



D6.9

Usability and Interaction Design (3)

Document Identification	
Date	30.11.2019
Status	Final
Version	Version 1.0

Related WP	WP 6	Related Deliverable(s)	D6.3, D6.4
Lead Authors	Stephanie Weinhardt	Dissemination Level	PU
Lead Participants	USTUTT	Contributors	TUG
Reviewers	FHG		

This document is issued within the frame and for the purpose of the LIGHT^{est} project. LIGHT^{est} has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No 700321.

This document and its content are the property of the *Lightest* Consortium. All rights relevant to this document are determined by the applicable laws. Access to this document does not grant any right or license on the document or its contents. This document or its contents are not to be used or treated in any manner inconsistent with the rights or interests of the *Lightest* Consortium or the Partners detriment and are not to be disclosed externally without prior written consent from the *Lightest* Partners.

Each *Lightest* Partner may use this document in conformity with the *Lightest* Consortium Grant Agreement provisions.

Document name:	Usability and Interaction Design (3)	Page:	1 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



1. Executive Summary

The goal of this deliverable is to show how the interaction concept (see D6.4) was implemented in the TPAT and evaluated one more time. The main work of the interaction concept was done in the first two years of the project.

The first iteration (covered in D6.3) shows the first interaction concept that was visualized with mock-ups and evaluated in the first usability evaluation with end users. Based on the results from the first iteration, the interaction concept was improved and implemented into a high-fidelity prototype. This high-fidelity prototype was the objective of the second elaborate usability study with end users. The results from the usability study were then integrated in a revised interaction design (see D6.4).

The focus of the third concept iteration was to make sure that the interaction concept was implemented in the software as well as providing support to the developers with arising issues that could not be addressed before. Therefore, this deliverable shows, how the interaction concept was implemented (see chapter 5) and provides a third usability evaluation using a heuristic expert walkthrough (see chapter 7).

Document name:	Usability and Interaction Design (3)	Page:	2 of 46		
Dissemination:	PU	Version:	Version 1.0	Status:	Final



2. Document Information

2.1 Contributors

Name	Partner
Stephanie Weinhardt	USTUTT
Lukas Alber	TUG

2.2 History

Version	Date	Author	Changes
0.1	03/09/2019	Stephanie Weinhardt	First version
1.0	30/11/2019	Stephanie Weinhardt	Final version

Document name:	Usability and Interaction Design (3)	Page:	3 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



3. Table of Contents

1. Executive Summary	2
2. Document Information	3
2.1 Contributors	3
2.2 History	3
3. Table of Contents	4
3.1 Table of Figures.....	6
3.2 Table of Acronyms.....	6
4. Summary of previous Deliverables	7
4.1 D6.3 – Usability and Interaction Design	7
4.2 D6.4 – Usability and Interaction Design	9
5. Implementation in the Trust Policy Authoring Tool	11
5.1 Getting started.....	11
5.2 Graphical Editor.....	13
5.3 Natural Language Layer	15
6. Results from D6.4 User Study	25
6.1 Improvements in the implementation	26
6.1.1 Usability Issue 1: Saving Trust Policies.....	26
6.1.2 Usability Issue 2: Adding Building Blocks.....	28
6.1.3 Usability Issue 3: Syntax of statement.....	29
6.1.4 Usability Issue 4: Confirm changes to building blocks	30
7. Usability Evaluation	31
7.1 Methodology.....	31
7.1.1 Collection of Usability Heuristics	32
7.2 Results	33
7.2.1 Not Applicable.....	33
7.2.1.1 General	34
7.2.1.2 Graphical Editor.....	34
7.2.1.3 Natural Language Layer.....	34
7.2.2 Neutral	34
7.2.2.1 General	34
7.2.2.2 Graphical Editor.....	34
7.2.2.3 Natural Language Layer	35
8. References	36
9. Project Description	38
10. Appendix	40

Document name:	Usability and Interaction Design (3)	Page:	4 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



10.1	Results from expert walkthrough of general aspects of the TPAT	40
10.2	Results from expert walkthrough of the Graphical Editor.....	42
10.3	Results from expert walkthrough of the Natural Language Layer	44

Document name:	Usability and Interaction Design (3)	Page:	5 of 46		
Dissemination:	PU	Version:	Version 1.0	Status:	Final



3.1 Table of Figures

Figure 1: Interaction of layers in the Trust Policy Authoring Tool.....	8
Figure 2 TPAT with no existing Trust Policy	11
Figure 3 Dialogue to select the layer in which the users want to create their trust policy in	12
Figure 4 Dialogue to name the trust policy.....	13
Figure 5 New Trust Policy using the Graphical Editor	13
Figure 6 Graphical Layer with two added Trust Schemes	14
Figure 7 Graphical Layer - Trust Scheme Selection	15
Figure 8 Newly created Trust Policy in the Natural Language Layer.....	16
Figure 9 Natural Language Layer with one added building block	17
Figure 10 Natural Language Layer with a second building block added to the statement	18
Figure 11 Natural language layer with several building blocks.....	19
Figure 12 Natural Language Layer with several building blocks (2).....	20
Figure 13 Natural Language with Layer several building blocks (3).....	21
Figure 14 Natural Language Layer with several building blocks (4).....	22
Figure 15 Natural Language Layer with several building blocks (5).....	23
Figure 16 Natural Language Layer: first statement correct, second statement added	24
Figure 17 Automatic Saving within the TPAT.....	27
Figure 18 Undo-function after deleting a trust policy.....	28
Figure 19 Adding a building block to a statement in the Natural Language Layer	29
Figure 20 Syntax aid message: Example 1	29
Figure 21 Syntax aid message: Example 2	29
Figure 22 Syntax aid message: Example 3	30
Figure 23 Syntax aid message: Example 4	30
Figure 24 Syntax aid message: Example 5	30
Figure 25 Syntax aid message: Example 6	30
Figure 26 Syntax aid message: Example 7	30
Figure 27 Syntax aid message: Example 8	30
Figure 28 Syntax aid message: Example 9	30

3.2 Table of Acronyms

ATV	Automatic Trust Verifier
GE	Graphical Editor
N	Nominations
NL	Natural Language
UI	User Interface
UX	User Experience
TP	Trust Policy
TPAT	Trust Policy Authoring Tool
TPL	Trust Policy Language



4. Summary of previous Deliverables

4.1 D6.3 – Usability and Interaction Design

The goal of *D6.3 Usability and Interaction Design* was to establish the first interaction concept for the LIGHT^{est} tools. The deliverable gave an overview of the related literature as well as usability principles and design suggestions important for the design of the LIGHT^{est} tools. Usability requirements were being defined for each tool and based on those, user stories and user interface guidelines were formulated. On the basis of those stories and guidelines the first low-fidelity prototype was created.

The three-layer concept

In *D6.3 Usability and Interaction Design* the three-layer approach was established as a technical frame condition. The Trust Policy Authoring Tool will create and edit trust policies in the context of LIGHT^{est} (it can also be used outside of LIGHT^{est} to create trust policies). The tool must be usable to a wide range of different types of users with different levels of knowledge. This brought up the decision in LIGHT^{est} D6.1 (LIGHTest, 2017) to use the tool for creating trust policy at different layers of complexity.

The concept expects that three categories will use the Trust Policy Authoring Tool.

The first set of users are those that have no technical background, no programming skills and no experience in using such tools. We call such users further in the text beginner users. The second user group, also called intermediate users, are supposed to have little to intermediate programming skills and no to little knowledge of such tools. The last user groups, also called expert users, are supposed to have good to expert programming skills and intermediate to expert knowledge in such tools. The needs of each user group are met in a different, so called, layer:

The lowest layer is the Trust Policy Language (TPL) layer which will be used by expert users. This layer provides the most power and freedom in specifying a trust policy but it has a steep learning curve. The next layer above the TPL layer is the Natural Language Layer. This layer is less complicated compared to the lowest layer but is still complex and might have a steep learning curve, therefore considered to be used by intermediate users. It offers less power and freedom compared to the TPL layer but more than the next layer which is the graphical language layer. The graphical language layer is the next and highest layer of abstraction in the tool. The graphical layer is intended for the beginner users and it does not require a lot of training to start using it to write basic trust policies for electronic transactions. However, out of the three layers, this layer provides the least freedom and flexibility. Most of the policies can be designed in this layer but really complex policies have to be written at a lower layer.

Document name:	Usability and Interaction Design (3)	Page:	7 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



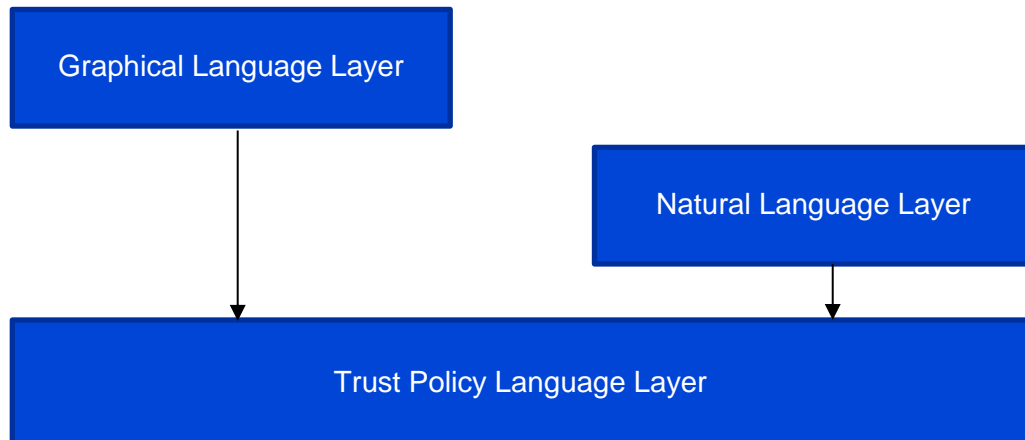


Figure 1: Interaction of layers in the Trust Policy Authoring Tool

On the basis of the description given above, the tool has the following technical limitations:

- The tool will not be able to translate a policy created in the graphical language format to the natural language format.
- The tool will only translate a policy created in the graphical language to the trust policy language format
- The tool will only translate a policy created in the natural language format to the trust policy language format
- The tool will not translate a policy created in the trust policy language format to the graphical language format
- The tool will not translate a policy created in the trust policy language format to the natural language format

A low-fidelity prototype was implemented to visualize the first interaction concept. The following briefly touches the results from the first evaluation using the low-fidelity prototype.

Summary of usability evaluation results

With the Low-Fidelity Prototype a usability test was conducted. As this was a large part of the deliverable D6.3, only those results relevant for this deliverable will be mentioned.

Overall, the navigation through the tool was understandable and straight-forward. Tasks like creating, editing or deleting an existing Trust Policy were easy to accomplish for the users.

The main problem occurred when users had to create their own Trust Policy. Despite the fact that it was specified in detail what they should express, the overall concept and principle behind it was cognitively overloading the users. Interaction itself did not cause problems, but some

Document name:	Usability and Interaction Design (3)	Page:	8 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



users mentioned that they were missing help positions or more information, because they were just reproducing what they were told without knowing if this made sense or is of high quality.

As at this stage of the project the detailed design of the policy creation was not defined and there was still major work required within this task, we concluded the results as positive.

For the overall interaction concept this meant that no major changes needed to be made.

4.2D6.4 – Usability and Interaction Design

D6.4 goes into detail of the design and implementation of the revised interaction concept in a high-fidelity prototype using axure¹. The main purpose of the prototype was to evaluate the overall interaction concept and especially the creation of trust policies in the Graphical Layer and Natural Language Layer. Thus, not all possible functions are implemented in the prototype.

With this prototype an elaborate user study regarding its usability and user experience was conducted.

The deliverable goes into detail with the methodology used in the study as well as its results. Based on the results of the study, a revised and more detailed interaction concept was defined and documented in D6.4. The following provides a short overview over the used methodology and the next chapter gives an overview of the results.

Methodology

The evaluation with the high-fidelity prototype based on HTML-files took place in a laboratory setting. 18 participants took part in the evaluation. The overall study goal was to get formative and summative feedback on the basis of the evaluated prototype. With the existing prototype, the LIGHTest client's functionality of the envisioned system could be presented in an interactive way and evaluated ideally for usability outcomes. In addition, we introduced a User Experience measuring questionnaire – the User Needs Questionnaire (UNeeQ). The UNeeQ measures the extent to which a set of ten basic human needs is addressed and how the overall user experience went using the product. It measures that by letting participants rate their accordance with predefined sentences on a five level Likert scale from “not at all” to “highly”. As these measurements are highly subjective and depend on the current mood of the participants, they had to fill out the UNeeQ right before and directly after interacting with the prototype. Thus, the data could be compared as we measured the users' mindset before the interaction and how the interaction changed it. In combination with the follow-up interviews we expected to find out which needs exactly are addressed when using the product.

¹ www.axure.com/

Document name:	Usability and Interaction Design (3)	Page:	9 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



The study was conducted in September 2018 and consisted of three parts:

- Preliminary interview
- UNeeQ
- Testing phase
- UNeeQ
- Follow-up interview

The preliminary interview, which can be found in the appendix, covers some basic questions to get to know the participants a little bit better. The participants were asked if they ever created some sort of policy before and if yes, which tools they used. This question was necessary to clarify the background of the participants in this topic. Experience with different tools or with creating policies in general could influence the test results. Then they were asked to rank their coding skills and name their job title. As the task given in the evaluation is somewhat coding related, these questions could offer some insights about the experience the participants had with coding or in this topic in general. The last two questions were about physical limitations and if the participants would consider themselves technophile. The preliminary interviews closed with an explanation about LIGHT^{est}, the purpose of the evaluation and the test settings.

After the preliminary interview, the moderator provided the participants with the necessary information and asked them to comment or ask questions at any time. Also the participants were asked to think aloud during the test phase. According to the typical usability testing method, in each test session the participants were instructed to follow a predefined scenario and procedure. The participants were given one task to execute during which they were observed and notes were taken of their behaviour. After finishing the first task, the same methodology was applied for the second task.

