ETSI TS 103 378 V1.2.1 (2026-01)

TECHNICAL SPECIFICATION

Smart Body Area Networks (SmartBAN);
Unified data representation formats,
semantic open data model and corresponding ontology;
Associate service model/ontology/enablers extensions for SmartBAN semantic interoperability

Reference

RTS/SmartBAN-009r1

Keywords

application layer, control, health,
information model, interoperability, ontology,
security, semantic, service

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the
ETSI Search & Browse Standards application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on ETSI deliver repository.

Users should be aware that the present document may be revised or have its status changed,
this information is available in the Milestones listing.

If you find errors in the present document, please send your comments to
the relevant service listed under Committee Support Staff.

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure (CVD) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2026.
All rights reserved.

Contents

Intellectual Property Rights 5
Foreword 5
Modal verbs terminology 5
1 Scope 6
2 References 6
2.1 Normative references 6
2.2 Informative references 6
3 Definitions of terms, symbols and abbreviations 9
3.1 Terms 9
3.2 Symbols 9
3.3 Abbreviations 10
4 Introduction 12
5 Ambit and induced constraints 12
6 SmartBAN open data model and ontology 14
6.1 SmartBAN information analysis 14
6.2 SmartBAN core ontologies 20
6.2.0 Introduction 20
6.2.1 WBAN ontology 23
6.2.2 Nodes ontology 24
6.2.3 Process ontology 26
6.2.4 Ontofull ontology 27
6.2.5 OWL-DL formalization of SmartBAN core ontologies 28
6.3 Semantic interoperability and upper level model for SmartBAN 28
6.4 SmartBAN open service model and ontology 28
6.4.0 Introduction 28
6.4.1 SmartBAN ServiceProfile ontology 29
6.4.1.0 Introduction 29
6.4.1.1 Service Profile Model 29
6.4.1.2 Service Profile Ontology Description 31
6.4.2 SmartBAN ServiceProcess ontology 32
6.4.2.0 Introduction 32
6.4.2.1 Service Process Model 32
6.4.2.2 Service Process Ontology Description 34
6.4.3 SmartBAN ServiceGrounding ontology 35
6.4.3.0 Introduction 35
6.4.3.1 Service Grounding ontology description 36
6.4.4 SmartBAN Service ontology 36
6.4.5 OWL-DL formalization of SmartBAN Service ontologies 37
6.5 SmartBAN ontologies pre-validation 37
6.5.1 Introduction 37
6.5.2 WBAN for runners 38
6.5.2.1 WBAN Description 38
6.5.2.2 SWRL rules 39
6.5.2.3 SPARQL queries 40
7 SmartBAN modular ontologies mapping with of interest reference ontologies 40
7.1 Introduction 40
7.2 Main reference ontologies to be considered 41
7.2.1 Semantic Sensor Network (SSN/SOSA) ontology 41
7.2.2 oneM2M ontology 41
7.2.3 Smart Applications REFerence model (SAREF) ontology 41
7.2.4 SAREF ontology extension for the eHealth/Ageing Well domain (SAREF4EHAW) 41
7.3 SmartBAN core ontologies mapping with main reference ontologies 42

7.3.0 Introduction......42
7.3.1 Mapping of classes......42
7.3.2 Mapping of object properties......43

Annex A (informative): Background and SoA......45

A.0 Introduction......45

A.1 Existing sensor/actuator and BAN data representation......45
A.1.0 Introduction......45
A.1.1 Wireless Sensor Networks (WSNs) common used data representation models and ontologies......45
A.1.1.0 Introduction......45
A.1.1.1 OGC's Observations & Measurements (O&M) and SensorModel Language (SensorML)......46
A.1.1.1.1 Observations and Measurements (O&M)......46
A.1.1.1.2 Sensor Model Language (SensorML)......47
A.1.1.2 Existing WSN ontologies......49
A.1.1.3 OntoSensor ontology......50
A.1.1.4 Semantic Sensor Network ontology (SSN)......51
A.1.1.5 WSSN ontology......52
A.1.1.6 Semantic Web Based Architecture for Managing Hardware Heterogeneity......53
A.1.2 Proposed sensor ontologies in the eHealth sector......54
A.1.2.1 CEN TC 251 Health Informatics......54
A.1.2.2 IEEE™ reference models for medical device communications......55
A.1.2.3 Bluetooth LE (Low Energy) profiles for medical devices proposed by Continua Alliance......56
A.1.2.4 ASTM E31......57
A.1.3 Existing Semantic Web Service Architectures......58
A.1.3.1 OWL-S......58
A.1.4 Existing generic service layers and enablers for in particular WSNs and WBANs......60
A.1.4.1 Service Oriented Architecture for WSN......60
A.1.4.2 Semantic SOA for WSN......63
A.1.4.3 Open Sensor Web Architecture (OSWA)......65
A.1.4.4 W3C Web of Things (WoT) ontology: Thing Description......66
A.1.5 Ontology validation methods......67
A.1.5.0 Introduction......67
A.1.5.1 Common used symbolic-based method......67
A.1.5.2 Common used attribute-based methods......68
A.1.6 Preliminary conclusion concerning existing sensor ontologies......69

