Lead contractor deliverable:	Corvinus University Budapest
Participating contractors:	Alma Mater Studiorum - Universita di Bologna (IT), University of Liverpool (UK), Fraunhofer Gesellschaft zur foerderung der angewandten forschung e.v. (DE), RuleWise b.v. (NL), RuleBurst (EUROPE) Ltd. (UK), knowledgeTools International GmbH (DE), Interaction Design Ltd. (UK), SOGEI - Societa Generale d'Informatica S.P.A. (IT), Ministro per le Riforme e le Innivazioni nella Publica Amministrazione (IT), Hungarian Tax and Financial Control Administration (HU), Budapesti Corvinus Egyetem (HU), Ministero dell'Economia e delle Finanze (IT), Consorzio Pisa Ricerche SCARL (IT)

Project funded by the European Community under the 6th Framework Programme

	Dissemination Level	
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Estrella User Report

Haley Office Rules (formerly RuleBurst Studio) is a business rules management system that combines propositional logic with a natural language parser to run web-based interviews with human users. Haley Office Rules gives users the ability to export their projects as a PIF (Project Interchange File). A PIF is a stand-alone XML file that represents the project attributes, business rules, and rule-associated data. Apart from the rules, much of the data in this file is associated with natural language parsing and is not directly pertinent to the inference engine. Conceptually, each rule is composed of exactly one 'conclusion' statement which is the root node of a tree of 'premise' statements. Each premise can be composed of more premise statements. The breadth and depth of this tree can be as large as necessary. In practice, the PIF XML is not structured like a tree but contains explicit 'level' attributes that indicate the depth at which the statement is located. This is the main structural difference between the two formats as LKIF theory rules are structured as trees from the start.

A LKIF – PIF translator was built to prove the interchange-ability of the two formats. Translating from a PIF to an LKIF is useful to Haley because it strips the PIF format of its Haley-specific project and natural language parsing data while still providing an expressive language that can represent Haley rules. The data lost in this translation is not pertinent to the representation of rules and can be re-derived after being imported back into Haley Office Rules. Even though there is a distinct difference between the intended uses of the two formats (propositional logic vs. predicate logic), LKIF is still able to properly express the Haley rules.

Because of its expressiveness however, not all of LKIF's features can be directly translated into a PIF. For instance, Haley Office Rules does not allow exception rules (rule:excluded). This like other features that do not have a direct translation, can still be worked around.

The Estrella project has provides Haley with the opportunity to collaborate with fellow consortium members on the formation of these new standards and how they might be applied in a real-world scenario. This knowledge keeps us at the forefront of emerging methodologies in the field. We are interested in adopting LKIF as a supported standard to allow customers freedom with regards to interoperability. Haley also has many European Government clients interested in open standards for rules:

- Forsakringskassan, Danish SKAT, HMRC, BusinessLink, Swedish Tax, Dutch Statistics, Swedish Social Care
- Outside Europe IRS, Singapore MinDef, Australian Tax
- Other sectors e.g. BUPA, Nuon, BAT

We are keen to be at the forefront of technology, to win and keep customers not by restricting choice but by being the best in their field. Our involvement in the Estrella project will certainly help us towards this goal.

haley™

Technical Specification

Haley® Interchange Format Converter

Copyright

Document Properties

Title:	Haley Interchange Format Converter Technical Specification	
Filename:	Converter Technical Specification.doc	
Product Analyst:		
Developer:		
Tester:		
Reviewed By:		
Signed Off By:		Date:
		Date:
		Date:

Document History

Version	Date Released	Author	Comments
0.1		Brandon B.	First Draft

Contents

Document Information

Purpose

This document is the Technical Specification for the Interchange Format Converter.

Audience

The intended audience for this document is the Haley development team.

Documentation Set

Haley Interchange Format Converter Development Documentation

Haley Interchange Format Converter Functional Specification Documentation

Requirement Naming

To facilitate references to these requirements in other documents such as project plans and defect reports, each requirement is identified by an abbreviated name, given in the heading. Use Cases, and Error/Warning Conditions are referenced by numbers beginning with U, E or W respectively.