The two tasks applied in the usability test are “Trust Policy Creation in Graphical Editor” and “Trust Policy Creation in Natural Language”.

The evaluation was wrapped up with a follow-up interview, with which the subjective view about positive and negative aspects of the interaction with the prototype were retrieved. Those questions complemented the objective observations of the moderator, with subjective test data.

A detailed listing of the overall tasks and interviews used in the tests can be found in D6.4.

Document name:	Usability and Interaction Design (3)	Page:	10 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



5. Implementation in the Trust Policy Authoring Tool

This chapter gives a brief overview of the implementation of the trust policy authoring tool (TPAT) with a main focus on the interaction concept and the improvements made regarding the identified usability issues in the user study and the interaction concept that was developed after.

The implementation is different from the interaction concept stated in D6.4 as there were technical limitations that had to be considered as well as new project results/decisions that lead to a different set of functionalities.

Below, the concept is shown in screenshots.

5.1 Getting started

Figure 2 shows the TPAT when the users open the tool for the first time.

In the middle section, a short explanation gives a hint to the users to how to create their first trust policy.

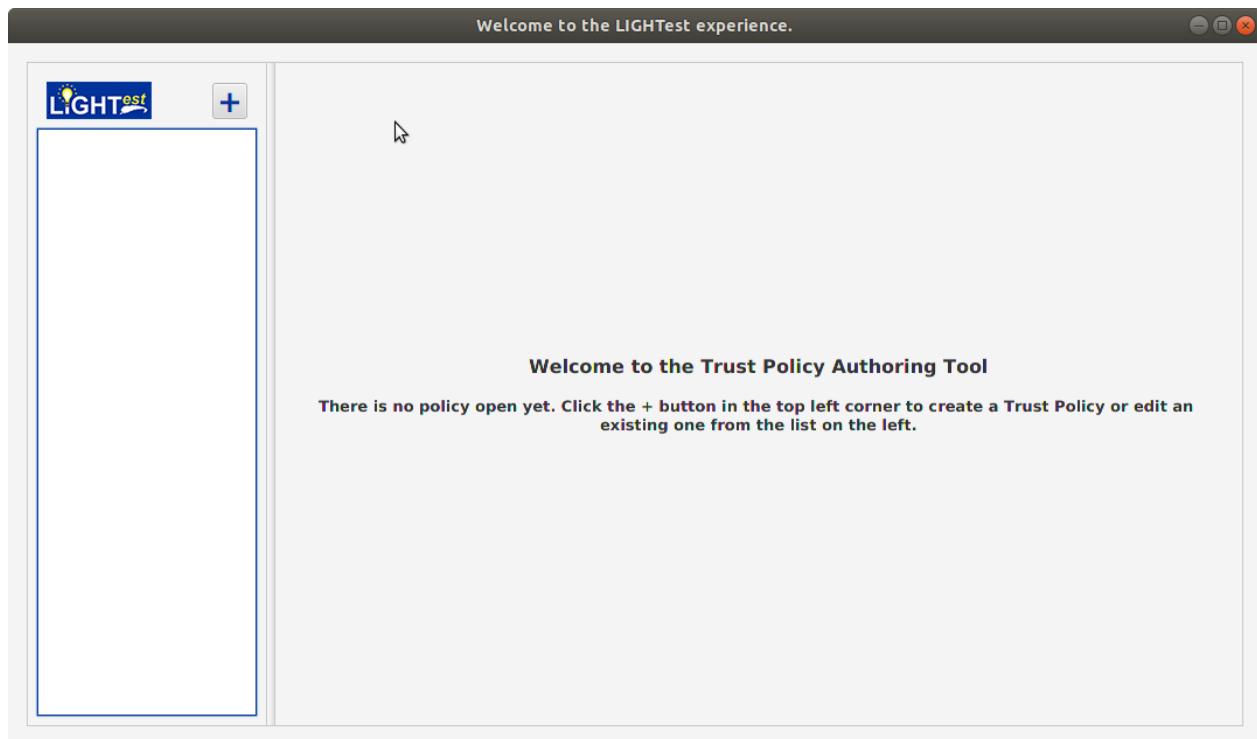


Figure 2 TPAT with no existing Trust Policy

By clicking on the “+” icon in the upper left corner the users can create a new Trust Policy.

Figure 3 shows the step after clicking the “+” button. The dialogue displays the choices that the users have. They can create a trust policy using either the graphical editor, the natural language

Document name:	Usability and Interaction Design (3)	Page:	11 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



layer or the trust policy layer. For a detailed description of this concept please see D6.3 and D6.4.

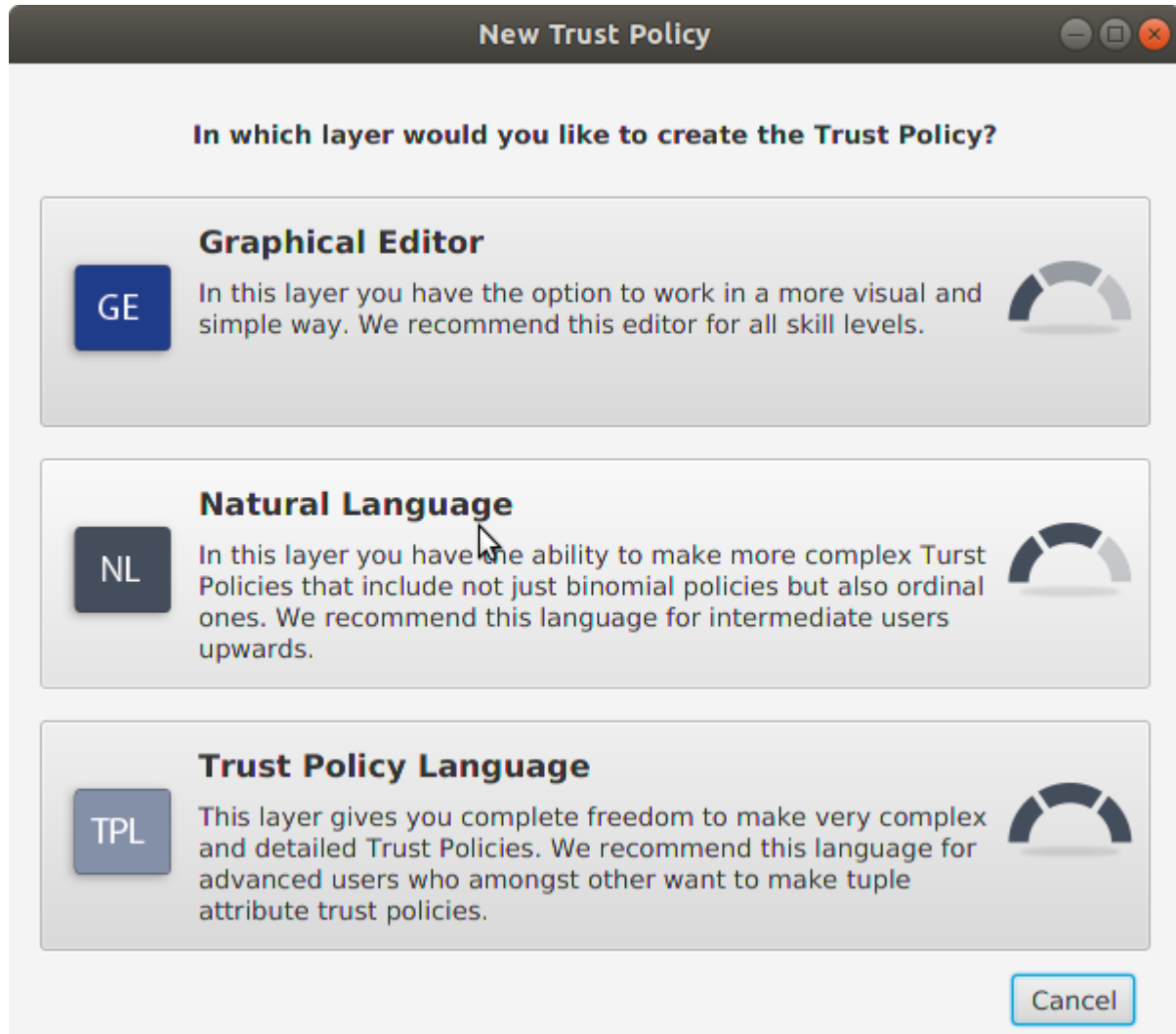


Figure 3 Dialogue to select the layer in which the users want to create their trust policy in

After users select the layer in which they want to create their trust policy in, they can name the trust policy. This helps to identify them later. Figure 4 shows this dialogue screen.

Document name:	Usability and Interaction Design (3)	Page:	12 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



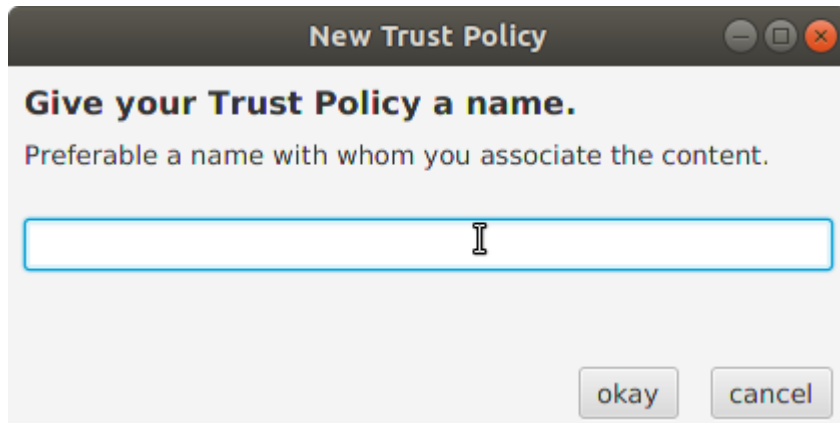


Figure 4 Dialogue to name the trust policy

After the trust policy is named, the dialogues close and the newly created Trust Policy is displayed in the list on the left side whereas the previous selected layer is shown in the middle part of the screen.

5.2 Graphical Editor

Figure 5 shows a newly created Trust Policy with the Graphical Editor.

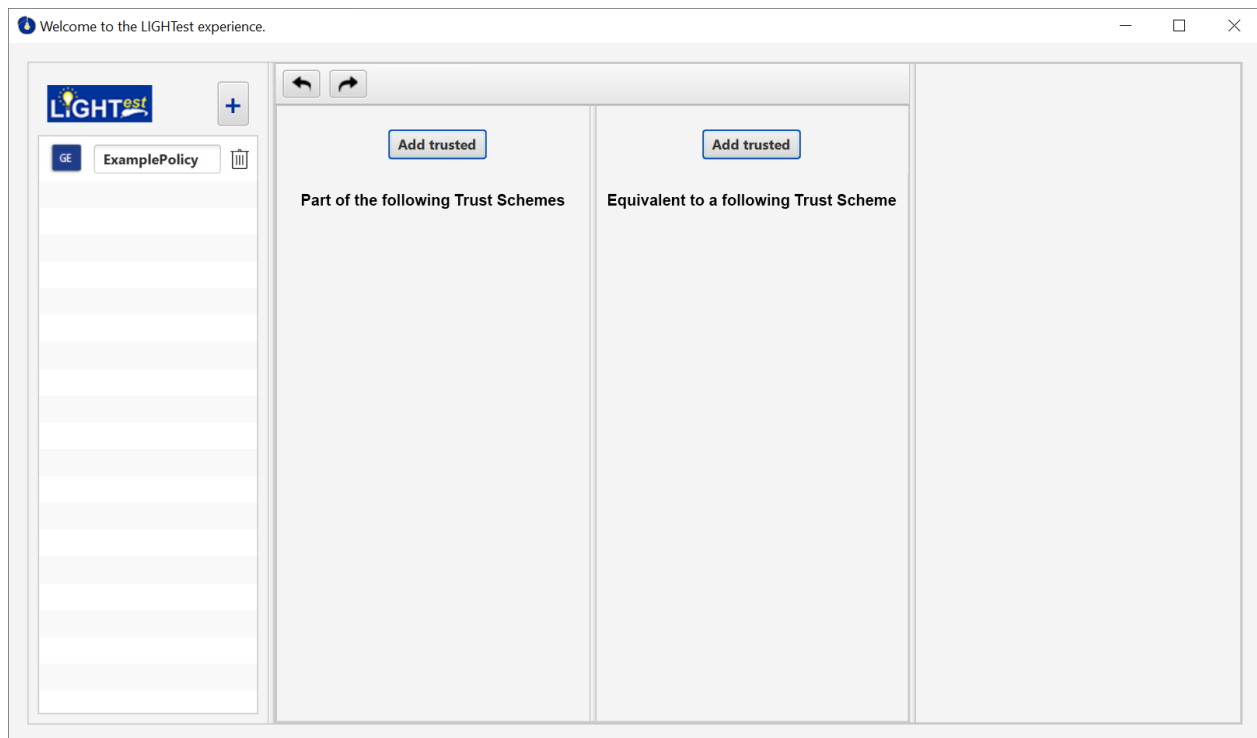


Figure 5 New Trust Policy using the Graphical Editor

The middle part is split into two columns. Users are able to add Trust Policies to the left side to state that they trust an entity if it is a part of a certain trust scheme.

Document name:	Usability and Interaction Design (3)	Page:	13 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



Adding a trust scheme to the right side makes it possible to have easy translations within the graphical editor, as this means that users trust an entity if it is equivalent to a trust scheme that is added to the right side.

Considering the Graphical Editor this is the biggest change in the concept between the concept stated in D6.4 and the implementation of it. The interaction concept in D6.4 foresaw that users can add Trust Schemes to the left side that they trust and add Trust Schemes to the right that they do not trust. During the implementation phase, the differentiation between those two columns became a bigger issue. If a user adds something to the “I trust the Trust Scheme” side – this automatically makes all the other Trust Schemes not trustworthy and therefore the right column becomes obsolete. The first solution was to just provide users with one column – the “Trust Schemes I trust” column. Nevertheless, after trying different options we decided to add more functionality to this layer – the translation.