Annex B (normative): OWL-DL in Turtle formalization of the SmartBAN ontology......70

B.1 quantity.ttl......70

B.2 location.ttl......71

B.3 wban.ttl......73

B.4 node.ttl......79

B.5 process.ttl......87

B.6 ontofull.ttl......92

B.7 ServiceProfile.ttl (SmartBAN ServiceProfile module is called MyOntoServiceProfile ontology)......94

B.8 ServiceProcess.ttl (SmartBAN ServiceProcess module is called MyOntoServiceProcess ontology)......101

B.9 ServiceGrounding.ttl (SmartBAN ServiceGrounding module is called MyOntoServiceGrounding ontology)......106

B.10 Service.ttl (SmartBAN Service module is called MyOntoService ontology)......111

History......112

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

BLUETOOTH® is a trademark registered and owned by Bluetooth SIG, Inc.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Smart Body Area Network (SmartBAN).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies and formalizes SmartBAN unified data representation formats (including in particular sensor/actuator/relay/coordinator/Hub descriptions and sensed/measured data), semantic and open data model and corresponding ontology.

The present document addresses semantic interoperability. Therefore, the SmartBAN open data model and associated modular ontologies are extended with service/application level concepts, service modular ontology, and associated enablers. Furthermore, the linkage between SmartBAN modular ontologies and others of interest reference ontologies for smart BANs are also addressed in the present document.

The present document is applicable to a BAN and/or a Smart BAN comprising wearable sensor/actuator devices, a relay/coordinator device and a Hub. The relay/Coordinator and the Hub functionalities may be handled by a single device or by two distinct devices.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 103 325 (V1.1.1): "Smart Body Area Network (SmartBAN); Low Complexity Medium Access Control (MAC) for SmartBAN".

[2] ETSI TS 103 326 (V1.1.1): "Smart Body Area Network (SmartBan); Enhanced Ultra-Low Power Physical Layer".

[3] R. Hodgson, P. Keller, J. Hodges, J. Spivak: "QUDT - Quantities, Units, Dimensions and Data Types Ontologies", 2025.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith and A. Jamalipour: "Wireless Body Area Networks: A Survey", In IEEE™ Communications Surveys & Tutorials, issue 99, pp. 1-29, January 2014.

[i.2] J. Y. Khan and M. R. Yuce: "Wireless Body Area Network (WBAN) for Medical Applications", New Developments in Biomedical Engineering, Domenico Campolo (Ed.), 2010, ISBN: 978-953-7619-57-2.

[i.3] A. B. M. Moniruzzaman, S. A. Hossain: "NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison," International Journal of Database Theory and Application, Vol. 6, August 2013.

[i.4] I. döt Net, C. Evans, O. Ben-Kiki: "YAML Ain't Markup Language", 2011.

[i.5] ECMA-404: "The JSON Data Interchange Standard".

[i.6] Apache Avro™ ++version++ Documentation.

NOTE: Apache Avro™ is an example of a suitable product freely available. This information is given for the convenience of users of the present document and does not constitute an endorsement by ETSI of this product.

[i.7] IEEE 802.15.6 standard tutorial.

[i.8] J. Ison, M.Kalas: "EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats", 2025.

[i.9] Location owl ontology.

[i.10] M. Smethurst, R. Styles, T. Scott: "The Places ontology", 2007.

[i.11] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof and M. Dean: "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", W3C Recommendation, May 2004.

[i.12] D. P. Swain, K. S. Abernathy, C. S. Smith, S. J. Lee, S. A. Bunn: "Target heart rates for the development of cardio respiratory fitness", Med Sci Sports Exerc., 26(1):112-116, PMID:8133731, 1994.

