



Hosted by

**British  
Geological Survey**

NATURAL ENVIRONMENT RESEARCH COUNCIL

Sponsored by



# **Core and Extensions PubSub SWG lessons learned**

96th OGC Technical Committee

Nottingham, UK

Lorenzo Bigagli, CNR-IIA

17 September 2015

# “Extending” requirements



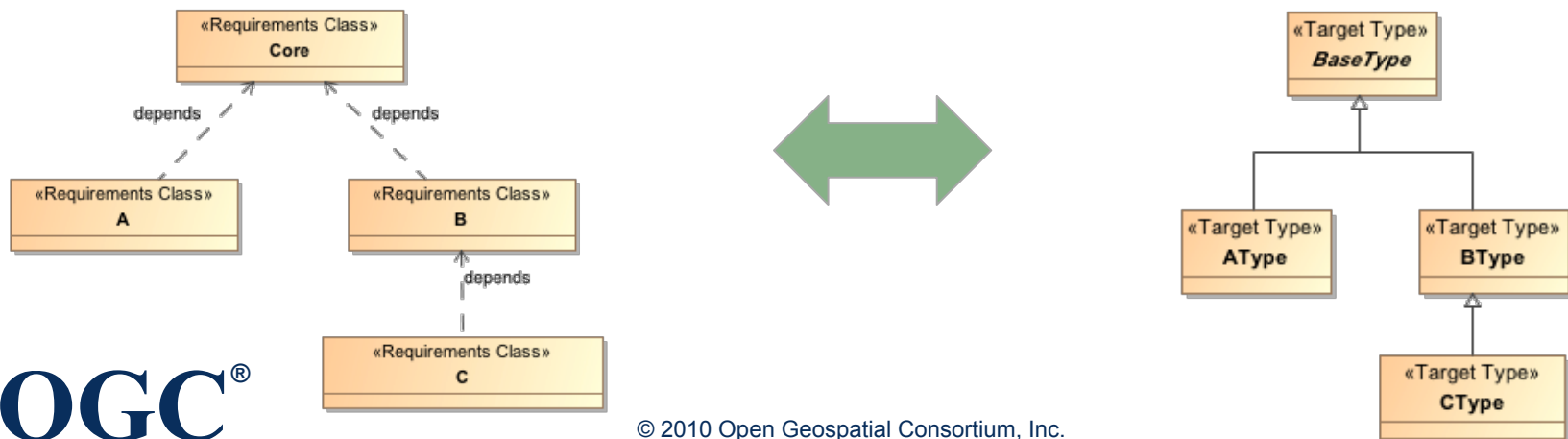
- Example:
  - In PSC Basic Publisher
    - “A Publisher shall offer the *Subscribe* operation ...”
  - In PSSB SOAP Basic Publisher
    - “A Publisher shall offer the **WS-BaseNotification NotificationProducer** *Subscribe* operation...”
- The requirement in the extension is more specific
  - Can it “override” the former?

| Requirement   |                                     |
|---|-------------------------------------|
| /req/soap/basic-publisher/subscribe   |                                     |
| Override  | /req/core/basic-publisher/subscribe |
| <b>Req 5</b> A <b>Publisher</b> shall offer the WS-BaseNotification NotificationProducer <i>Subscribe</i> operation |                                     |

# Semantics of requirements class extension



- Extensions adds requirements (**UML dependency**)
- Relationship between requirements classes and standardization target types (not necessarily 1-1)
  - Core (least requirements) corresponds to the most generic target type (**UML inheritance**)
- *Req 25 - A specification conformant to this standard shall never restrict in any manner future, logically-valid extensions of its standardization targets*



# “Negative constraints”

---



- But restrictive requirements are common
  - **Application schemas**
    - “Contents restriction is expected to be frequently used to restrict contents, in order to increase interoperability and reduce ambiguity when greater flexibility is not needed for applications” (OWS Common, §11.6.6)
  - **Profiles, enumerations**, anything with an **upper bound**, etc.
  - UML **specialization, cardinality**, etc.
  - Almost all requirements of **08-131r3!**
- Req 25 itself is a restriction
  - It forbids restrictions...