Figure 6 shows the Graphical Layer with one Trust Scheme added to each side.

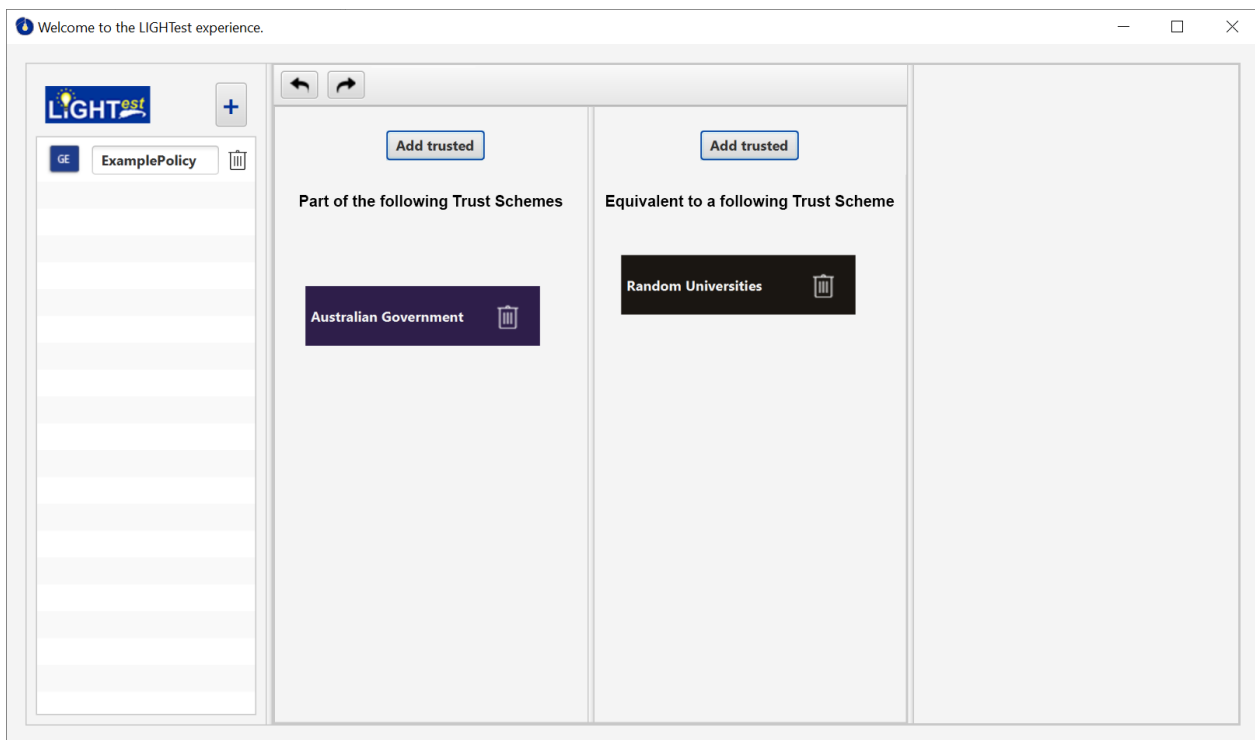


Figure 6 Graphical Layer with two added Trust Schemes

Figure 7 shows the right column. If users add a Trust Scheme or want to change an existing one, they can select a Trust Scheme from the list on the right or use the search bar above the list.

By clicking on a Trust Scheme the building block in the middle immediately changes its content to the selected Trust Scheme.

Document name:	Usability and Interaction Design (3)	Page:	14 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



Therefore, users do not have to explicitly click on a “Select” button as the concept in D6.3 and D6.4 foresaw. This change was made due to the usability issues identified in D6.4.

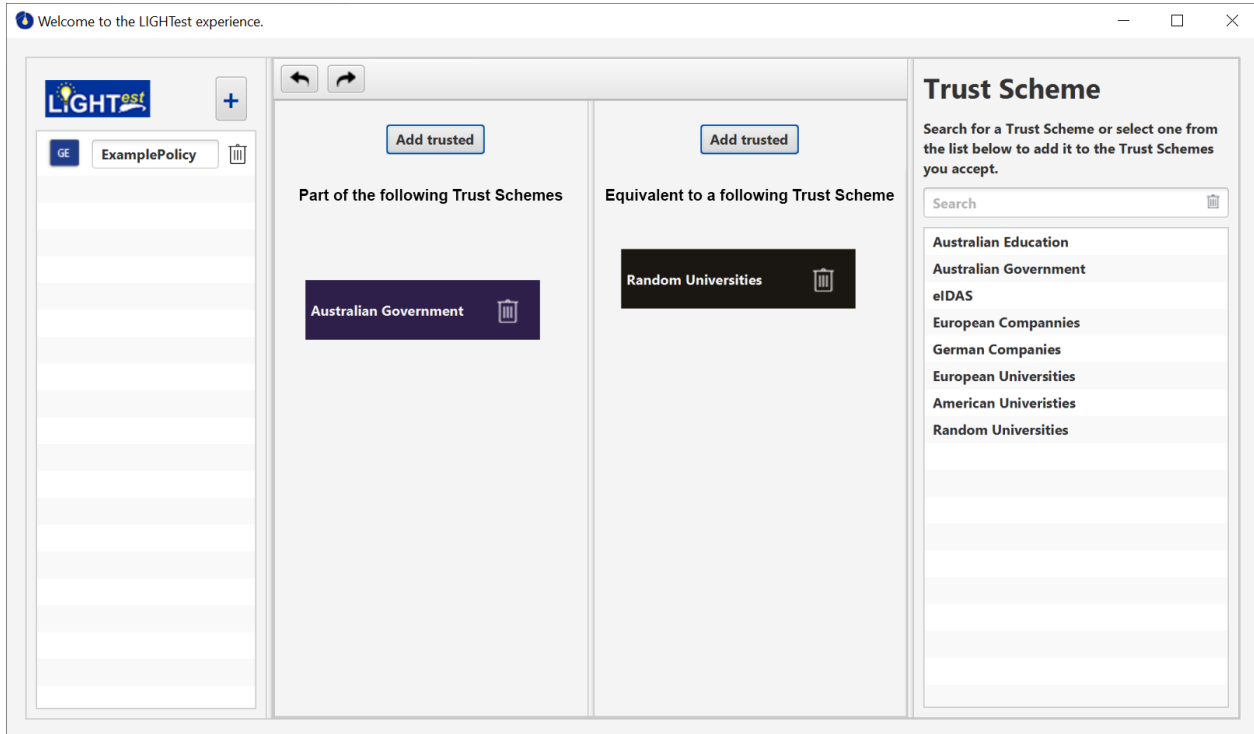


Figure 7 Graphical Layer - Trust Scheme Selection

5.3 Natural Language Layer

If users choose to create a Trust Policy using the Natural Language Layer, the middle part will be displayed as shown in Figure 8. On the top, five different building blocks are displayed. Those can be used to create statements for the Trust Policy.

Document name:	Usability and Interaction Design (3)	Page:	15 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



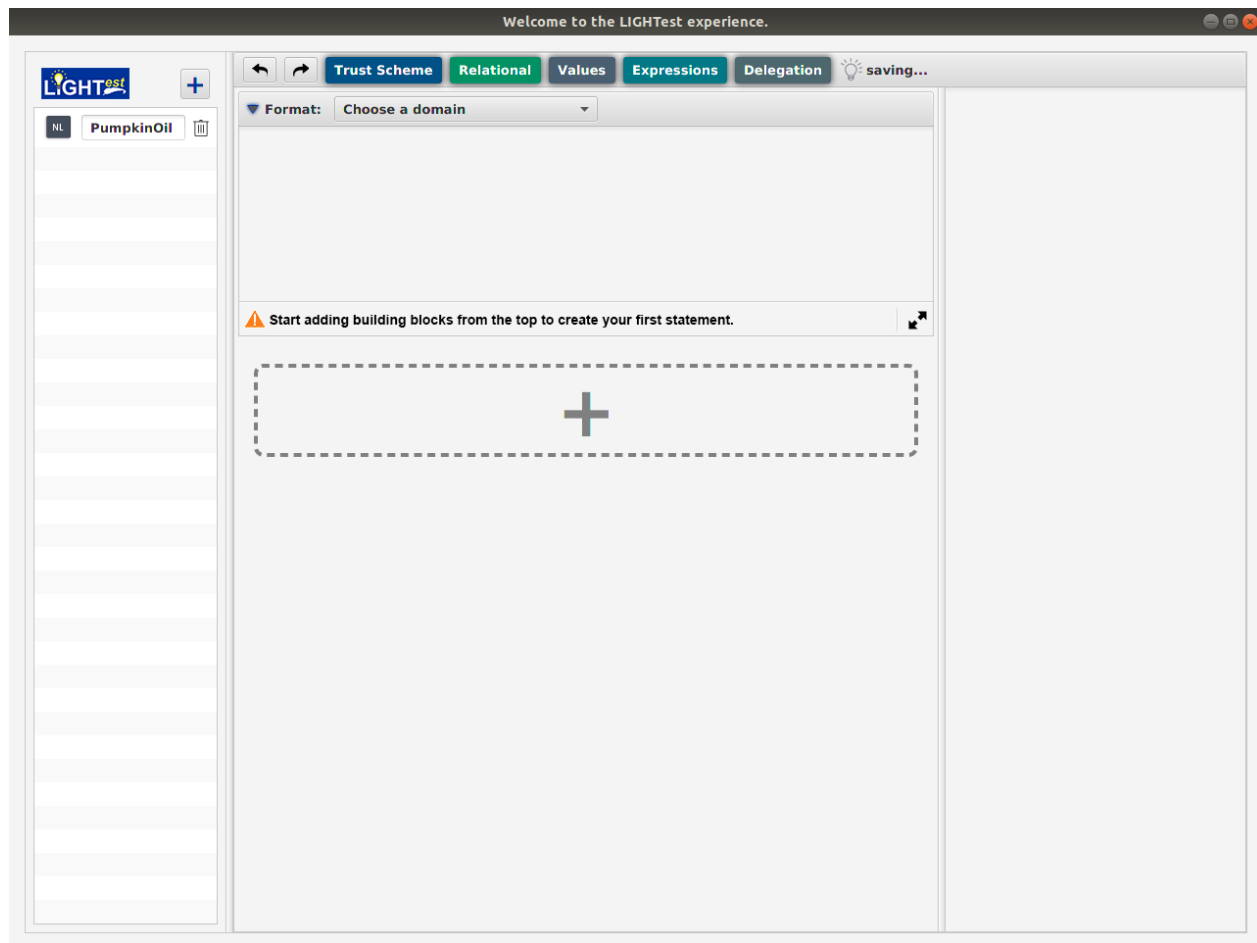


Figure 8 Newly created Trust Policy in the Natural Language Layer

By clicking on one of the building blocks, it is directly added to the statement beneath. The interaction concept described in D6.4 foresaw that users could add a building block to a statement by using drag and drop. By clicking on the building block users would get a preview of the content of the building block on the right side. This was changed as it caused some usability issues. Users did not intuitively drag and drop the building blocks – they stated the wish to select them, change the content of them on the right side of the screen and then add it to the statement.

After discussion, we chose the option of directly adding the building block to the statement and displaying the content of the building block on the right side. This is shown in Figure 9.

Document name:	Usability and Interaction Design (3)	Page:	16 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



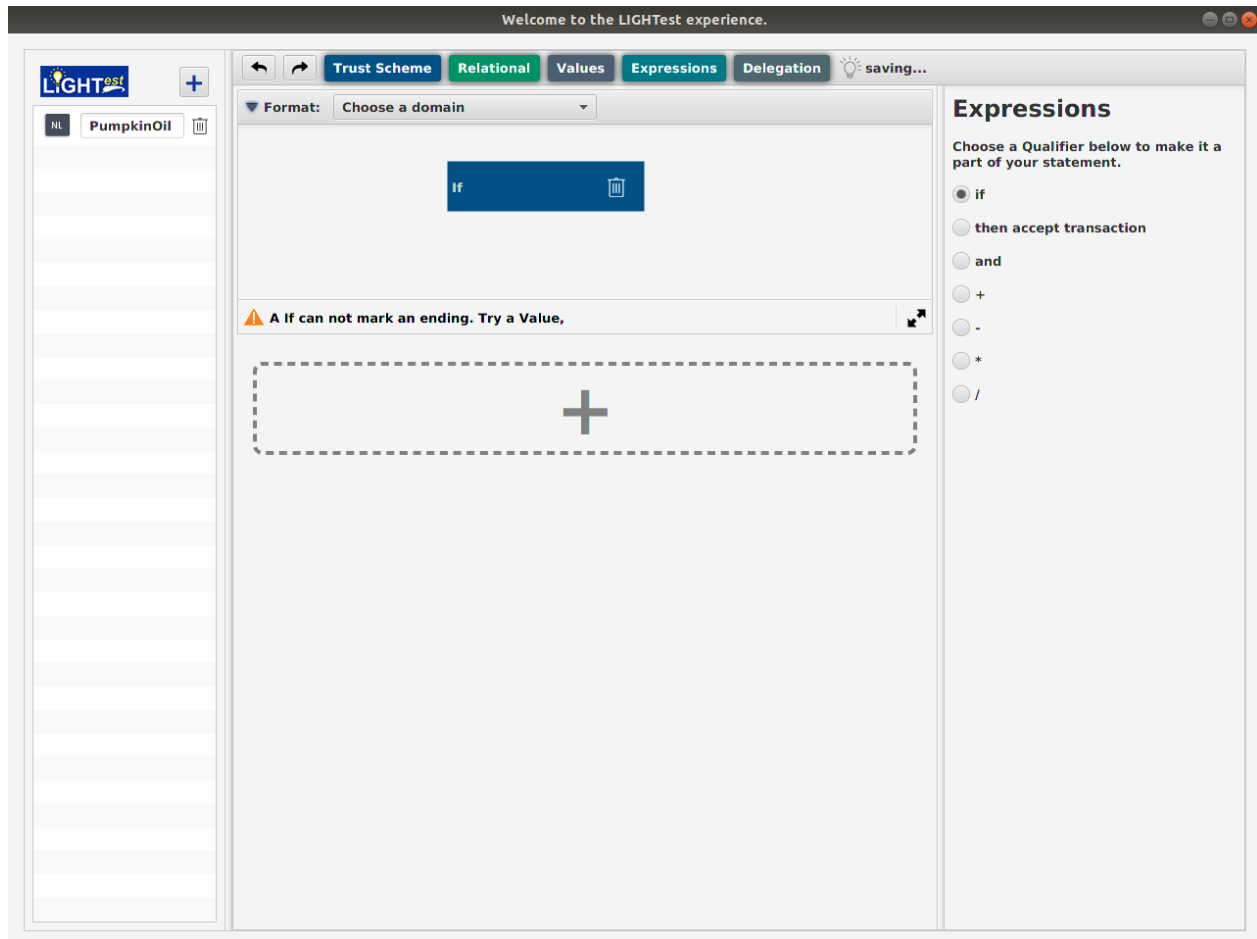


Figure 9 Natural Language Layer with one added building block

Although users would have to click another time to delete an accidentally selected building block, we went with this solution as it was the easiest and most intuitive way to add a building block. User click and get an immediate feedback on what happened and how it works, as the building block directly appears in the middle and the right side shows its content.