[i.13] W3C® Recommendation, 19 October 2017: "Semantic Sensor Network Ontology", A. Haller, K. Janowicz, S. Cox, D. Le Phuoc, K. Taylor, M. Lefrançois, R. Atkinson, J. Lieberman, R. García-Castro, C. Stadler.

[i.14] ETSI TS 118 112 (V2.0.0): "oneM2M; Base Ontology (oneM2M TS-0012 version 2.0.0 Release 2)".

[i.15] ETSI TS 103 264 (V3.1.1): "SmartM2M;Smart Applications; Reference Ontology and oneM2M Mapping".

[i.16] ETSI TS 103 410-8 (V1.1.1): "SmartM2M;Extension to SAREF; Part 8: eHealth/Ageing-well Domain".

[i.17] ETSI TR 103 509 (V1.1.1): "SmartM2M; SAREF extension investigation; Requirements for eHealth/Ageing-well".

[i.18] S. Cox: "Observations and Measurements - XML Implementation" OGC document, 2011.

[i.19] ISO 19156: "Geographic information — Observations, measurements and samples".

[i.20] OGC® Sensor Web Enablement.

[i.21] OGC® Sensor Model Language (SensorML).

[i.22] R. Bendadouche, C. Roussey, G. De Sousa, J. Chanet and K. Mean Hou: "Extension of the Semantic Sensor Network Ontology for Wireless Sensor Networks: The Stimulus-WSNnode-Communication Pattern", In Proceedings of the 11th International Semantic Web Conference, Boston, USA, November 2012.

[i.23] M. Compton, C. Henson, L. Lefort and H. Neuhaus: "A survey of the semantic specification of sensors," In Proceedings of the 2nd International Workshop on Semantic Sensor Networks, Washington DC., USA, October 2009.

[i.24] D.J. Russomanno, C. Kothari and O. Thomas: "Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models," In Proceedings of 2005 International Conference on Artificial Intelligence, pp. 637-643, Las Vegas, NV USA, 2005.

[i.25] ISO 19115-1:2014: "Geographic information — Metadata — Part 1: Fundamentals".

[i.26] W3C® Semantic Sensor Network Incubator Group.

[i.27] S. Nicoliċ, V. Penca, M. Segedinac and Z. Konjoviċ: "Semantic Web Based Architecture for managing hardware heterogeneity in Wireless Sensor Network", International Journal of Computer Science Issues, Vol. 8, Issue 4, No 2, July 2011.

[i.28] CEN/TC 251: "Health Informatics".

[i.29] ISO 13606: "Health informatics — Electronic health record communication".

[i.30] ISO/IEEE™ 11073-20601: "Health informatics — Device interoperability — Part 20601: Personal health device communication — Application profile — Optimized exchange protocol".

[i.31] Bluetooth® Core Specification 4.2 document.

NOTE: Bluetooth® is an example of a suitable product available commercially.. This information is given for the convenience of users of the present document and does not constitute an endorsement by ETSI of this product.

[i.32] Recommendation ITU-T H.810: "Interoperability design guidelines for personal health systems".

[i.33] ASTM - Committee E31 on Healthcare Informatics.

[i.34] ASTM E1238-97 (Withdrawn 2002): "Standard Specification for Transferring Clinical Observations Between Independent Computer Systems".

[i.35] ASTM E1394-97 (Withdrawn 2002): "Standard Specification for Transferring Information Between Clinical Instruments and Computer Systems".

[i.36] ASTM E1467-94 (Withdrawn 2004): "Standard Specification for Transferring Digital Neurophysiological Data Between Independent Computer Systems".

[i.37] ASTM E2369-12 (Withdrawn 2021): "Standard Specification for Continuity of Care Record (CCR)".

[i.38] ASTM E1384-07 (Withdrawn 017): "Standard Practice for Content and Structure of the Electronic Health Record (EHR)".

[i.39] S. Tartir, I. Arpinar and A. Sheth: "Ontological Evaluation and Validation", Theory and Applications of Ontology: Computer Applications, Springer, pp. 115-130, 2010.

[i.40] W3C® Member Submission 22 November 2004: "OWL-S: Semantic Markup for Web Services".

[i.41] Lan, M. Qilong, J. Du: "Architecture of wireless sensor networks for environmental monitoring. In Education Technology and Training", in Proceedings of IEEE™ IITA-GRS 2008, Shanghai, China, December 2008.

[i.42] E. Avilés-López, J. García-Macías, J. Antonio: "TinySOA: a service-oriented architecture for wireless sensor networks", Service Oriented Computing and Applications. 2009. Vol. 3, n 2, pp. 99-108.