To prevent references from becoming out of date, when requirements or Error/Warning Conditions are added they must be added to the end of the existing list. Do not renumber the remaining requirements when a requirement is deleted.

Definitions, Acronyms, and Abbreviations

Term	Meaning
Converter	Haley Interchange Format Converter
XML	Extensible Markup Language
LKIF	Legal Knowledge Interchange Format
PIF	Project Interchange Format
RB Project	RuleBurst Studio Project

System/Release Overview

Purpose

The Haley Interchange Format Converter provides a translator between the two XML interchange formats LKIF and PIF.

Objectives

The solution is comprised of two main stages.

Input

- Parse the XML document
- Save the necessary data to internal data structures

Output

- Write internal data structures to XML format of choice

Context

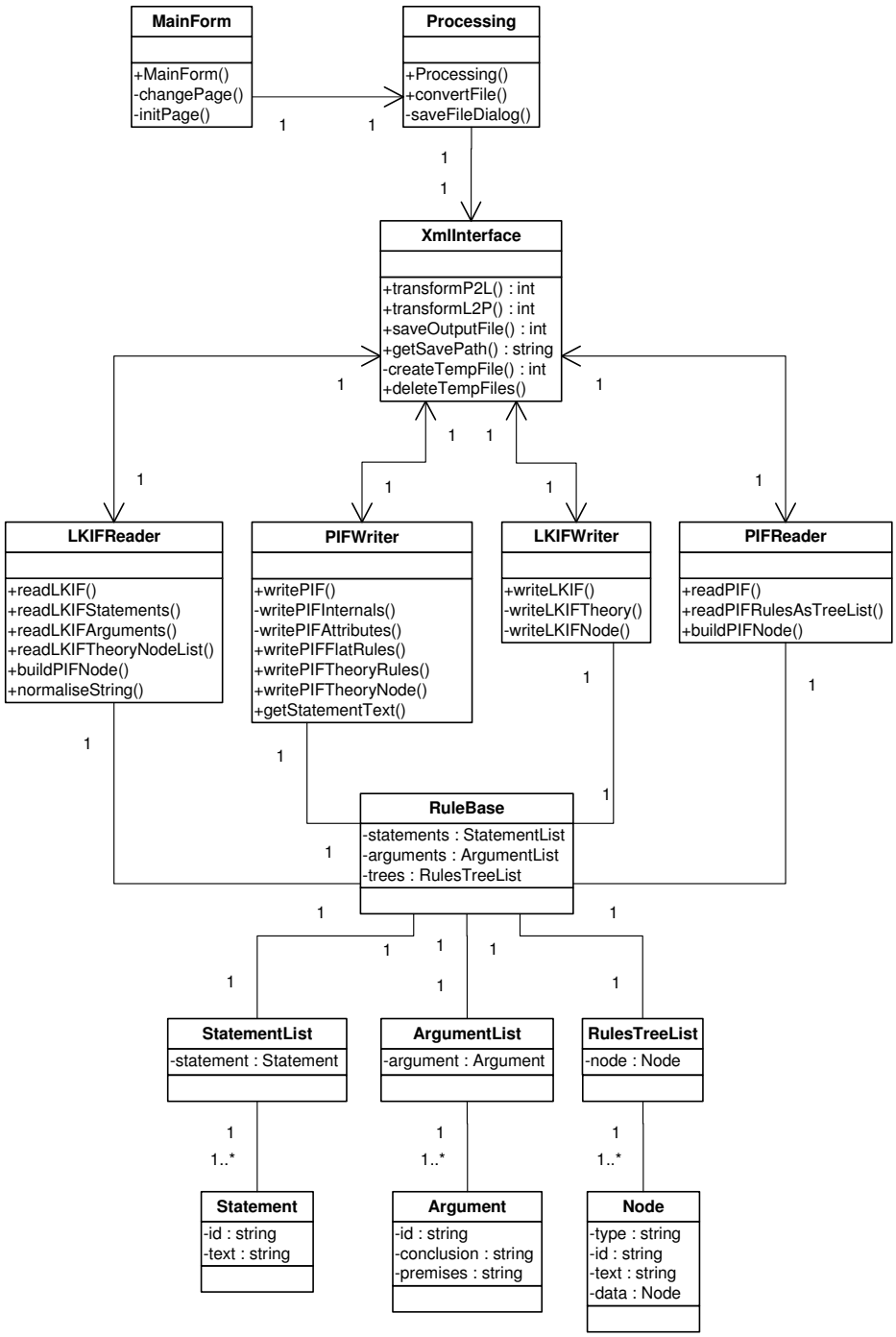
Build Summary

Specification	Detail
Language	Microsoft Visual C# 2008
Environment	Microsoft Visual Studio 2008 9.0
Framework	.NET 3.5

