### …5.4.2 Nested Assertions

The framework provides the ability to "nest" policy assertions. For domains with a complex set of options, nesting provides one way to indicate dependent or associated elements within a behavior (See Policy Framework, Section 3.1 [link]). The granularity of assertions is determined by the authors and it is recommended that care be taken when defining nested policies to ensure that the options provided appropriately specify policy alternatives within a specific behavior.

In particular, when assertion authors define nested assertions, it is important that the semantic of an empty policy element be defined.

The following design questions below can help to determine when to use nested policy expressions:

- Are these assertions designed for the same policy subject?
- Do these assertions represent dependent behaviors?

If the answers are yes to both of these questions then leveraging nested policy expressions is something to consider. Keep in mind that a nested policy expression participates in the policy intersection algorithm. If a requester uses policy intersection to select a compatible policy alternative then the assertions in a nested policy expression play a first class role in the outcome. If there is a nested policy expression, an assertion description should declare it and enumerate the nested policy assertions that are allowed. Additional information could be needed in the description to explain the relationship between the nested assertion semantics and the parent assertion. Assertion authors should be aware that the meaning of a nested assertion is always evaluated in the context of or relationship to its parent. This can be illustrated by the WS-Addressing assertions.

**Deleted:** might

…

There is one caveat to watch out for: policy assertions with deeply nested policy can greatly increase the complexity of a policy and should be avoided unless necessary.

**Deleted:** when they are not needed.