For a detailed description of the building blocks and its contents please see D6.3 and D6.4.

On the bottom of a statement the users find messages that are going to help them create an actually working Trust Policy.

Under the first statement the users are able to add as many statements as they want. This can help them organize the content of the Trust Policy and break it in smaller parts rather than having one long list of building blocks.

Document name:	Usability and Interaction Design (3)	Page:	17 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



With the help of the “two arrow” icon in the right corner on the bottom of a statement users are able to resize the statement as they need it. As users add more and more building blocks to the statement the statement should resize itself (if a user does not purposely downsize it).

Figure 10 shows the statement with another building block added. The building blocks attach themselves automatically to the bottom of the previous building block, to make a list. Users are able to rearrange the building blocks at any time.

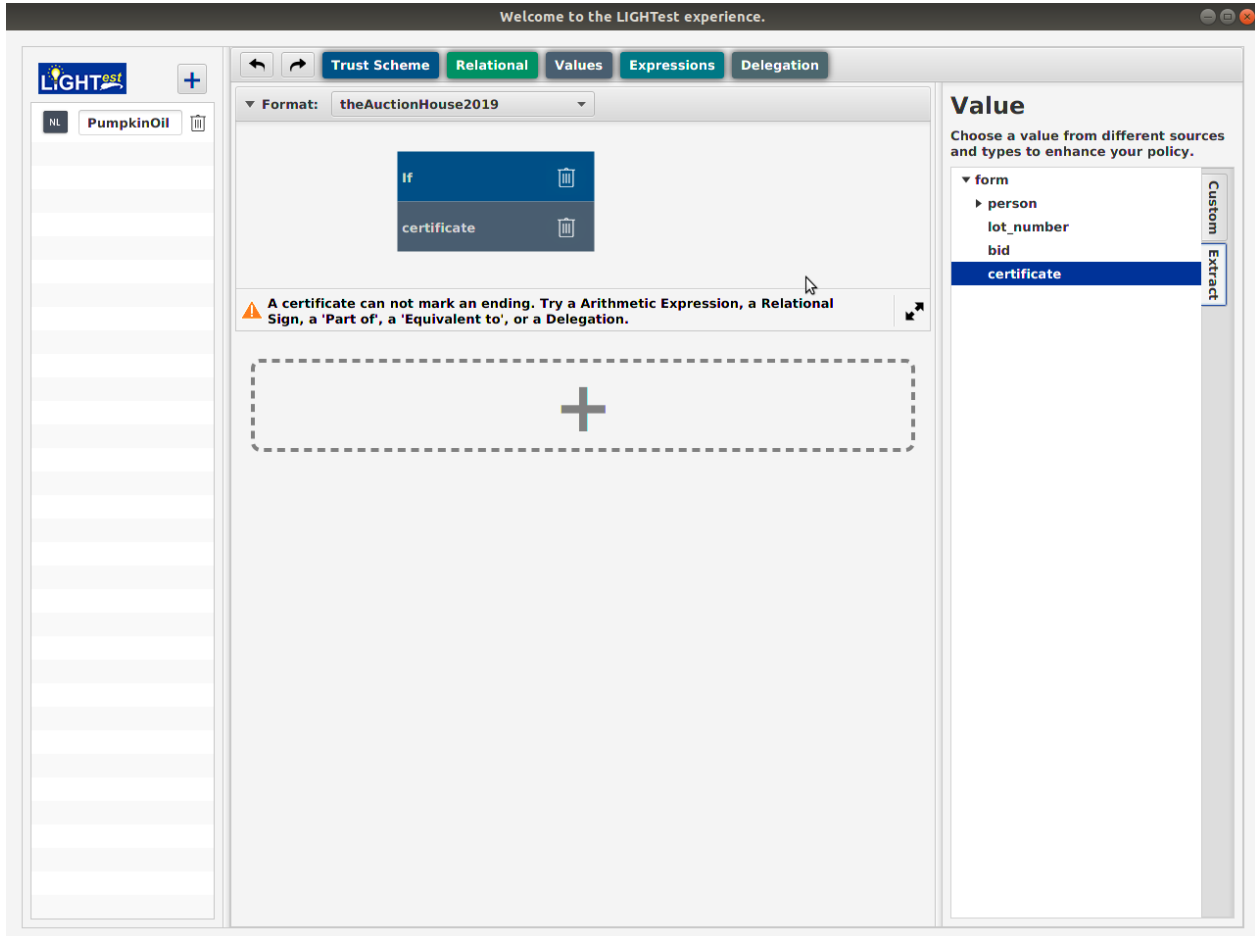


Figure 10 Natural Language Layer with a second building block added to the statement

Nevertheless, in the end the building blocks have to be docked to each other. This is due to technical reasons, as the parser needs certainty with the order of the building blocks to be able to process them in the correct manner.

Figure 11 shows several building blocks being added to the statement and two building blocks being not yet attached to the list.

Document name:	Usability and Interaction Design (3)	Page:	18 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



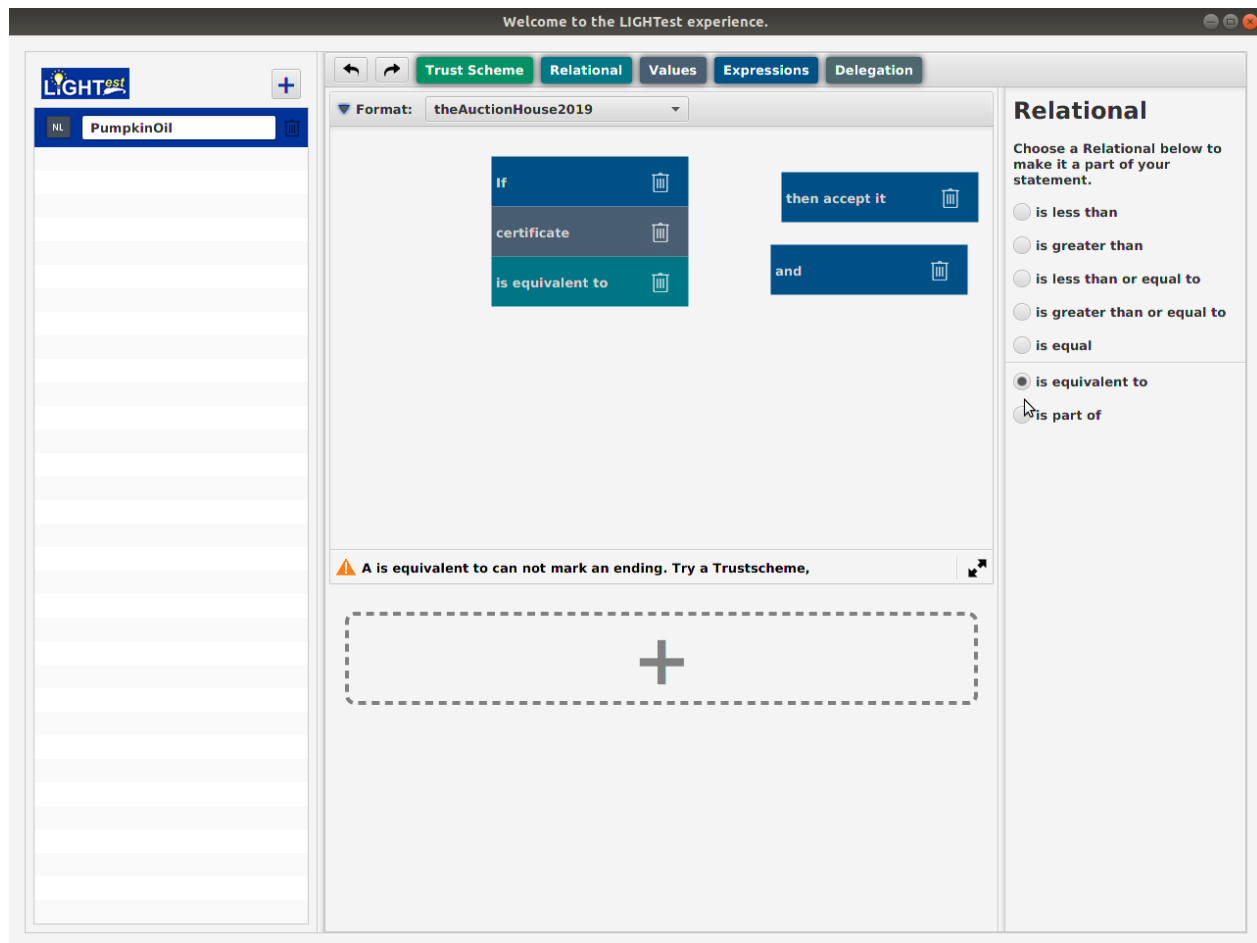


Figure 11 Natural language layer with several building blocks

The figures Figure 11 Figure 15 are showing different types of messages/warnings that can come up within a statement to guide users to a working Trust Policy.

Document name:	Usability and Interaction Design (3)	Page:	19 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



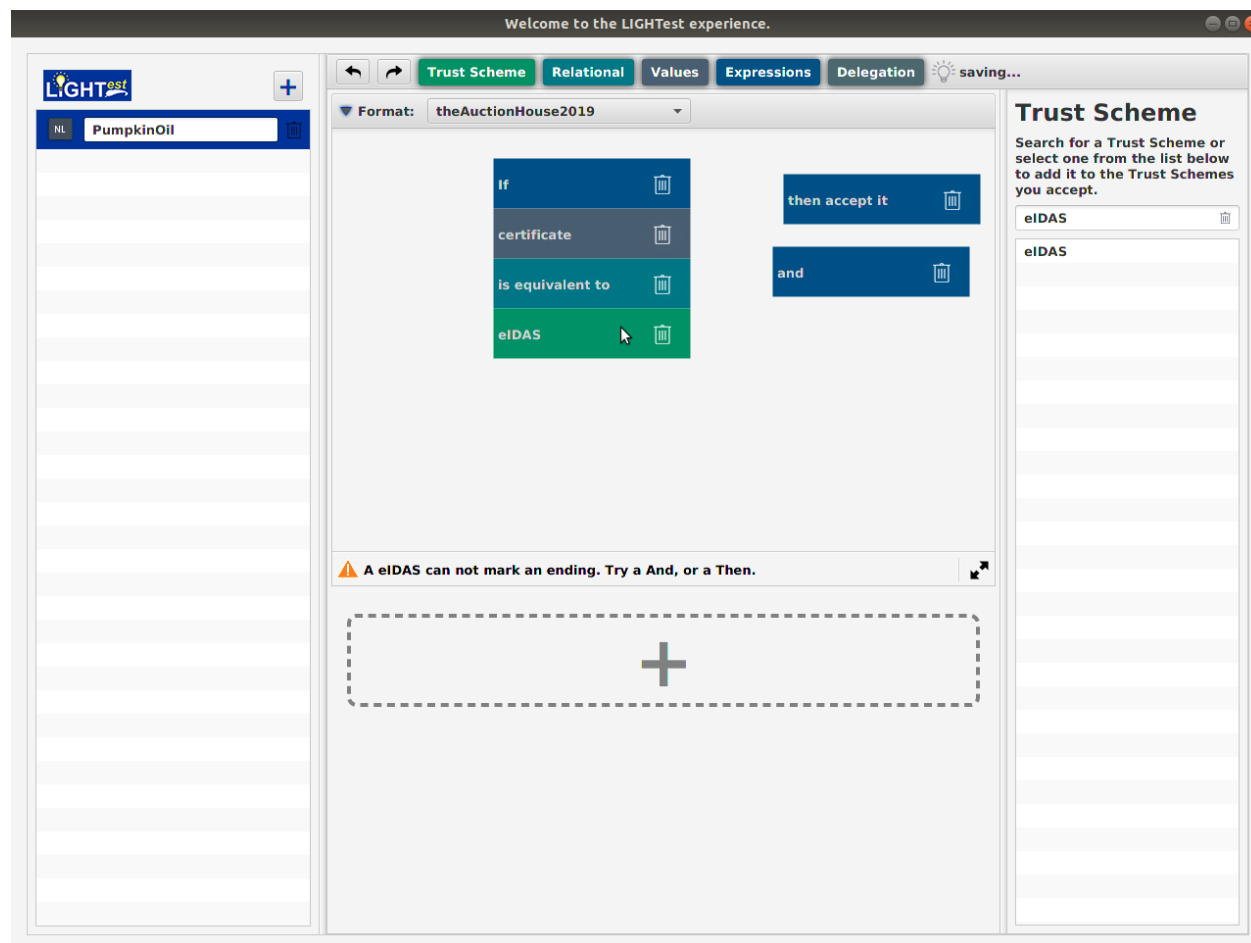


Figure 12 Natural Language Layer with several building blocks (2)