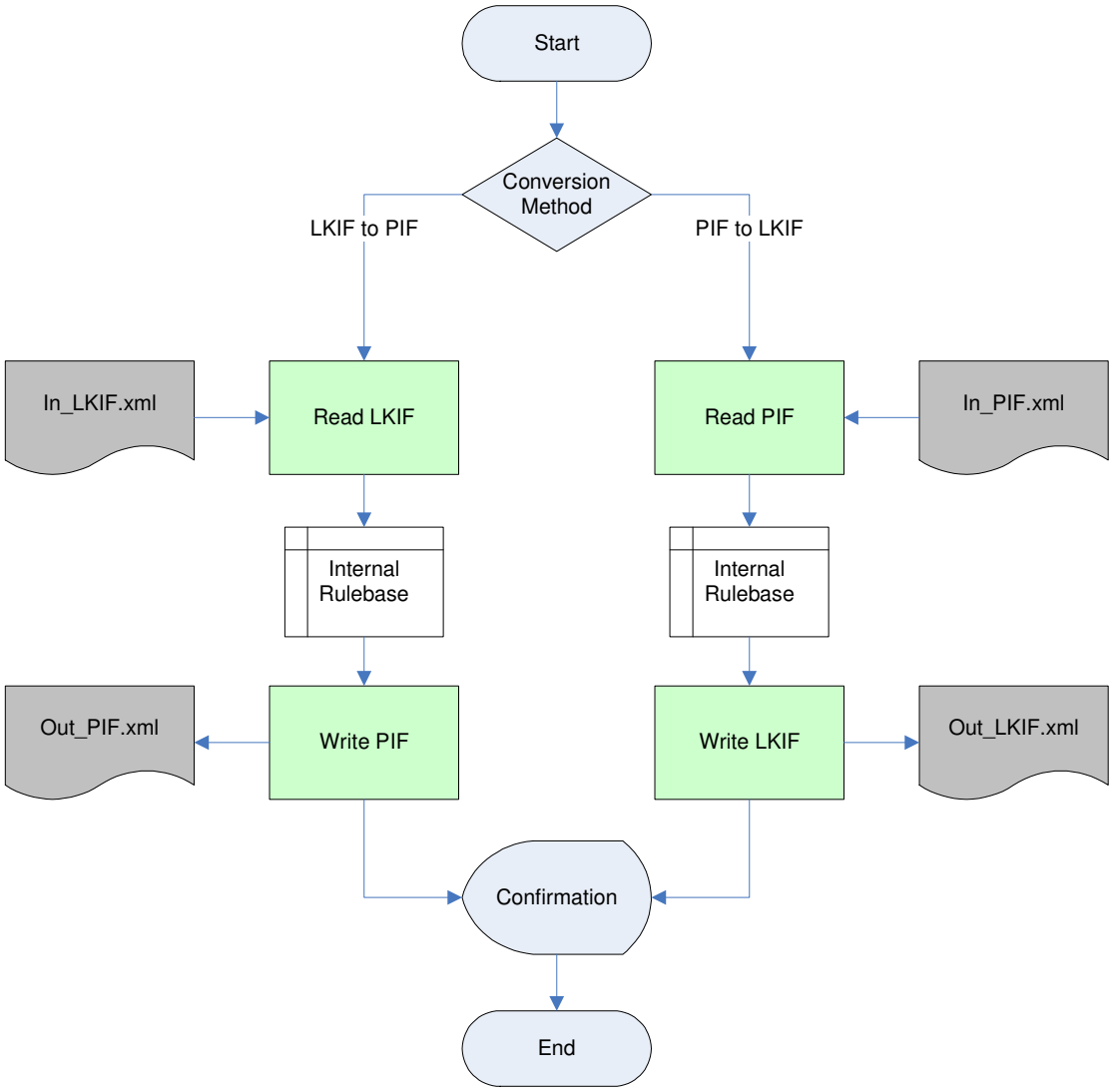
Dependencies

.NET Framework 3.5 http://www.microsoft.com/net/DownloadCurrent.aspx

Class Structure (UML)

