# Process, domain, and simplification modelling mythology

This document describes the methodology introduced in the Small Claim Form modelling.

The Small Claim Form is the digital transposition of a process which gives the possibly to a claimant to send a claim to the European court. It is a simple and a strait forward process consisting in different steps the claimant must follow to complete the required information. At the end of the procedure a PDF file is generated.

The following picture shows the Small Claim Form process from the start to the generation of the PDF file that must be sent to the Court.

|  |  |  |
| --- | --- | --- |
| Immagine che contiene testo  Descrizione generata automaticamente | Immagine che contiene testo  Descrizione generata automaticamente | Immagine che contiene testo  Descrizione generata automaticamente |
| Immagine che contiene testo  Descrizione generata automaticamente | Immagine che contiene testo  Descrizione generata automaticamente |  |

The idea is to model different aspects of the process separately. Then, starting from these aspects, machinery will generate the final form. The aspects that must be modelled are the process, the domain, and the information. For each of these aspects, a model had been defined. In the following, the different models are described.

## Process model

The *process model* describes the different steps the user has to complete to fulfil all required *blocks* of the form. In the process model, a block consists of a set of information and data referring to a single concept (claimant data, defendant data, claim data, eg.).

The *process model* refers to the specific *domain and simplification* model used in the form completion. The blocks element defines the steps of the process. If required, the process model will manage also the connection between the model element and the form page. In particulare, it uses XPATH to link a specific model element (eg. a block or a field) to the specific HTML element (eg. a <div …> o <input …> element). This will be done using the attribute elements (this will be implemented if required).

{

"model": "2.0",

"concepts": [

"http://scanii.org/domain"

],

"simplification": "http://scanii.org/simplification/smallclaimform",

"blocks": [],

"elements": []

}

Each block has an id, and a type, and refers to a specific HTML through the XPath element. As defined above a block refers to a specific domain concept. In the example below, the block refers to the claimant concept. The ONE\_OR\_MANY is a keyword which specifies that the user can add one or many claimants.

{

"id": "step1",

"type": "BLOCK",

"xpath": "//\*[@id='step1']",

"concept": [

"ONE\_OR\_MANY(http://scanii.org/domain/claimant)"

],

"blocks": [],

"dependencies": [],

"inCondition": [],

"outCondition" : []

}

To generalize the use of the process model a *condition specification language* has been introduced. The *condition specification language* can manage *priority conditions* (eg. step 2 must be executed only the step 1 is completed) and *value base conditions* (eg. if a specific field in concept x is true then execute step 3 otherwise execute step 4). In the model, the *priority conditions* are defined in the dependencies element. Meanwhile, the *value base condition*s are defined in the inCondition and outCondition elements. The conditions are specified based on the domain concepts.

{

"id": "step2",

"type": "BLOCK",

"xpath": "//\*[@id='step2']",

"concepts": [

"ONE\_OR\_MANY(http://scanii.org/domain/defendant)"

],

"dependencies": [

"step1"

],

"inCondition": [],

"outCondition" : ["http://scanii.org/domain/defendant.name != ''''", "http://scanii.org/domain/defendant.surname != ''''"]

}

When a specific step of the process refers to different *domain concepts* a set of sub-blocks is introduced. This can preserve the unique link between *the process block* and *domain concept*.

{

"id": "step4",

"type": "BLOCK",

"xpath": "//\*[@id='step4']",

"blocks": [

{

"id": "step4\_1",

"type": "BLOCK",

"xpath": "//\*[@id='step4\_1']",

"concepts": [

"http://scanii.org/domain/crossborderNature"

],

"dependencies": []

},

{

"id": "step4\_2",

"type": "BLOCK",

"xpath": "//\*[@id='step4\_2']",

"concepts": [

"http://scanii.org/domain/bankDetails"

],

"dependencies": [

"step4\_1"

]

}

],

"dependencies": [

"step3"

]

}

## Domain model

The domain model specifies all the entities that must be completed by the final user.

{

"domain": "http://scanii.org/domain",

"concepts": []

}

For each entity, both the metainformation and the properties must be defined. The meta element contains the concept uri and other required information. Multiple hierarchies have been implemented through the elements extends, extendsOr and extendsAnd. This is required because in many cases the concept derives from different concepts. As an example, a *claimant* could be a *citizen* or an *organization*. The props element gives the possibility to define the basic property of the concept.

{

"meta": {

"uri": "http://scanii.org/domain/claimant",

"extendsOr": [

"http://scanii.org/datastructure/citizen",

"http://scanii.org/datastructure/organization"

]

},

"props": {

"representative": "ZERO\_OR\_MANY(http://scanii.org/domain/representative)",

"otherDetails": "string"

}

}

## Information model

As presented in the following picture a form can contain different types of information to help the final user to complete the task. The *information model* describes this data. The idea then is that this kind of information (description text, Pop-up information and external resources) can be adapted, through some simplification techniques, to better fit the final user's capabilities and needs. This is done by applying to the original form a set of simplification techniques which automatically generate an annotated version of the original information. The annotations are that presented to the final user and this helps him/her in processing and understanding the original information.

| Immagine che contiene testo  Descrizione generata automaticamente |  |  |
| --- | --- | --- |

The model identifies all the information associated with a set of concepts in a specific domain.

{

"domain": "http://scanii.org/smallclaimform",

"concepts": []

}

For each concept it is possible to specify a different type of pieces of information:

* DESCRIPTION: it is a text that (may contain links to external resources) that aims at giving the final user the context and the information required to complete the concept data.
* LEGAL\_REFERENCE: it is a text that (may contain links to external resources) that aim at giving the final user the legal context and the legal information required to complete the concept data.
* INFORMATION: it is a text that (may contain links to external resources) that aim at giving the final user-specific information required to complete the concept data. It can be viewed as the information which appears when the mouse goes in over on a specific form element.

{

"id": "http://scanii.org/domain/claimant",

"infoEements": [

{

"type": "DESCRITPION|LEGAL\_REFERENCE|INFORMATION",

"text": "Original text",

"simplifiedText": "Simplified text",

"link": [

{

"url": "resource url",

"name": "resource name",

"description": "resource description"

}

],

"words": [

{

"position": "word position in the original text",

"definition": "word definition",

"synonyms": [

"synonym 1”, "synonym 2”

],

"links": [

"relevant link 1", "relevant link 2"

]

}

]

}

]

}

The idea is that the information model is created in two moments. The first version of the model is implemented by the modeller. Then the model is passed through a machine which applies some simplification techniques. This machinery compiles part of the information of the model. In particular, it completes the simplifiedText element and the words element.