Document name:	Usability and Interaction Design (3)	Page:	20 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



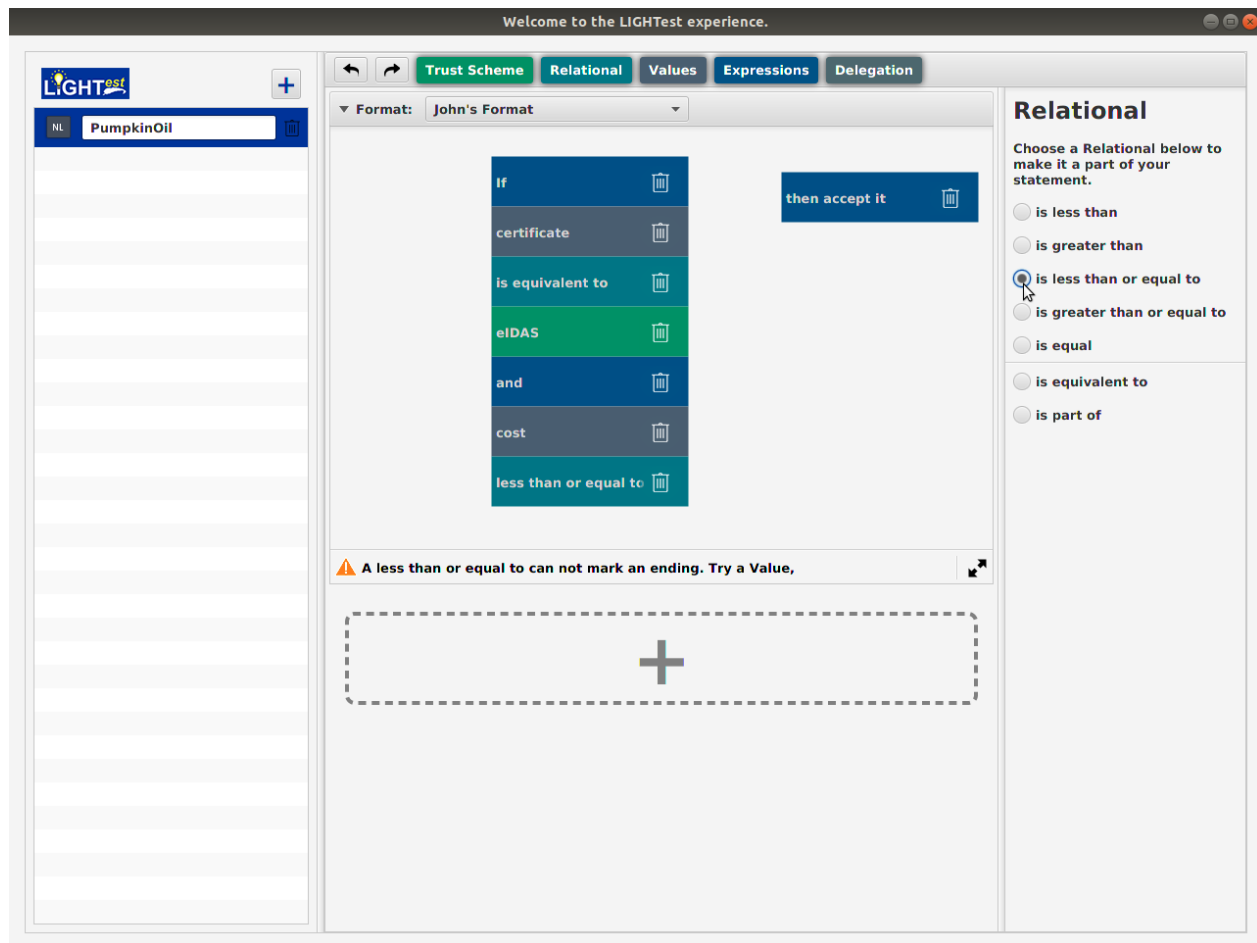


Figure 13 Natural Language with Layer several building blocks (3)

Document name:	Usability and Interaction Design (3)	Page:	21 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



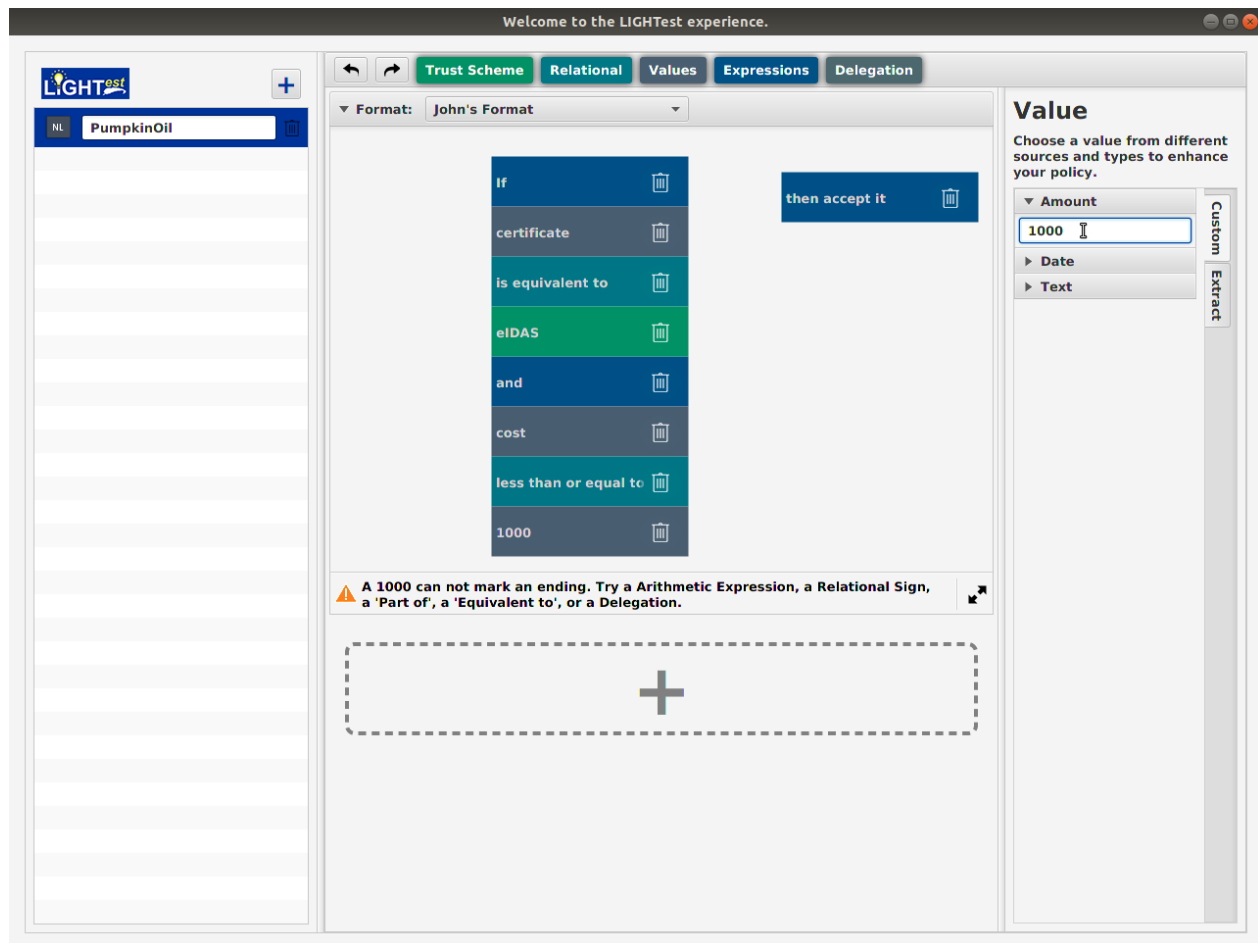


Figure 14 Natural Language Layer with several building blocks (4)

Document name:	Usability and Interaction Design (3)	Page:	22 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



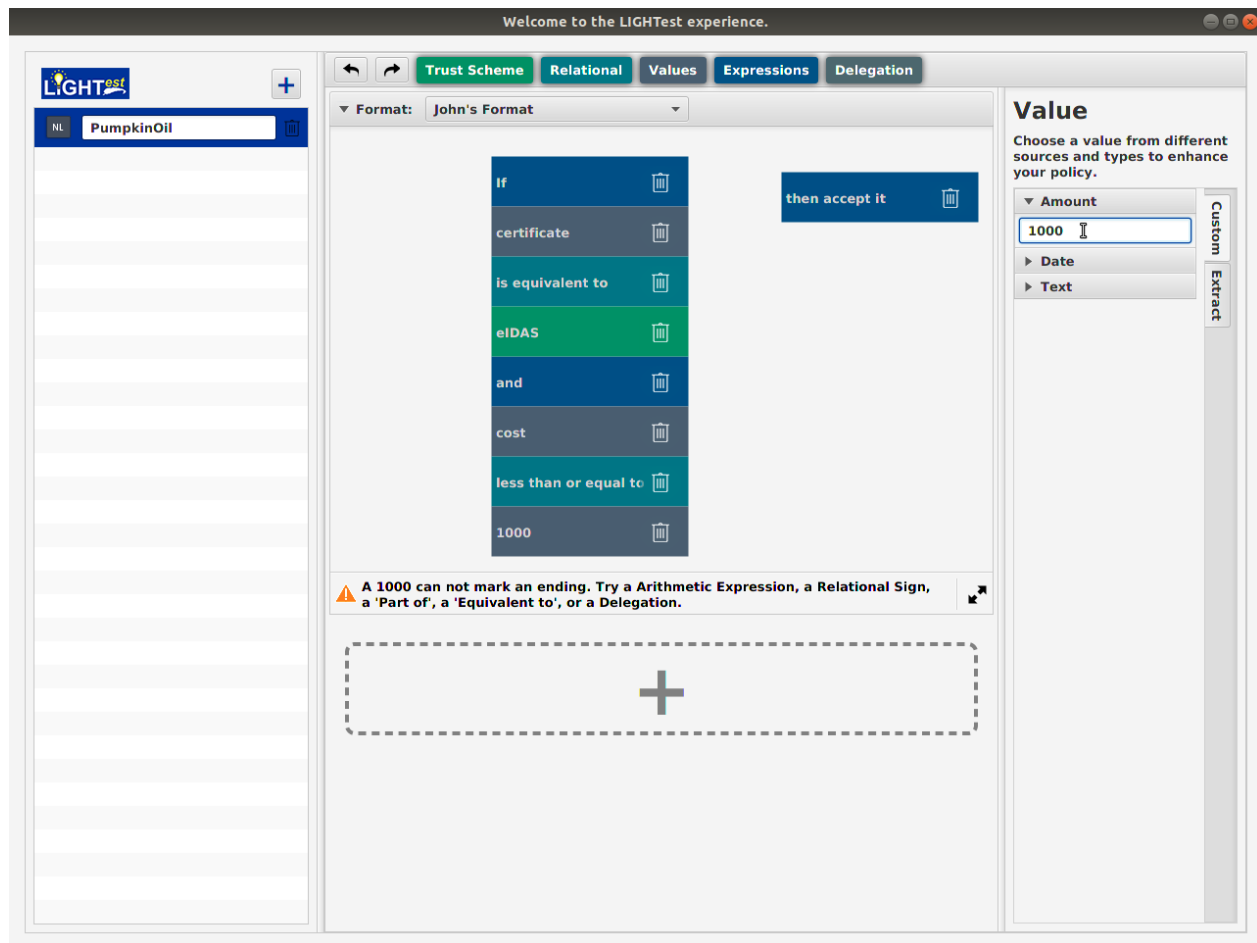


Figure 15 Natural Language Layer with several building blocks (5)

Document name:	Usability and Interaction Design (3)	Page:	23 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



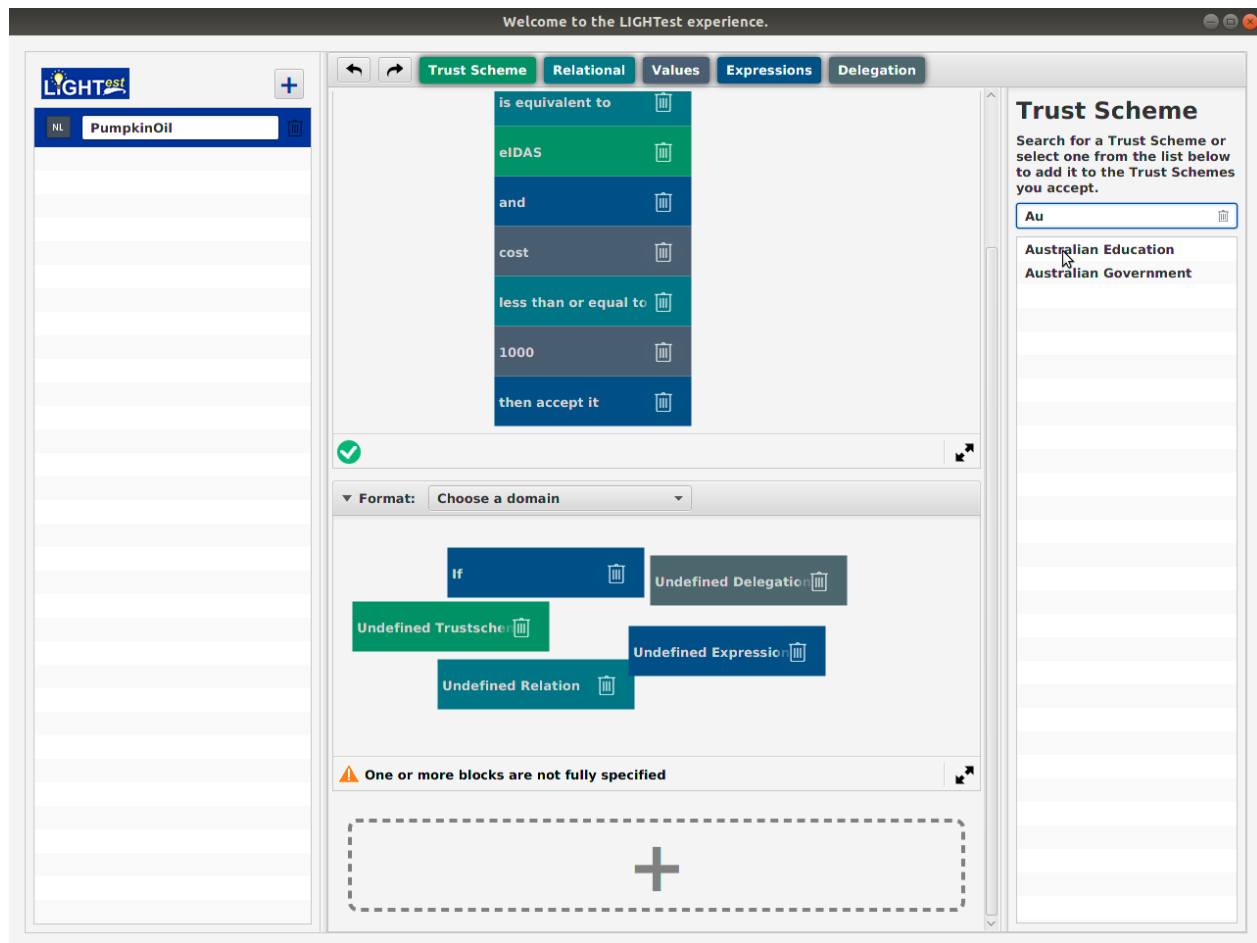


Figure 16 Natural Language Layer: first statement correct, second statement added

Figure 16 shows one finished correct statement. This is indicated with the help of the green control hook.