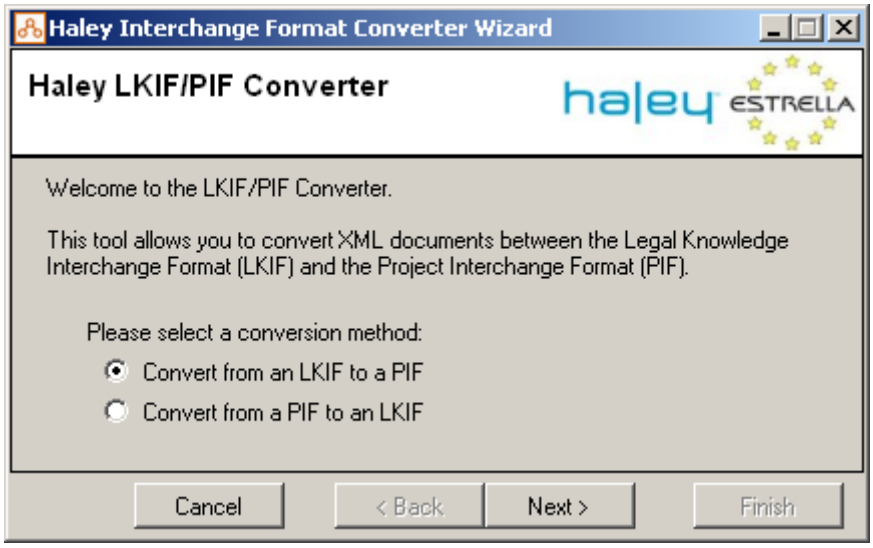


Data Flowchart

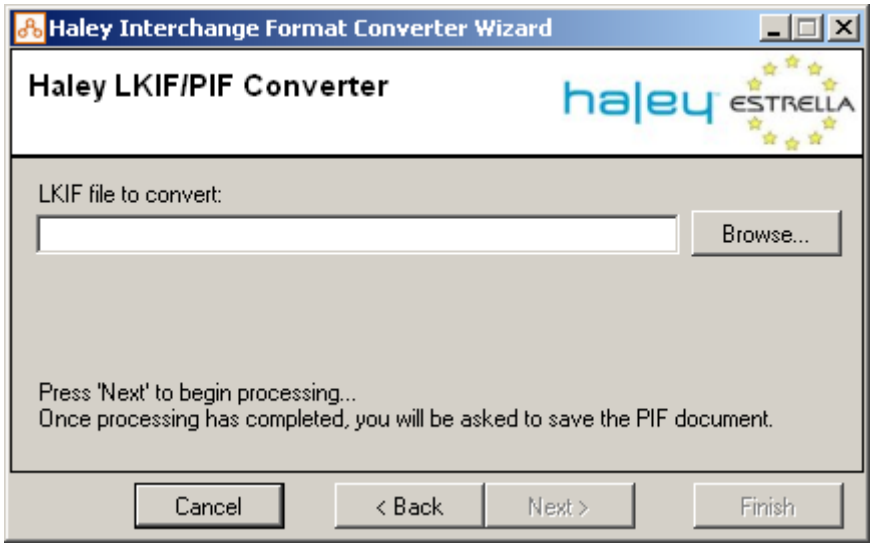


Screenshots

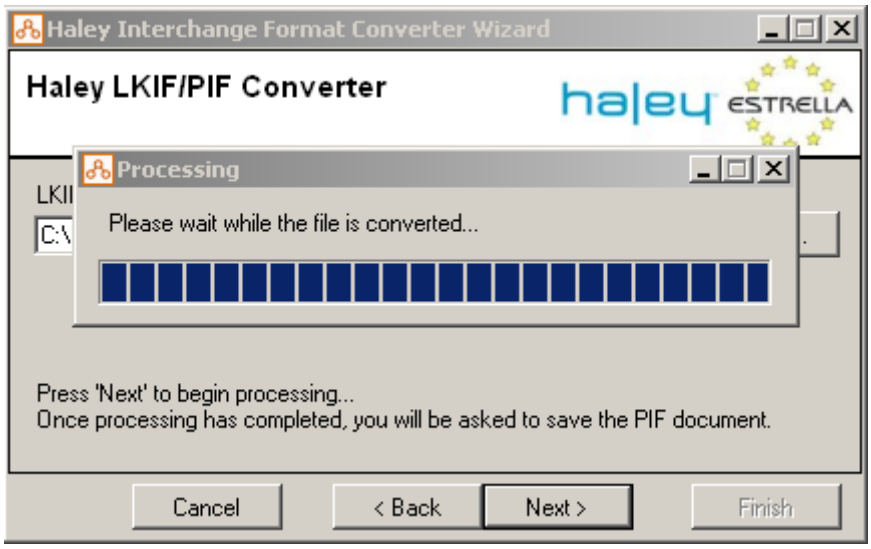
Screen 1 – Select conversion method



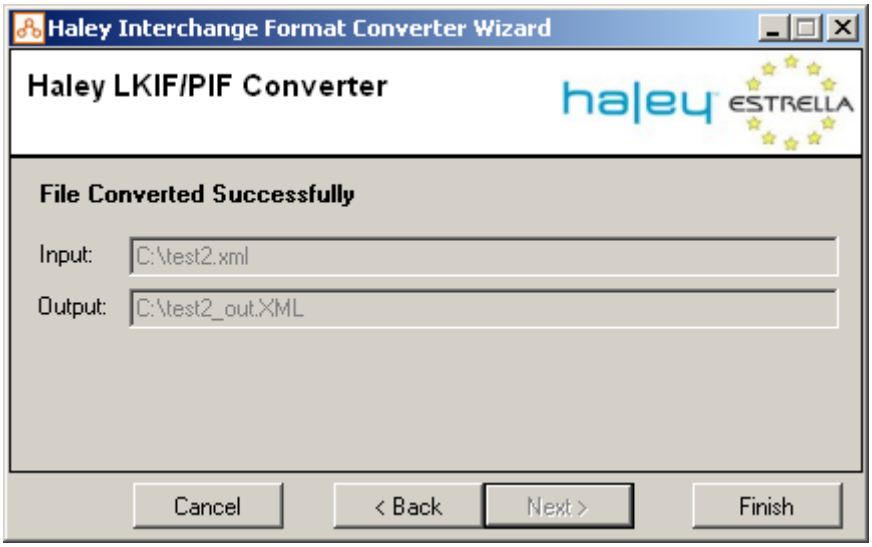
Screen 2 – Choose input file



Screen 3 – Processing



Screen 4 – Successful Finish



Scenarios

S01. Convert from LKIF to PIF

The user has an LKIF file and wishes to import it into RuleBurst Studio. The user must convert the file from LKIF to PIF then import the file using the "import from existing project" option.