# Applying Req 25 to the ModSpec



= 08-131r3 itself (if it is coherent)

- *Req 25 - A specification conformant to this standard shall never restrict in any manner future, logically-valid extensions of its standardization targets*

= OGC standards

?!?



- *Req 25 – I shall never restrict in any manner future, logically-valid OGC standards*

# Conclusions

---



- Requirements class extension should be clarified
  - see generalization/specialization in UML
  - see the extension/restriction mechanism of the XML Schema Language
- Restrictive constraints should be allowed and inheritable just like any other
  - The OOPL notion of "final" (not-extensible) classes may be introduced
- Req 25 should be relaxed/clarified
- The example at page 20 should be amended

# Suggested ATS organization



- Annex with a table for each Conformance/Requirement Class, listing its requirements
- The table of a depending class references its dependencies
  - Related ETS tests may be executed serially
  - The ETS tests for a profile are the union of the tests related to (the transitive closure of) all the corresponding tables
- Possible extensions
  - A requirement in an extension may override one in a dependency
    - Cf. Req 18 (ib., §6.5.2)
  - A requirement in an extension may hide one in a dependency





Hosted by

**British  
Geological Survey**

NATURAL ENVIRONMENT RESEARCH COUNCIL

Sponsored by



# **Core and Extensions PubSub SWG lessons learned**

96th OGC Technical Committee

Nottingham, UK

Lorenzo Bigagli, CNR-IIA

17 September 2015



# BACKUP SLIDES

# Requirements satisfied by design

---



- Example:
  - In Core
    - Req xx - A PublicationType shall support provision of a human readable description of the PublicationType
  - In Extension XYZ
    - PublicationType provides for a "description" property of type xs:String
- Req xx is satisfied by design in Extension XYZ
  - Do we still need to test it?

# Semantics of requirements class extension



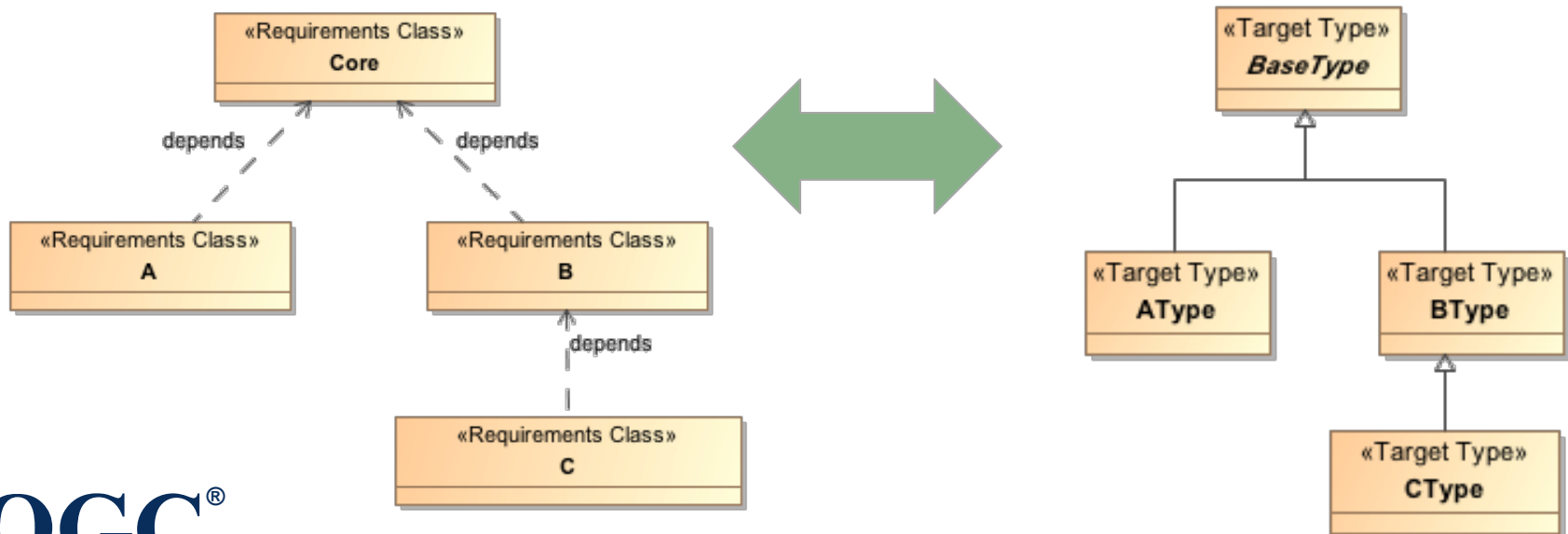
- In 08-131r3
  - "class B extends class A"  $\rightarrow$  B directly depends on A (ib., §4.9, §4.11)
  - The mandatory core class is a (possibly transitive) direct dependency of every class (ib., §6.5.4)
- Hence
  - any target certified for B is also certified for A
  - The core is the "widest" requirements class, corresponding to the most generic target type