Moreover, a second statement was added to the Trust Policy.

The following chapter highlights the most important aspects of the usability study. Chapter 7 gives an overview of the results from the third usability evaluation of the implementation of the Trust Policy Authoring Tool.

Document name:	Usability and Interaction Design (3)	Page:	24 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



6. Results from D6.4 User Study

In the iteration between D6.4 and D6.9 we focused on improving the actual implementation of the TPAT rather than having a revised prototype.

As our main focus of the implementation remained to solve the usability issues that were identified in the usability study, the following briefly displays its results. For detailed results please see D6.4.

The evaluation collected quantitative data through questionnaires and qualitative data in the form of observational and verbal data (e.g. comments and arguments by the participants, “think-aloud” data, observed problems and statements).

Both scenarios from D6.4 were split into several subtasks for the evaluation in order to detect the task completion rate.

Scenario 1 – Trust Policy Creation in Graphical Editor – was divided into two subtasks:

1. Create a new Trust Policy using the Graphical Editor
2. Add Trust Schemes to the newly created Trust Policy

With this scenario, users had almost no problems. One minor usability issue was that participants did not know whether they have to save the policy or it is automatically saved. They were also somewhat confused that after clicking the “save” button the trust policy was still selected and displayed in the middle. They were lacking the visual feedback that the trust policy was actually saved.

Scenario 2 – Trust Policy Creation in Natural Language – was divided into three subtasks:

1. Create a new Trust Policy using the Natural Language
2. Formulate a first statement
3. Formulate a second statement

The creation of a first statement in the second scenario seemed to be a bit challenging for some participants. Shortcomings in the prototype were a big factor for that.

Three usability issues were observed:

1. Participants were not sure how to add a building block to the statements. Dragging one from the top and dropping it into the statement did not seem to be the intuitive way. This caused quiet some confusion and took them long to figure out.
2. Participants were unsure in which sequence they had to add the building blocks to the statement. Some participants wanted to put them in a different syntax but the prototype had only one way implemented.
3. Participants managed to change the content of a building block but they had problems with confirming the changes that they made. The confirmation should have been done by

Document name:	Usability and Interaction Design (3)	Page:	25 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



clicking the “Select” button in the bottom right corner of the screen. This button was difficult to detect.

The following table sums up the four major usability issues that were identified in the usability study.

No.	Layer	Usability Issue
1	GE	Saving Trust Policies
2	NL	Adding building blocks
3	NL	Syntax of statement
4	NL	Confirm changes to building blocks

Table 1 Overview of usability issues identified in the usability study

The following addresses explicitly the stated issues and how they were addressed in the implementation of the Trust Policy Authoring Tool.

6.1 Improvements in the implementation

6.1.1 Usability Issue 1: Saving Trust Policies

Although this issues mainly occurred within the Graphical Editor (probably because of the learning effect afterwards), we decided to change this throughout the whole tool to have a seamless interaction.

The original concept was to have a “Save” button in the top right corner so that users save their changes explicitly by clicking on this button. The idea behind was that users could play around with their trust policies and experience their possibilities without having to worry that it will have an immediate effect or “break” the trust policy, as they always had the possibility of canceling them.

We changed this concept to an automatic save function. Meaning that in the implementation there is no “Save” button anymore. Any changes that users make are immediately saved.

Figure 17 Automatic Saving within the TPAT shows the automatic saving function. Every time a change is made to a policy, it is automatically saved. This is visualized with the help of the hourglass animation on top of the middle part.

Document name:	Usability and Interaction Design (3)	Page:	26 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



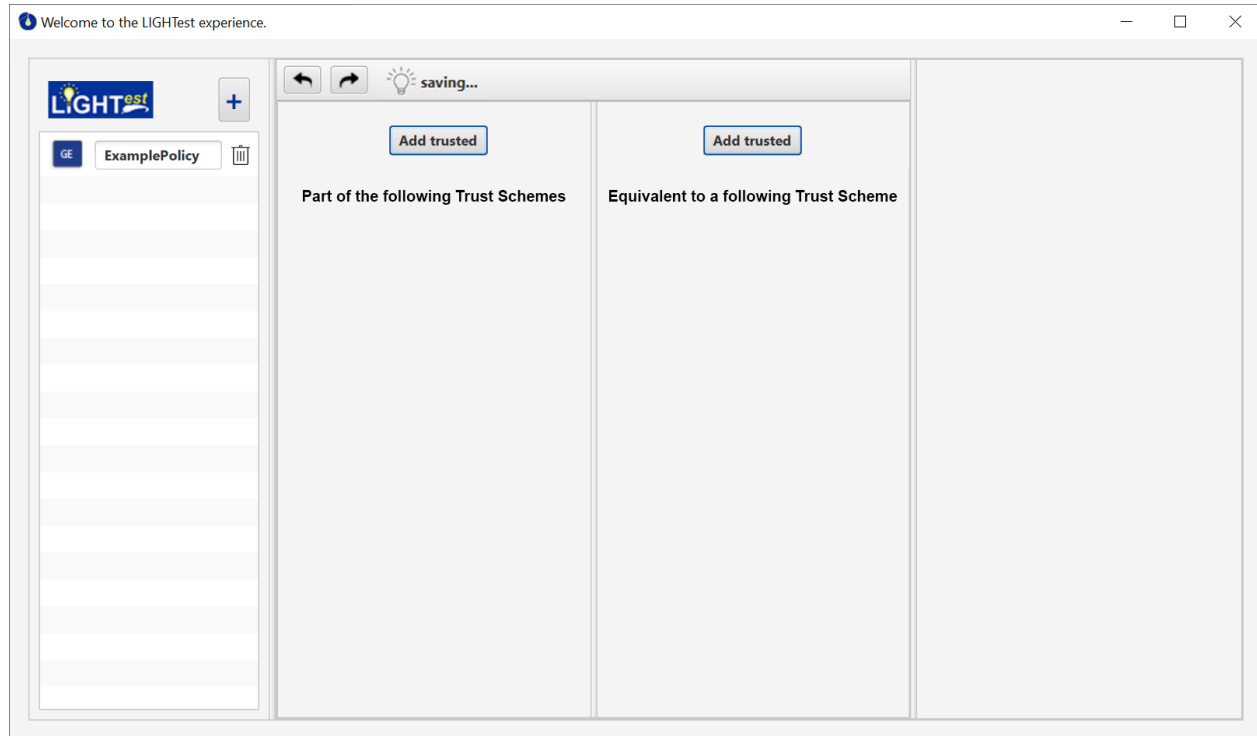


Figure 17 Automatic Saving within the TPAT

The advantage of this is that users do not have to save their changes explicitly anymore, which meets users' expectations as identified in the usability study. To keep the possibilities for users to try out different things without having to worry that they cannot return to the previous version, we implemented an “undo” function.

Figure 18 shows the “undo” function, after deleting a Trust Policy. This concept is well-known within Smartphone Apps and Web Applications.

The same principal applies for any change users make within the TPAT.

Document name:	Usability and Interaction Design (3)	Page:	27 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



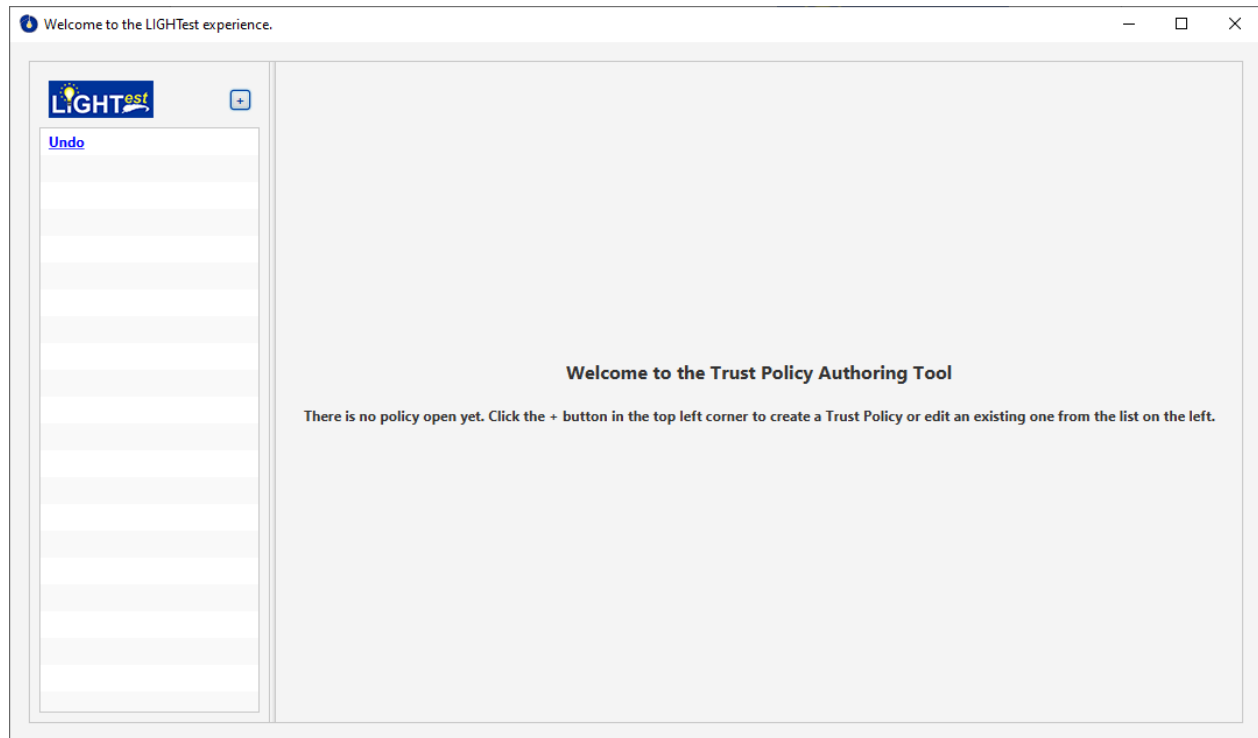


Figure 18 Undo-function after deleting a trust policy

6.1.2 Usability Issue 2: Adding Building Blocks

This problem was solved by adopting the expectations of the participants from the user study:

A building block is added to a statement by clicking on one of the building blocks in top middle part.

The building block immediately appears within the currently selected statement and its parameters show up in the right column. This way users are able to directly change the content of the building block. Every change is immediately saved and the changes are shown within the building block.