S02. Convert from PIF to LKIF

The user has a RuleBurst project and wishes to use it as an LKIF file.

Format Examples

RuleBurst (Rules1.doc)

Person1 is a grandparent of Person2 if

Person1 is a parent of Person2 and
Person2 is a parent of Person3

Person1 is an adult if

Person1's age is more than 70 or
all

Person1 is a parent of Person2
any

Person1 is not happy
Person2 is happy

Person2's age is more than 40

LKIF (Rules1.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<?oxygen RNGSchema="LKIF2.rnc" type="compact"?>
<?xml-stylesheet type="text/css" href="LKIF2.css"?>
<lkif haley:xmlns="http://haley.com/officerules/interchange"
xmlns:haley="http://haley.com/officerules/interchange">
    <theory id="rb1">
        <imports>
            <import uri="http://fokus.fraunhofer.de/elan/estrella/family.owl" />
        </imports>
        <axioms />
        <rules>
            <rule id="r01">
                <head>
                    <s>Person1 is a grandparent of Person2 if</s>
                </head>
                <body>
                    <and>
                        <s>Person1 is a parent of Person2</s>
                        <s>Person2 is a parent of Person3</s>
                    </and>
                </body>
            </rule>
            <rule id="r11">
                <head>
                    <s>Person1 is an adult if</s>
                </head>
                <body>
                    <or>
                        <s>Person1's age is more than 70</s>
                        <and>
                            <s>Person1 is a parent of Person2</s>
                            <or>
                                <s>Person1 is not happy</s>
                                <s>Person2 is happy</s>
                            </or>
                            <s>Person2's age is more than 40</s>
                        </and>
                    </or>
                </body>
            </rule>
            <rule id="r21">
                <head>
                    <s>Rules2 is included in the rulebase if</s>
                </head>
                <body>
                    <and>
                        <s>Rules2 is a rule document</s>
                        <s>Rules2 is not excluded</s>
                    </and>
                </body>
            </rule>
        </rules>
    </theory>
</lkif>
```

Use Cases

U01. LKIF to PIF

The Haley Interchange Format Converter is used to convert from an LKIF to a PIF.

Priority: Essential

Primary Actor: User

Files:

Input LKIF	\Cases\01 - LKIF to PIF\test1_inLKIF.xml
Output PIF	\Cases\01 - LKIF to PIF\test1_outPIF.xml

Success Condition: A new PIF XML file is created including the four rules and attributes for each of the 18 statements.

Trigger: User wizard traversal

Main Sequence

1. Run the program from the Start Menu. Default Location: Start… All Programs… Haley Interchange Format Converter… Haley Interchange Format Converter
2. Choose the "Convert from an LKIF to a PIF" radio button. Press "Next".
3. Select the LKIF file to be converted, by browsing to it or entering its location manually. (test1_inLKIF.xml) Press "Next".
4. Wait a moment for the conversion process to finish. You are now prompted to save the converted file. (test2_outLKIF.xml)
5. The final screen is displayed showing the "File Converted Successfully" message.
6. Press "Finish" to close the program.

Extensions

3a. Input XML is Invalid: Error message displayed, does not progress to save screen.

3b. Input XML is not LKIF: Error message displayed, does not progress to save screen.

3c. Input LKIF contains no data: PIF skeleton created also lacking any data.

4a. Save location is the same as source location: Error message displayed, falls back to input screen.

U02. PIF to LKIF

The Haley Interchange Format Converter is used to convert from a PIF to an LKIF. All functionality is the same as U01, with the input and output formats swapped.

Priority: Essential

Primary Actor: User

Files:

Input PIF	\Cases\02 - PIF to LKIF\test2_inPIF.xml
Output LKIF	\Cases\02 - PIF to LKIF\test2_outLKIF.xml

Success Condition: A new LKIF XML file is created including the four rules and attributes for each of the 18 statements.

