## Alignment of key concepts of greenfox and SHACL

This appendix summarizes the conceptual alignment between greenfox and SHACL. The striking correspondence is a consequence of our decision to use SHACL as a blueprint for the conceptual framework underlying the greenfox language. Greenfox can be thought of as a combination of SHACL’s abstract validation model with a view of the file system through the prism of a unified value and expression model (XDM, XPath/XQuery + foxpath).

The alignment is described in two tables. The first table provides an aligned definition of the validation process as a decomposable operation as defined by greenfox and SHACL. The second table is an aligned enumeration of some building blocks of the conceptual frameworks underlying greenfox and SHACL.

**Part 1. Alignment: validation model**

|  |  |
| --- | --- |
| **Greenfox operation** | **SHACL operation** |
| Validation of a file system  against a greenfox schema | Validation of a data graph  against a shapes graph |
| =  Union of the results of the  validation of the file system against all shapes | =  Union of the results of the  validation of the data graph against all shapes |
| Validation of a file system against a shape | Validation of a data graph against a shape |
| =  Union of the results of  all focus resources in the target of the shape | =  Union of the results of  all focus nodes in the target of the shape |
| Validation of a focus resource against a shape  =  Union of the results of the  validation of the focus resource against  all constraints declared by the shape | Validation of a focus node against a shape  =  Union of the results of the  validation of the focus node against  all constraints declared by the shape |
| Validation of a focus node against a constraint  = function(  constraint parameters ,  focus resource,  resource values? | Validation of a focus node against a constraint  = function(  constraint parameters ,  focus node,  property values? |
| Resource values =  XPath(resource) | foxpath (resource) | Property values =  SPARQL property path (node) |

**Part 2. Alignment: conceptual building blocks**

|  |  |  |
| --- | --- | --- |
| **Greenfox concept** | **SHACL** | **Remark** |
| Resource shape:   * Folder shape * File shape | Node shape | Common key concept: shape =  set of constraints for a  set of resources |
| Focus resource | Focus node | Common view: validation can be partioned into validation of a single resource against a single shape |
| Target declaration   * Foxpath expression * Literal file system path | Target declaration   * Class members * Subjects of predicate IRI * Objects of predicate IRI * Literal IRI (node target) | Difference: in greenfox a target declaration is essentially a navigation result, in SHACL it tends to be derived from class membership (ontological) |
| Resource value | Value node | Common view: non-trivial validation requires mapping resources to values |
| Mapping resource to value:   * XPath expression * Foxpath expression | Mapping resource to property:   * SPARQL property path | Common view: the mapping of a resource to a value is an expression |
| Value shape:   * XPath shape * foxpath shape | Property shape | Common view: usefulness of an entity combining a single mapping of the focus resource to a value with a set of constraints for that value |
| Constraint declaration   * Constraint component * Constraint parameters | Constraint declaration   * Constraint component * Constraint parameters | Common view: a constraint declaration can be thought of as a function call |
| Constraint component   * Signature * Mapping semantic | Constraint component   * Signature * Mapping semantic | Common view: a constraint component can be thought of as a library function |
| Constraint parameter   * atomic * structured | Constraint parameter   * atomic | Difference: in greenfox constraint parameter may have any degree of complexity |
| Extension language:   * XPath/XQuery expression * foxpath expression | Extension language:   * SPARQL SELECT queries * SPARQL ASK queries | Common view: extension of functionality is based on an expression language for mapping resources to values and values to a result |
| Mediatype integration:   * Common data model * Common navigation model | - | Difference: in contrast to SHACL, greenfox faces a heterogeneous collection of validation targets, calling for integration concepts |