Document name:	Usability and Interaction Design (3)	Page:	28 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



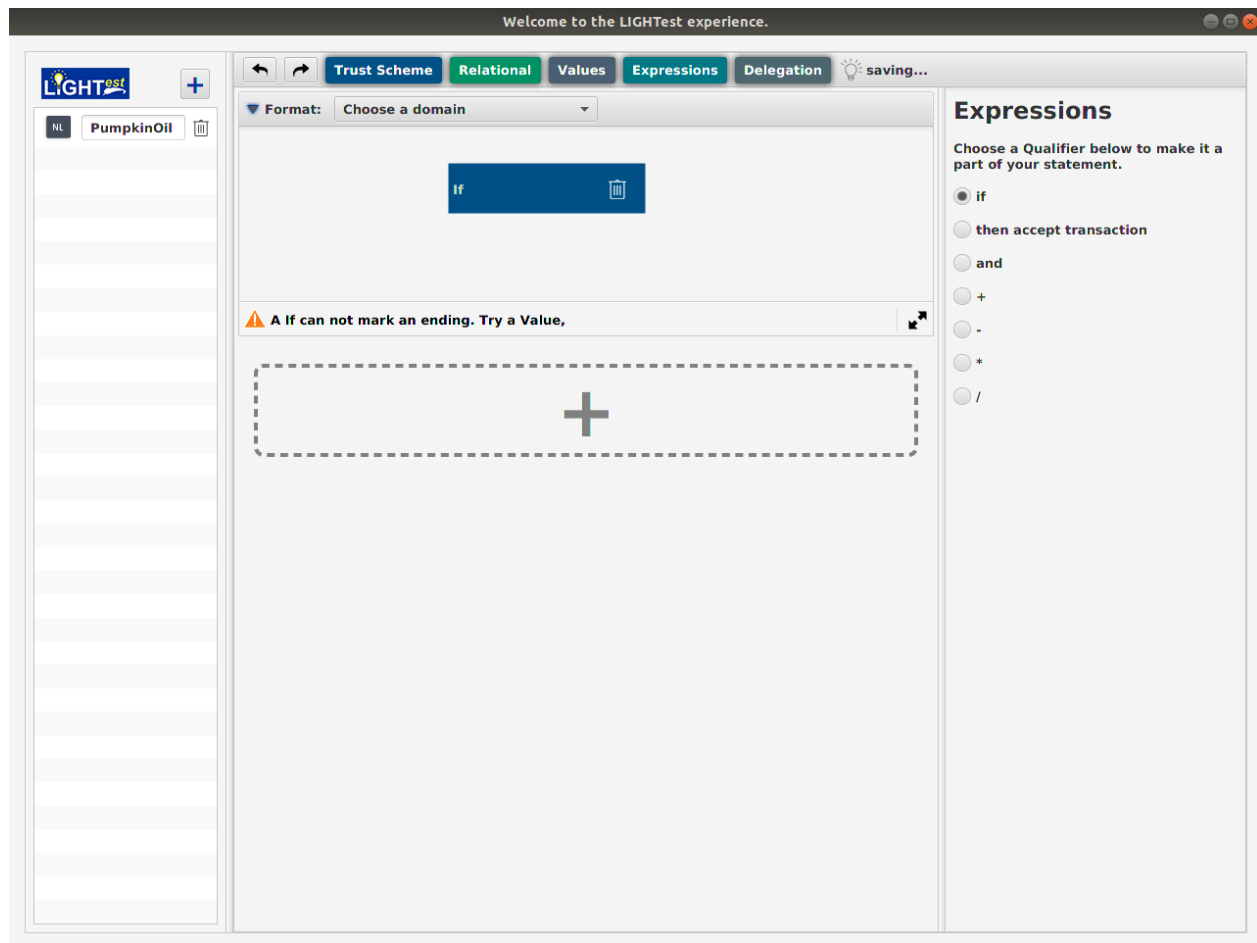


Figure 19 Adding a building block to a statement in the Natural Language Layer

6.1.3 Usability Issue 3: Syntax of statement

This issue was solved by adding syntax correction to the implementation. This is presented to users by messages beneath the currently active statement.

The messages are guiding users through creating a working statement.

Figures Figure 20 to Figure 28 show some examples of such messages.

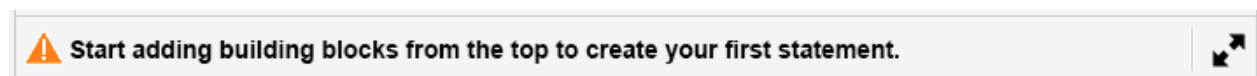


Figure 20 Syntax aid message: Example 1

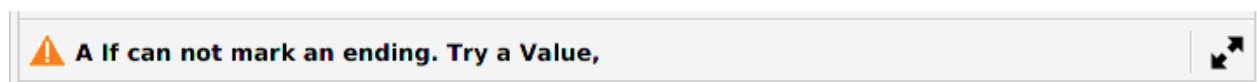


Figure 21 Syntax aid message: Example 2

Document name:	Usability and Interaction Design (3)	Page:	29 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



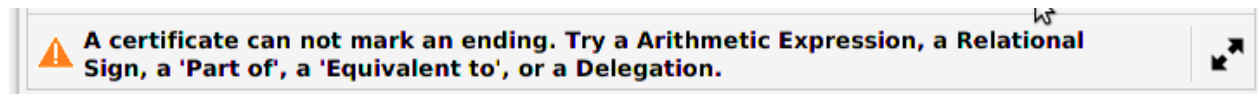


Figure 22 Syntax aid message: Example 3

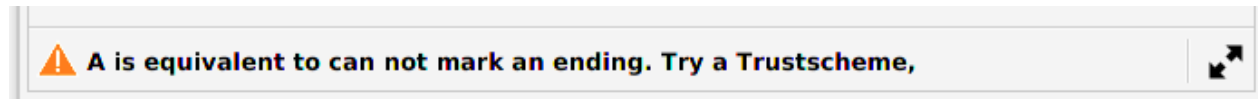


Figure 23 Syntax aid message: Example 4

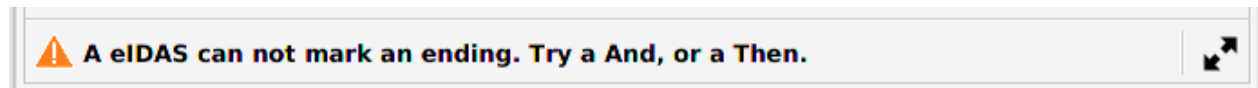


Figure 24 Syntax aid message: Example 5

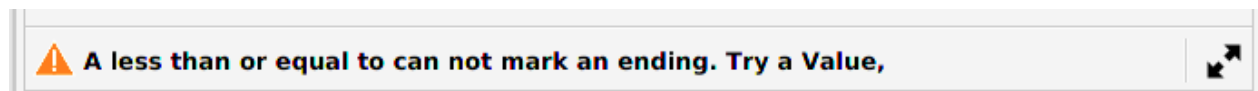


Figure 25 Syntax aid message: Example 6



Figure 26 Syntax aid message: Example 7



Figure 27 Syntax aid message: Example 8

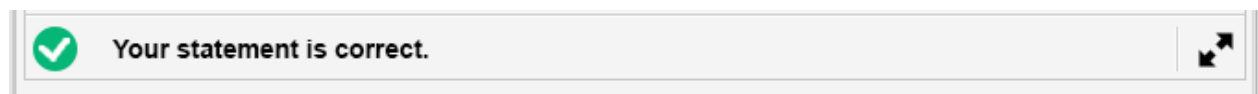


Figure 28 Syntax aid message: Example 9

6.1.4 Usability Issue 4: Confirm changes to building blocks

This issue was solved by automatically saving changes users make to a building block. Users do not have to explicitly save changes anymore.

Document name:	Usability and Interaction Design (3)	Page:	30 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



7. Usability Evaluation

To guarantee an overall good usability, all concept changes after an iteration should be evaluated in regards to their usability.

For this third usability evaluation the method of a heuristic expert evaluation was chosen as it provides consistent outcomes and is sufficient after a bigger and more complex user study like one described in D6.4.

7.1 Methodology

In an expert walkthrough, an expert runs through the system and identifies possible usability problems. The focus is on the processes and procedures involved.

The foundation for that is typical tasks carried out by future users. The expert performs these tasks and places herself in the position of the user.

With the help of this method, processes can be evaluated by usability experts. This allows to identify problems that only become clear in context, such as whether the connection between an interaction and an action of the system is detectable, whether the current status of the system is clear and whether additional operating aids are necessary for certain tasks.

The method is carried out with the help of the following process:

1. Preparation

Typical tasks are prepared in such a way that they are clear and understandable for the usability expert. In addition, the ideal way to work with the system is analyzed and described. If the tasks are similar, it should be considered whether only certain parts of the task are considered for the walkthrough. The system or the prototype is prepared in such a way that the tasks can be carried out exemplarily.

2. Execution

The usability expert works on the defined tasks. She logs the processes in the system, i.e. how she proceeds and all the points that can cause problems.

During the processing she asks herself the following questions again and again:

- Will the user try to achieve the right effect?
- Will the user recognize that the correct action is available?
- Will the user make a connection between the correct action and the desired effect?
- Will the user see the progress when the correct action has been performed?

During the execution step, also well-known usability design guidelines or heuristics can be used.

Document name:	Usability and Interaction Design (3)	Page:	31 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



3. Evaluation

All potential problems are collected and assigned a severity level to. In addition, the processes for the individual tasks are questioned in relation to the problems found.

7.1.1 Collection of Usability Heuristics

The collections of usability heuristics presented below are well-known and well-established guidelines that serve as general statements for improving usability in systems or products. All these guidelines were taken under consideration within the expert evaluation.

Nielsen's Ten Usability Heuristics (Nielsen, 1995)

1. Visibility of system status
2. Match between system and real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

Shneiderman's Eight Golden Rules of Interface Design (Shneiderman & Plaisant, 2009)

1. Strive for consistency: Consistent sequences of actions in similar situations; identical terminology in prompts, menus, and help screens; and consistent commands.
2. Enable frequent users to use shortcuts: Abbreviations, function keys, hidden commands, and macro facilities.
3. Offer informative feedback: For every operator action, there should be system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.
4. Design dialog to yield closure: Sequences of actions should be organized into groups with a beginning, middle, and end. Feedback at the completion of a group of actions.
5. Offer simple error handling: As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple mechanisms for handling it.
6. Permit easy reversal of actions: Errors can be undone. The units of reversibility may be a single action, a data entry, or a complete group of actions.
7. Support internal focus of control: Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions.

Document name:	Usability and Interaction Design (3)	Page:	32 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



8. Reduce short-term memory load: Displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

Norman's 7 Design Principles (Norman, 2002)

1. Use both knowledge in the world and knowledge in the head.
2. Simplify the structure of tasks.
3. Make things visible: Bridge the gulfs of Execution and Evaluation.
4. Get the mappings right.
5. Exploit the power of constraints, both natural and artificial.
6. Design for error.
7. When all else fails, standardize.

7.2 Results

For the evaluation the expert review was conducted by a Usability and User Experience expert from the University of Stuttgart.

The expert followed the methodology described in chapter 7.1 going through each Usability Heuristic in chapter 7.1.1 and each has a results:

- **Passed**
 - o A heuristic is "Passed" if it meets the implementation.
- **Neutral**
 - o A heuristic is "Neutral" in case we cannot evaluate it in the current implementation or if it is possible to be applied in the future
- **Not Passed**
 - o A heuristic is "Not Passed" if it does not meet the heuristics requirements and cannot be applied in the future
- **Not Applicable**
 - o A heuristic is "Not Applicable" if the heuristic cannot be applied to the implementation

For the full list of evaluation results see Chapter 10 Appendix.

In this chapter we will only address the parts as identified "neutral" and "not passed" and give reasoning for this decision as well as an outlook what should be improved/done.

7.2.1 Not Applicable

All heuristics marked further in the document as "Not Applicable" are listed and more details are provided on each of them.

Document name:	Usability and Interaction Design (3)	Page:	33 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



7.2.1.1 General

There are no heuristics marked as “Not Applicable”.

7.2.1.2 Graphical Editor

There are no heuristics marked as “Not Applicable” regarding the Graphical Editor.

7.2.1.3 Natural Language Layer

There are no heuristics marked as “Not Applicable” regarding the Natural Language Layer.

7.2.2 Neutral

All heuristics marked further in the document as “Neutral” are listed and more details are provided on each of them.

7.2.2.1 General

There are three heuristics marked as “neutral” regarding the general interaction of the TPAT.

The following table gives an overview of them, as well as reasons why they received such an evaluation.

Table 2 Evaluation results - neutral, overall TPAT

No	Heuristic	Creating a new policy	Selecting a layer	Giving a name	Renaming a policy	Deleting a policy	Selecting a policy	Undo Changes
8	Aesthetic and minimalist design	Neutral The current implementation focuses on functionality rather than design. Due to quiet some technical limitations the design could not be applied as stated in D6.4. This does not harm the functionality and can be improved in future releases.						
10	Help and documentation	Neutral Although this is a part of the concept, it was not the main focus of it. To provide help and documentation to users takes a whole team to build. As the project has limited time and resources it was decided to not focus on this aspect. Future work should consider it though.						
12	Enable frequent users to use shortcuts	Neutral Also, the topic of shortcuts was not addressed explicitly within the concept. The Tool can be perfectly used without shortcuts. Nevertheless, they do add a certain benefit and should be considered in future works.						

7.2.2.2 Graphical Editor

There are three heuristics marked as “neutral” regarding the Graphical Editor.

The following table gives an overview of them, as well as reasons why they received such an evaluation.

Document name:	Usability and Interaction Design (3)	Page:	34 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



Table 3 Evaluation results - neutral, Graphical Editor

No	Heuristic	Adding Trust Schemes	Change Trust Scheme	Delete Trust Scheme	Undo Changes
8	Aesthetic and minimalist design	Neutral The current implementation focuses on functionality rather than design. Due to quiet some technical limitations the design could not be applied as stated in D6.4. This does not harm the functionality and can be improved in future releases.			
9	Help users recognize, diagnose, and recover from errors	Neutral The interaction concept assumes that nothing can go wrong with the creation of a trust policy in the graphical editor. The possibility of a not correct policy created in the graphical editor should be addressed in the future.			
10	Help and documentation	Neutral Also the topic of shortcuts was not addressed explicitly within the concept. The Tool can be perfectly used without shortcuts. Nevertheless they do add a certain benefit and should be considered in future works.			

7.2.2.3 Natural Language Layer

There are three heuristics marked as “neutral” regarding the Natural Language Layer.

The following table gives an overview of them, as well as reasons why they received such an evaluation.

Table 4 Evaluation Results - neutral, Natural Language Layer

No	Heuristic	Adding Building Block	Change content	Delete Building Block	Create correct Statement	Add statement
8	Aesthetic and minimalist design	Neutral The current implementation focuses on functionality rather than design. Due to quiet some technical limitations the design could not be applied as stated in D6.4. This does not harm the functionality and can be improved in future releases.				
10	Help and documentation	Neutral Also the topic of shortcuts was not addressed explicitly within the concept. The Tool can be perfectly used without shortcuts. Nevertheless they do add a certain benefit and should be considered in future works.				

Overall, the evaluation results are positive.

All the heuristics that were marked as “neutral” are not crucial and can be addressed in future work. Therefore, the Trust Policy Authoring Tool has an overall good usability.



8. References

Ackerman, M. S. & Mainwarin, S. D., 2005. Privacy Issues and Human-Computer Interaction. In: L. & G. S. Cranor, Hrsg. *Security and Usability - Designing Secure Systems That People Can Use*. Sebastopol, CA: O'Reilly Media, pp. 381-400.

D2.3, 2017. *Requirements and Use Cases*, s.l.: s.n.

D2.4, 2017. *Format Description and Analysis of Concept*, s.l.: LIGHTest Project Deliverable.