Trigger: User wizard traversal

Main Sequence

1. Run the program from the Start Menu. Default Location: Start… All Programs… Haley Interchange Format Converter… Haley Interchange Format Converter
2. Choose the "Convert from a PIF to an LKIF" radio button. Press "Next".
3. Select the PIF file to be converted, by browsing to it or entering its location manually. (test2_inPIF.xml) Press "Next".
4. Wait a moment for the conversion process to finish. You are now prompted to save the converted file. (test2_outLKIF.xml)
5. The final screen is displayed showing the "File Converted Successfully" message.
6. Press "Finish" to close the program.

Extensions

3a. Input XML is Invalid: Error message displayed, does not progress to save screen.

3b. Input XML is not LKIF: Error message displayed, does not progress to save screen.

3c. Input PIF contains no data: LKIF skeleton created also lacking any data.

4a. Save location is the same as source location: Error message displayed, falls back to input screen.

U03. RB Project to PIF to LKIF to PIF to RB Project

The Haley Interchange Format Converter is used to convert from a PIF to an LKIF and back to a PIF. This case's purpose is to show the differences between two RuleBurst projects after undergoing the conversion process.

Priority: Essential

Primary Actor: User

Files:

RuleBurst Rules Document 1	\Cases\03 – Full Circle\01 – ExportedRules1.doc
RuleBurst Rules Document 2	\Cases\03 – Full Circle\01 – ExportedRules2.doc
Input PIF	\Cases\03 – Full Circle\02 – inPIF.xml
Output LKIF	\Cases\03 – Full Circle\03 – outLKIF.xml
Output PIF	\Cases\03 – Full Circle\04 – outPIF.xml
RuleBurst Rules Document 1	\Cases\03 – Full Circle\05 – ImportedRules1.doc

Success Condition: Loading the output PIF into RuleBurst Studio gives a logically equivalent rulebase to that of the original RuleBurst rulebase.

Trigger: RuleBurst Export Wizard, Converter Wizard Traversal, RuleBurst Import Project

Main Sequence

1. From RuleBurst Studio, export the PIF XML. (02 – inPIF.xml)
2. Using the converter, convert from PIF (02 – inPIF.xml) to LKIF (03 -outLKIF.xml).
3. Once the conversion is complete, the user may restart the program or press back until the start of the wizard has been reached.
4. Using the converter, convert from LKIF (03 – outLKIF.xml) to PIF (04 – outPIF.xml).
5. From RuleBurst Studio, create a new project by importing the output PIF.

   (04 – outPIF.xml)

Non-Functional Specifications

NF1. Upgrade

Currently, the converter must be uninstalled from Add/Remove Programs before installing a newer version.

NF2. Security

For a short time, the converted files exist as two temporary files (readTempEst.dat & writeTempEst.dat) within the program folder. If the converter is unexpectedly closed, these files may continue to exist until the next time the converter is run. Care should be taken if using the converter with sensitive data.

NF3. Installation

The converter is installed using a Microsoft self-installer package. It installs the necessary program files, start menu groups, and shortcuts. It can be removed by using Add/Remove Programs from the Control Panel.

Known Issues

1. LKIF: Does not correctly convert 'if, iff, not, exists'
2. LKIF: Negation only works with 'it is not the case that' prefix, does not correctly write the <not> atom
3. No direct PIF counterpart to the LKIF "exclusion" rule. Creates multiply proven attributes
4. PIF: Does not write PIF parse statements (requires full parser)
5. [Resolved 03.07.2008] PIF: Does not read PIF 'both, either' statements
6. PIF: Does not read 'RB-Alternativeconclusion' statements
7. PIF: Does not allow PIF intermediates (sub-premises)
8. [Resolved 03.07.2008] PIF: Does not read rulebases that include excel tables
9. PIF: Does not read excel tables
10. Potential issue if missing elements. Problem if valid xml, but not containing all the elements needed for reading
11. XML Reader/Writer may not always safely close. Have not fully exhausted testing of unexpected termination
12. Error checking does not distinguish between types of input errors (file vs. xml)
13. PIF importer can in some cases import an LKIF without creating an error