# Semantics of requirements class extension



- Notes

- Requirements class extension is a dependency relationship
- Extensions can only add requirements
- Standardization target types “inherit from one another in the same way that UML classes do” (ib., §4.24)
- n-1 relationship between requirements classes and standardization target types



# Semantics of requirements class extension



- Example of core class
  - Ib., p. 20, top: *“For example, the core of a refactored GML might be the equivalent of “GML for Simple Feature” profile [06-049r1]”*
  - Example requirement of 06-049r1: *“Spatial properties are limited to being of type: point, linearly interpolated curve, planar surface, or aggregates thereof” (ib., §2.1)*
- Issue: core requirements must hold for every sub-class, i.e. must be obeyed by every standardization target (i.e., any GML document, including future extensions)
- (Apparent) Solution: this example is incorrect and 06-049r1 can’t be the core of a refactored GML
- Problem: actually, **no requirements class can capture 06-049r1**

# “Negative constraints”



- Or ‘restrictive requirements’, ‘restrictions’
  - Forbidden by the current conceptual framework
  - “Need to be reformulated to be more clearly extensible”
- The rationale of 08-131r3 is that a requirements class
  - “says what must be possible, but it cannot restrict what may be possible in the future”
  - “cannot restrict its extension”
- *Req 25 - A specification conformant to this standard shall never restrict in any manner future, logically-valid extensions of its standardization targets*

# “Negative constraints”

---



- But restrictive requirements are common
  - **Application schemas**
    - “Contents restriction is expected to be frequently used to restrict contents, in order to increase interoperability and reduce ambiguity when greater flexibility is not needed for applications” (OWS Common, §11.6.6)
  - **Profiles, enumerations**, anything with an **upper bound**, etc.
  - UML **specialization, cardinality**, etc.
  - Almost all requirements of **08-131r3**!
- Remarkably, Req 25 itself is a restriction
  - It forbids restrictions...



# “Negative constraints”

---



- ISO 19105 – Conformance and Testing
  - “[...] conformance requirements may be stated
    - A) **positively**: they state what is required to be done;
    - B) **negatively**: they state what is required not to be done” (ib., §6.3)
  - “A conforming implementation may support additional capabilities not described in the standard, providing those capabilities are not explicitly prohibited in the standard” (ib., §6.5)

