4.3.8 Interoperability

Return to Non-Functional Requirements

- ☐ **[char]Please Review**

About

https://www.isko.org/cyclo/interoperability#3.1

Interoperability is a characteristic of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, in either implementation or access, without any restrictions. However, there are different levels of interoperability. The lowest most basic level of interoperability is at the network level. The degree of interoperability increases in difficulty as one moves up the interoperability levels. The most difficult level is the semantic level, at this level two assets can communicate with little or no prep work using common detailed formal, machine readable Ontologies including a vocabulary of terms.

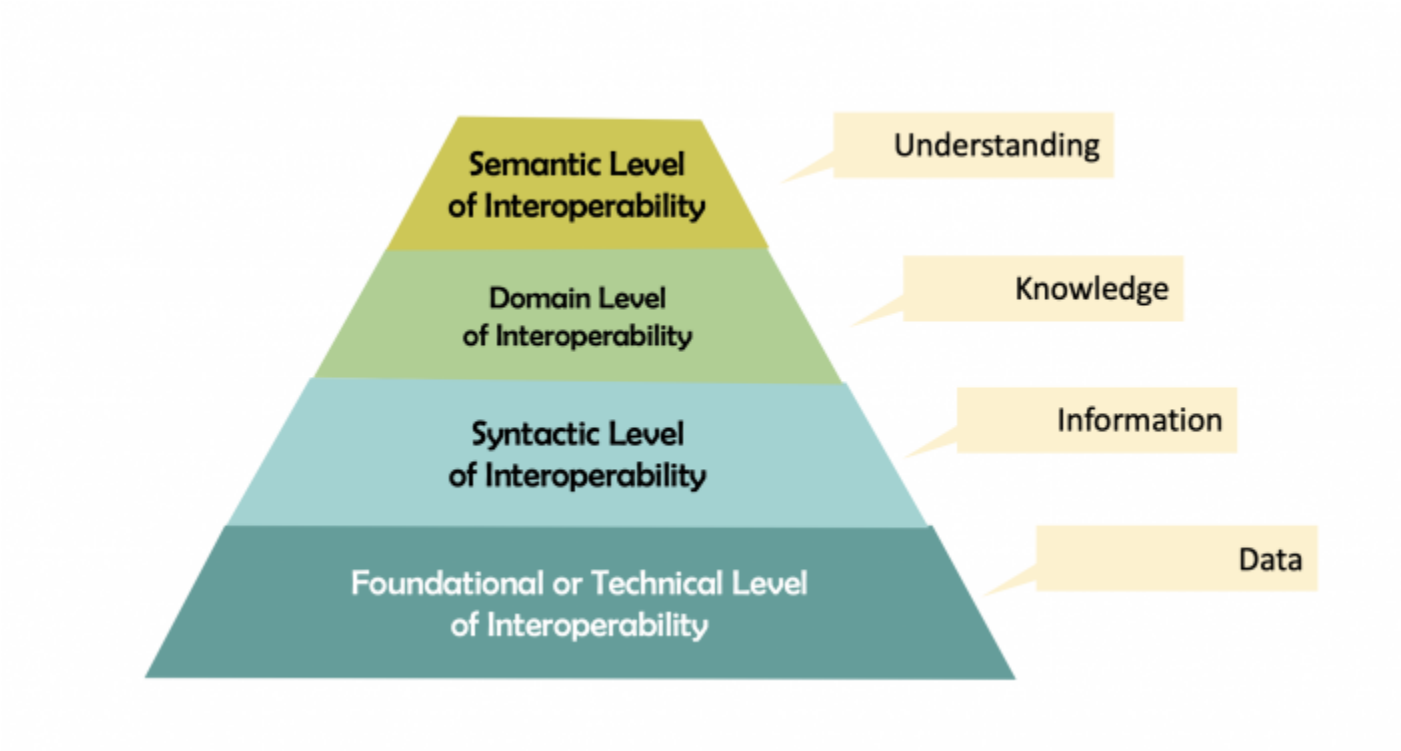


Figure 1: The Automation Pyramid and Interoperability

Foundational or Technical Level

return to Top

Foundational or Technical Interoperability Level is concerned with the most fundamental and basic kind

of interoperability. It provides the “foundation” on which the other kinds of interoperability are built upon. It identifies the technical aspects of peer-to-peer interoperability regardless of hardware, operating system, or middleware platforms. There are two perspectives to the Foundational or Technical Interoperability Level: Data and System.

- From the Data perspective, there is a clear, shared expectation for the contents, context and meaning of that data (e.g., Big-Endian or Little-Endian, or size of an integer, or character set used)
- From the system perspective the differences in computer technology are non-existent (e.g., 16-bit, 32-bit and 64-bit processors. Reduced Instruction Set Computer (RISC) versus Complex Instruction Set Computer (CISC) computers, etc.). In a Distributed System, the network protocols are another part of the Foundational or Technical Level of interoperability (e.g., Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) over Internet Protocol Address (IP Address), etc.).

Syntactic Level

return to Top

Syntax Level is concerned with the correct combination and sequence of the elements in a language and in this context, as it applies to Data Structure or in a Programming Language. For example, in English, we do not say `ball red large plastic`, we would say `large, red plastic ball`. The first form is syntactically incorrect, the second form is not. Interoperability at the syntactical Level means two different systems can exchange information and structurally equivalent. The interpretation of the syntactical data can be different (i.e., big to an ant is different than big to an elephant). The difference in interpretation is a semantic difference (see Semantic Level).

Domain or Structural Level

return to Top

Domain Level is knowledge of a specific, specialized subject area, discipline or field. For example, there is specific domain knowledge in Chemistry, Organic Chemistry, Physics, Mathematics, or Statistics. A specific example might be the use of the word `round`. There are completely different meanings for the word in Poker, Math, Cheese, and Military. This is in contrast to general knowledge, or domain-independent knowledge which in general would refer to the common usage of the word, sometimes referred to as dictionary usage of the word. The use of the word Domain or Domain knowledge is used within general purpose disciplines such as history, law, systems engineering or even computer science. People with knowledge in the generic area (i.e., law) might have specific domain knowledge in computers, electronics, computers or even specific computer architectures such s Reduced Instruction Set Computer (RISC).

Semantic Level

return to Top

Semantic Level is concerned with the meaning, relationship and restrictions on data and information described in the Domain or Structural and Syntactical Levels. At this level, computer systems unambiguously exchange information. The information that is exchanged might not be identical but there is a shared understanding between the two systems about the meaning of the information. Some examples of Semantic Interoperability are described by the ability of:[1)]:

- Two independent programs with reasoning capability arrive at the same conclusions from the same data
- An information system, without additional human intervention, to perform the tasks for which it was designed through the exchange of relevant data with other systems[2)], independently of how and what purpose this data was originally collected, stored and managed.
- One system to use data from an external source (without prior planning) and output useful (though not necessarily perfect) results.

DIDO Specifics

return to Top

☐ *TBD - to be added/expanded in future revisions of the DIDO RA*

1)

Ontology Taxonomy Coordinating WG / OntacGlossary, Accessed 10 July 2020, http://colab.cim3.net/cgi-bin/wiki.pl?OntologyTaxonomyCoordinatingWG%2FOntacGlossary

2)

Note: mediated by formally specific definitions and axioms

From:
https://omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
https://omgwiki.org/dido/doku.php?id=dido:public:ra:1.4_req:2_nonfunc:05_interoperability&rev=1622508108

Last update: **2021/05/31 20:41**