DIN ISO 9241-11, 2017. *Ergonomics of human-system interaction*. Berlin: Beuth Verlag GmbH.

DIN ISO 9241-210, 2011. *Ergonomics of human-system interaction - Part210: Human-centred design for interactive systems*. Berlin: Beuth Verlag GmbH.

Fischer-Hübner, S., 2011. Transparency Enhancing Technologies & HCI for Policy Display and Informed Consent. *Privacy, Accountability, Trust - Challenges and Opportunities, ENISA Report*, pp. 30-37.

Fischer-Hübner, S., Lo Iacono, L. & Möller, S., 2010. Usable Security und Privacy. *Datenschutz und Datensicherheit - DuD*, Volume 34, Issue 11 November, pp. 773-782.

Fronemann, N. & Peissner, M., 2014. *User Experience Concept Exploration - User Needs as a Source for Innovation*. Helsinki: ACM.

Hansen, M., 2013. Data Protection by Default in Identity-Related Applications. In: I. I. F. f. I. P. IFIP AICT 396, Hrsg. *Fischer-Hübner, S. et al. (Eds.), IDMAN*. s.l.:s.n., pp. 4-17.

Hassenzahl, M., 2008. User Experience (UX): Towards an Experiential Persepctive on Product Quality.. *Proceedings of the 20th French-speaking conference on Human-computer interaction*, pp. 11-15.

Lederer, S., Hong, J. I., Dey, A. & Landay, J., 2004. *Personal Privacy through Understanding and Action: Five Pitfalls for Designers, Personal and Ubiquitous Computing*. London, UK: s.n.

LIGHTest, 2017. *Project Deliverable - D6.1 Requirements and Design of a Conceptual Framework for Trust Policies*, s.l.: s.n.

Markotten, D. G. T., 2004. *Benutzbare Sicherheit in informationstechnischen Systemen, ALbert-Ludwigs-Universität Freiburg, Dissertation. 2004..* s.l.:Summarized in: Petterson, J.S. & Meints, M. (Eds.) (2009) Freiburg Guidelines for Security Tools. Chapter 4.1.3 in: Study on Usability of IMS. FIDIS - Future of Identity in the Information Society, Deliverable 3.13, FIDIS, pp. 26-27.

Nielsen, J., 1995. *10 Usability Heuristics for User Interface Design*, <https://www.nngroup.com/articles/ten-usability-heuristics/> (Accessed on 07-25-2017): Nielsen Norman Group.

Document name:	Usability and Interaction Design (3)	Page:	36 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



Usability and Interaction Design (3)



Norman, D., 2002. *The Design of Everyday Things*. 1 Hrsg. New York: Basic Book.

Patrick, A. S., Briggs, P. & Marsh, S., 2005. Designing Systems That People Will Trust. In: L. Faith Cranor & S. Garfinkel, Hrsg. *Security and Usability - Designing Secure Systems That People Can Use*. USA: O'Reilly Media Inc., pp. 75-99.

Shneiderman, B. & Plaisant, C., 2009. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 5 Hrsg. s.l.:Addison-Wesles Educational Publishers Inc..

WCAG2.0, 2008. <https://www.w3.org/TR/2008/REC-WCAG20-20081211/> (Accessed on 07-25-2017): W3C.

Whitten, A. & Tygar, J., 1999. Why Jhonny Can't Encrypt - A Usability Evaluation of PGP 5.0. *Proceedings of the 8th USENIX Security Symposium*, August, pp. 169-184.

Yee, K.-P., 2004. Aligning Security and Usability. *Security & Privacy Volume 2, Issue 5, IEEE*, pp. 48-55.

Document name:	Usability and Interaction Design (3)	Page:	37 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



9. Project Description

LIGHTest project to build a global trust infrastructure that enables electronic transactions in a wide variety of applications

An ever increasing number of transactions are conducted virtually over the Internet. How can you be sure that the person making the transaction is who they say they are? The EU-funded project LIGHTest addresses this issue by creating a global trust infrastructure. It will provide a solution that allows one to distinguish legitimate identities from frauds. This is key in being able to bring an efficiency of electronic transactions to a wide application field ranging from simple verification of electronic signatures, over eProcurement, eJustice, eHealth, and law enforcement, up to the verification of trust in sensors and devices in the Internet of Things.

Traditionally, we often knew our business partners personally, which meant that impersonation and fraud were uncommon. Whether regarding the single European market place or on a Global scale, there is an increasing amount of electronic transactions that are becoming a part of peoples everyday lives, where decisions on establishing who is on the other end of the transaction is important. Clearly, it is necessary to have assistance from authorities to certify trustworthy electronic identities. This has already been done. For example, the EC and Member States have legally binding electronic signatures. But how can we query such authorities in a secure manner? With the current lack of a worldwide standard for publishing and querying trust information, this would be a prohibitively complex leading to verifiers having to deal with a high number of formats and protocols.

The EU-funded LIGHTest project attempts to solve this problem by building a global trust infrastructure where arbitrary authorities can publish their trust information. Setting up a global infrastructure is an ambitious objective; however, given the already existing infrastructure, organization, governance and security standards of the Internet Domain Name System, it is with confidence that this is possible. The EC and Member States can use this to publish lists of qualified trust services, as business registrars and authorities can in health, law enforcement and justice. In the private sector, this can be used to establish trust in inter-banking, international trade, shipping, business reputation and credit rating. Companies, administrations, and citizens can then use LIGHTest open source software to easily query this trust information to verify trust in simple signed documents or multi-faceted complex transactions.

The three-year LIGHTest project starts on September 1st and has an estimated cost of almost 9 Million Euros. It is partially funded by the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321. The LIGHTest consortium consists of 14 partners from 9 European countries and is coordinated by Fraunhofer-Gesellschaft. To reach out beyond Europe, LIGHTest attempts to build up a global community based on international standards and open source software.

Document name:	Usability and Interaction Design (3)	Page:	38 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



Usability and Interaction Design (3)



The partners are ATOS (ES), Time Lex (BE), Technische Universität Graz (AT), EEMA (BE), G+D (DE), Danmarks tekniske Universitet (DK), TUBITAK (TR), Universität Stuttgart (DE), Open Identity Exchange (GB), NLNet Labs (NL), CORREOS (ES), Ubisecure (FI), and University of Piraeus Research Center – UPRC (GR). The Fraunhofer IAO provides the vision and architecture for the project and is responsible for both, its management and the technical coordination.

Document name:	Usability and Interaction Design (3)	Page:	39 of 46		
Dissemination:	PU	Version:	Version 1.0	Status:	Final



10. Appendix

10.1 Results from expert walkthrough of general aspects of the TPAT

Table 5 All Evaluation results on TPAT considering the general interaction

No	Heuristic	Creating a new policy	Selecting a layer	Giving a name	Renaming a policy	Deleting a policy	Selecting a policy	Undo Changes
1	Visibility of system status	Passed	Passed	Passed	Passed	Passed	Passed	Passed
2	Match between system and real world	Passed	Passed	Passed	Passed	Passed	Passed	Passed
3	User control and freedom	Passed	Passed	Passed	Passed	Passed	Passed	Passed
4	Consistency and standards	Passed	Passed	Passed	Passed	Passed	Passed	Passed
5	Error prevention	Passed	Passed	Passed	Passed	Passed	Passed	Passed
6	Recognition rather than recall	Passed	Passed	Passed	Passed	Passed	Passed	Passed
7	Flexibility and efficiency of use	Passed	Passed	Passed	Passed	Passed	Passed	Passed
8	Aesthetic and minimalist design	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
9	Help users recognize, diagnose, and recover from errors	Passed	Passed	Passed	Passed	Passed	Passed	Passed
10	Help and documentation	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
11	Strive for consistency	Passed	Passed	Passed	Passed	Passed	Passed	Passed
12	Enable frequent users to use shortcuts	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
13	Offer informative feedback	Passed	Passed	Passed	Passed	Passed	Passed	Passed

Document name:	Usability and Interaction Design (3)	Page:	40 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



Usability and Interaction Design (3)



14		Design dialog to yield closure	Passed	Passed	Passed	Passed	Passed	Passed	Passed
15		Offer simple error handling	Passed	Passed	Passed	Passed	Passed	Passed	Passed
16		Permit easy reversal of actions	Passed	Passed	Passed	Passed	Passed	Passed	Passed
17		Support internal focus of control	Passed	Passed	Passed	Passed	Passed	Passed	Passed
18		Reduce short-term memory load	Passed	Passed	Passed	Passed	Passed	Passed	Passed
19	Norman	Use both knowledge in the world and knowledge in the head.	Passed	Passed	Passed	Passed	Passed	Passed	Passed
20		Simplify the structure of tasks.	Passed	Passed	Passed	Passed	Passed	Passed	Passed
21		Make things visible: Bridge the gulfs of Execution and Evaluation.	Passed	Passed	Passed	Passed	Passed	Passed	Passed
22		Get the mappings right.	Passed	Passed	Passed	Passed	Passed	Passed	Passed
23		Exploit the power of constraints, both natural and artificial.	Passed	Passed	Passed	Passed	Passed	Passed	Passed
24		Design for error	Passed	Passed	Passed	Passed	Passed	Passed	Passed

Document name:	Usability and Interaction Design (3)	Page:	41 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



10.2 Results from expert walkthrough of the Graphical Editor

Table 6 All the evaluation results on the Graphical Editor

No	Heuristic	Adding Trust Schemes	Change Trust Scheme	Delete Trust Scheme	Undo Changes	
1	Nielsen	Visibility of system status	Passed	Passed	Passed	Passed
2		Match between system and real world	Passed	Passed	Passed	Passed
3		User control and freedom	Passed	Passed	Passed	Passed
4		Consistency and standards	Passed	Passed	Passed	Passed
5		Error prevention	Passed	Passed	Passed	Passed
6		Recognition rather than recall	Passed	Passed	Passed	Passed
7		Flexibility and efficiency of use	Passed	Passed	Passed	Passed
8	Shneidermann	Aesthetic and minimalist design	Neutral	Neutral	Neutral	Neutral
9		Help users recognize, diagnose, and recover from errors	Neutral	Neutral	Neutral	Neutral
10		Help and documentation	Neutral	Neutral	Neutral	Neutral
11		Strive for consistency	Passed	Passed	Passed	Passed
12	Shneidermann	Enable frequent users to use shortcuts	Passed	Passed	Passed	Passed
13		Offer informative feedback	Passed	Passed	Passed	Passed
14		Design dialog to yield closure	Passed	Passed	Passed	Passed



Usability and Interaction Design (3)



15		Offer simple error handling	Passed	Passed	Passed	Passed
16		Permit easy reversal of actions	Passed	Passed	Passed	Passed
17		Support internal focus of control	Passed	Passed	Passed	Passed
18		Reduce short-term memory load	Passed	Passed	Passed	Passed
19	Norman	Use both knowledge in the world and knowledge in the head.	Passed	Passed	Passed	Passed
20		Simplify the structure of tasks.	Passed	Passed	Passed	Passed
21		Make things visible: Bridge the gulfs of Execution and Evaluation.	Passed	Passed	Passed	Passed
22		Get the mappings right.	Passed	Passed	Passed	Passed
23		Exploit the power of constraints, both natural and artificial.	Passed	Passed	Passed	Passed
24		Design for error	Passed	Passed	Passed	Passed

Document name:	Usability and Interaction Design (3)	Page:	43 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



10.3 Results from expert walkthrough of the Natural Language Layer

Table 7 All evaluation results of the Natural Language Layer

No	Heuristic	Adding Building Block	Change content	Delete Building Block	Create correct Statement	Add statement
1	Visibility of system status	Passed	Passed	Passed	Passed	Passed
2	Match between system and real world	Passed	Passed	Passed	Passed	Passed
3	User control and freedom	Passed	Passed	Passed	Passed	Passed
4	Consistency and standards	Passed	Passed	Passed	Passed	Passed
5	Error prevention	Passed	Passed	Passed	Passed	Passed
6	Recognition rather than recall	Passed	Passed	Passed	Passed	Passed
7	Flexibility and efficiency of use	Passed	Passed	Passed	Passed	Passed
8	Aesthetic and minimalist design	Neutral	Neutral	Neutral	Neutral	Neutral
9	Help users recognize, diagnose, and recover from errors	Passed	Passed	Passed	Passed	Passed
10	Nielsen Help and documentation	Neutral	Neutral	Neutral	Neutral	Neutral
11	Shneiderman Strive for consistency	Passed	Passed	Passed	Passed	Passed
12	Shneiderman Enable frequent	Passed	Passed	Passed	Passed	Passed



Usability and Interaction Design (3)



		users to use shortcuts					
13		Offer informative feedback	Passed	Passed	Passed	Passed	Passed
14		Design dialog to yield closure	Passed	Passed	Passed	Passed	Passed
15		Offer simple error handling	Passed	Passed	Passed	Passed	Passed
16		Permit easy reversal of actions	Passed	Passed	Passed	Passed	Passed
17		Support internal focus of control	Passed	Passed	Passed	Passed	Passed
18		Reduce short-term memory load	Passed	Passed	Passed	Passed	Passed
19		Use both knowledge in the world and knowledge in the head.	Passed	Passed	Passed	Passed	Passed
20		Simplify the structure of tasks	Passed	Passed	Passed	Passed	Passed
21		Make things visible: Bridge the gulfs of Execution and Evaluation	Passed	Passed	Passed	Passed	Passed
22		Get the mappings right	Passed	Passed	Passed	Passed	Passed
23	Norman	Exploit the power of constraints , both	Passed	Passed	Passed	Passed	Passed

Document name:	Usability and Interaction Design (3)	Page:	45 of 46
Dissemination:	PU	Version:	Version 1.0
		Status:	Final



Usability and Interaction Design (3)



	natural and artificial					
24	Design for error	Passed	Passed	Passed	Passed	Passed

Document name:	Usability and Interaction Design (3)	Page:	46 of 46		
Dissemination:	PU	Version:	Version 1.0	Status:	Final