# “Negative constraints”



- OGC 09-110 – WCS Core

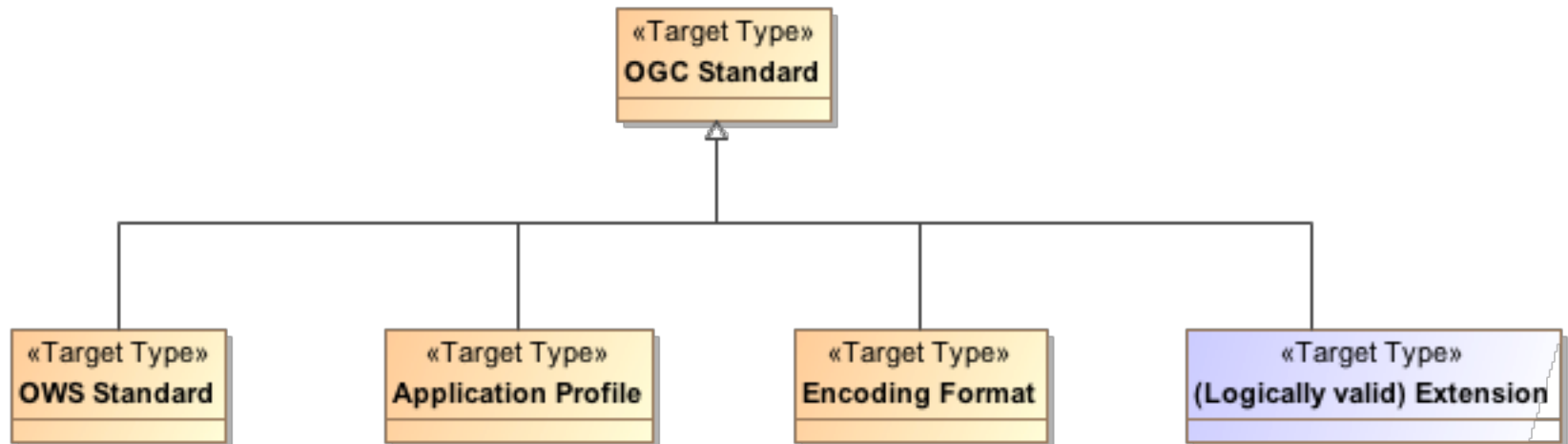
- Req 1: [...] Identifiers of coverages offered by a WCS server **shall** be immutable over the lifetime of the coverage identified, and not be reused for any other coverage on this service in future.
- Req 2: [...] supportedCrsList **shall** be empty (i.e., contain zero list elements).
- Req 3: For all WCS request types defined in this standard, the req version parameter **shall** have a fixed value of “2.0”.
- Req 22: The dimension value **shall** be equal to one of the dimension names specified in the coverage’s domain-Set [...]
- Req 23: A *GetCoverage* request **shall** contain at most one subsetting operation for each of the dimensions of the coverage addressed.

# 08-131r3 considered inconsistent



- 08-131r3 is “a standard for writing OGC standards” (ib., frontispice)
  - “considered one of its own standardization targets and thus a subject of its own requirements” (ib., §6.1)
  - Standardization targets = OGC standards
  - Any (logically-valid) extension of a target type is an instance of the core target

08-131r3 Target Type hierarchy (partial)



# 08-131r3 considered inconsistent



= 08-131r3 itself (if it is coherent)

- *Req 25 - A specification conformant to this standard shall never restrict in any manner future, logically-valid extensions of its standardization targets*

= OGC standards

# 08-131r3 considered inconsistent



- *Req 25 – I shall never restrict in any manner future, logically-valid OGC standards*



Write whatever  
you like!

# Conclusions

---



- Requirements class extension should be clarified
  - see generalization/specialization in UML
  - see the extension/restriction mechanism of the XML Schema Language
- Restrictive constraints should be allowed and inheritable just like any other
  - The OOPL notion of "final" (not-extensible) classes may be introduced
- Req 25 should be relaxed/clarified
- The example at page 20 should be amended

# Suggested ATS organization



- Annex with a table for each Conformance/Requirement Class, listing its requirements
- The table of a depending class references its dependencies
  - Related ETS tests may be executed serially
  - The ETS tests for a profile are the union of the tests related to (the transitive closure of) all the corresponding tables
- Possible extensions
  - A requirement in an extension may override one in a dependency
    - Cf. Req 18 (ib., §6.5.2)
  - A requirement in an extension may hide one in a dependency