Recommendations for Defining Conformance Requirement in Xt-EHR – Design Patterns

Purpose:
This document offers guidance for defining[1] "obligations" in the Xt-EHR specification, in light of ongoing changes to the original concept. This document has been prepared by the X-Net industry group and represents the collective consensus of the group. The goal is to steer the development toward a simple, manageable, and testable framework that ensures meaningful conformance without overregulating system behavior, targeting mainly HL7 FHIR. Audience for this document is Xt-EHR WP 8 Leadership, European Commission project officers for Xt-EHR and IHE-Europe/HL7 Europe IG development leadership.

General principles

- **Appropriate use of the different FHIR conformance qualifiers:** FHIR provides multiple mechanisms for defining constraints on resources, which increase in complexity and test efforts: cardinality, invariants, “must support” flag, and obligations (which have been introduced with FHIR R5). Only conformance to cardinality and invariants can be verified relatively effortlessly by validators. The others require potentially complex business logic testing, which in most cases cannot be implemented in an automated way that works across different vendors’ products. Therefore the more complex mechanisms should only be used when clearly justifiable and necessary.
  Use cardinality for defining how often an element can or has to appear.
  Use invariants to define logical constraints on resources which are always true for valid instances, e.g. birth date cannot be in the future, or: If the resource is contained in another resource, it SHALL NOT contain nested Resources. (Invariants need to be defined as expressions which can be validated by a FHIR validator.)
  Use “must support” to define that a creator shall populate the element if the information is known, and that a creator shall make use of the information if it is present.

[1] References / resources considered:
- Guidance for FHIR IG Creation: https://build.fhir.org/ig/FHIR/ig-guidance/
- IHE Technical Framework – General Introduction (Actor–Transaction–Content model): https://profiles.ihe.net/GeneralIntro/index.html
- W3C QA Framework - Specification Guidelines: https://www.w3.org/TR/qaframe-spec/
- ISO/IEC Directives, Part 2 (rules for normative statements and conformance): https://www.iso.org/sites/directives/current/part2/index.xhtml
- HL7 FHIR "Must Support" and "Shall Populate if Known" patterns: https://www.hl7.eu/obligations/

Use obligations only to mandate more complex behavior which is specific to a certain actor, e.g. Patient.birthDate is only required if the patient is human.

- **Distinguish between regulatory and quality enhancing conformance qualifiers:** the qualifiers which are required to be verified for getting market access need to be automatically testable through the member state operated test environments. Quality enhancing qualifiers (e.g. obligations) support manufacturers in fulfilling the needs, but often require product specific tests.
- **If obligations have to be used, keep them simple:** The obligations model should not become overly complex or burdensome. Focus on essential constraints that are clearly justifiable.
  Make use of the definitions in the FHIR obligation value set instead of creating own definitions.
  Use other type obligations beyond “SHALL:handle” only when clearly justifiable and necessary to ensure fulfillment of the EHDS goals. Otherwise they unnecessarily stifle product innovation. Or are not really conformance testable, like all “SHOULD” or “MAY” flavoured obligations.
- **Actor-specific obligations only:** Obligations should be clearly assigned to specific actors (e.g., the various producer, consumer actors). Question to be answered: should it be defined across all transactions using certain types of actors according to the classification and functional profiles of EHR systems which are defined in Xt-EHR deliverable 8.1 or should it be specified at the level of transaction and business actors specific to each data category, to avoid ambiguity.
- **Focus on interoperability, not UI behavior:** Do not regulate application design (e.g., what must be displayed). Obligations should concern data exchange, not user interfaces. Avoid creating additional workflow or documentation requirements for clinicians via cardinality or obligations for producers, except where absolutely needed to support EHDS goals.
- **Avoid untestable obligations:** Only include obligations that can be tested through automated or structured conformance processes. Avoid regulating runtime behavior that cannot be validated by the digital test environment operated by each member state.

Recommended Design Patterns

1. **Use "shall[2] populate if known" for optional fields**
   When data may not always be available but is valuable if known, use this obligation along with a 0..* cardinality. What is the difference with a must support requirement?
2. **Use cardinality only when the data is always required**
   Cardinality 1..* should be reserved for essential fields. Avoid setting cardinality in order to enforce business logic, which could create excessive Data Absent Reason (DAR) entries.
3. **Conservative approach for data consumers**
   For consumers, limit obligations to:
   a. Basic ability to receive and parse the document

[2] Conventions Used in this Guidance: "Shall" indicates a requirement, "Should" indicates a recommendation, "May" indicates a permission.

   b. Avoid requiring display, storage, or transformation logic

4. **Don’t regulate use cases beyond the exchange layer**
   Leave decisions on how to process or present data to implementers.
5. **Align with document/application context**
   Tailor data element constraints based on document content category (e.g., imaging, medication) and the application context. Avoid a one-size-fits-all approach.

Pitfalls to avoid

- **Field-by-field data element constraints assignment**
  Trying to define data elements constraints for every field leads to complexity and rigidity. Prioritize based on clinical and interoperability needs.
- **Overloading the conformance process**
  Avoid a data elements constraints framework that requires vendors to create and maintain extensive documentation or custom test suites just to demonstrate conformance.
  This would slow down innovation, delay product releases, and unnecessarily increase product prices.
- Conformance could combine testing, self-certification, and implementer assertions to reduce burden.
- **UI and workflow assumptions**
  Don't assume how users will generate content or how data will be used once received. Not all consumers will have UI capabilities; some may only store or extract specific data.

Next Steps

This document is intended as input to guide the redefinition of obligations within the Xt-EHR framework. We recommend limiting the initial implementation to a minimal, testable set of actor-specific obligations, using these design patterns to ensure clarity, flexibility, and interoperability.