[i.43] F. C. Dedicato, P. F. Pires, L. Pirmez, T. Batista: "Wireless Sensor Networks as a service", In Proceedings of IEEE™ ECBS2010, pp. 410-417, UK, March 2010.

[i.44] A. Ghobakhlou, A. Kmoch, P. Sallis: "Integration of Wireless Sensor Network and Web Services", in Proceedings of the 20th International Congress on Modelling and Simulation, pp. 1-6, Adelaide, Australia, 2013.

[i.45] K. K. Khedo, R. K. Subramanian: "A service-oriented component-based middleware architecture for wireless sensor networks", International Journal of Computer Science and Network Security, 2009. Vol. 9, n 3, pp. 174-182.

[i.46] A. P. Singh, O. P. Vyas, S. Varma: "Flexible Service Oriented Network Architecture for Wireless Sensor Networks", International Journal of Computers Communications & Control. 2014. Vol. 9, n 5, pp. 610-622.

[i.47] F. Amato, V. Casola, A., Gaglione, A. Mazzeo: "A semantic enriched data model for sensor network interoperability", Simulation Modelling Practice and Theory. 2011. Vol. 19, n 8, pp. 1745-1757..

[i.48] Web of Thing Description CHU, Xingchen: "Open sensor web architecture: core services", The University of Melbourne, Australia, 2005.

[i.49] N. Guarino, C. Welty: "An overview of OntoClean", Handbook On ontologies, Springer, Chapter 8, 2009.

[i.50] V. Cross and A. Pal: "OntoCAT: An Ontology Consumer Analysis Tool and Its Use on Product Services Categorization Standards", Computer Science and Systems Analysis, Miami University.

[i.51] A. Lozano-Tello, A. Gomez-Perez: "ONTOMETRIC: A method to choose the appropriate Ontology", Journal of Database Management, 2 (15) pp. 1-18, 2004.

[i.52] W3C® Recommendation 22 August 2024: "Data Catalog Vocabulary (DCAT) - Version 3".

[i.53] D. Brickley: "W3C Semantic Web Interest Group - Basic Geo (WGS84 lat/long) Vocabulary".

[i.54] L. Nachabe, F. Hannou, M. Lefrançois and M. Jubault: "Toward Agile Interaction Model based ontology development Methodology (AIME) for FAIR European data spaces", FAIR Principles for Ontologies and Medatata in Knowledge Management (FOAM), Jul 2024, Enschede, Netherlands.

[i.55] DCMI Metadata Terms.

[i.56] VANN: "A vocabulary for annotating vocabulary descriptions".

[i.57] ETSI TR 103 394: "Smart Body Area Networks (SmartBAN); System Description"..

[i.58] ETSI TS 103 378 (V1.1.1): "Smart Body Area Networks (SmartBAN) Unified data representation formats, semantic and open data model".

[i.59] Web of Thing Description.

[i.60] SAREF extension for environment.

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

example 1: text used to clarify abstract rules by applying them literally

3.2 Symbols

For the purposes of the present document, the following symbols apply:

bpm beats per minute
bps bit per second
s second

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

3G	Third Generation (réseau mobile)
AC	Alternating Current
ANP	Alert Notification Profile
API	Application Programming Interface
ASTM	American Standards for Testing and Materials
BAN	Body Area Network
BLE	Bluetooth Low Energy
CCR	Continuity of Care Record
CCU	Central Control Unit
CEN	European Committee for Standardisation
CESN	Coastal Environmental Sensor Networks
CoAP	Constrained Application Protocol
CRS	Coordinate Reference Systems
CSIRO	Commonwealth Scientific and Industrial Research Organisation
CSV	Comma Separated Values
DCAT	Data Catalog Vocabulary
DL	Description Logic
DT	Digital Twin
EC	European Commission
ECG	Electrocardiogram
EDAM	Embrace Data and Methods
EEG	Electroencephalogram
EHAW	eHealth/Ageing Well
EHR	Electronic Health Record
EWS	Early Warning System
FAIR	Findable, Accessible, Interoperable, and Reusable
FOAF	Friend Of A Friend
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GPS	Global Positioning System
GW	Gateway
HID	Human Interface Device
HL7	Health industry Level 7
HOGP	HID Over GATT Profile
HR	Heart Rate
HTTP	Hypertext Transfer Protocol
IC	Integrated Circuit
ICT	Information and Communications Technology
IEEE	Institute of Electrical and Electronics Engineers
IOPE	Input, Output, Precondition and Effect
IoT	Internet of Things
IP	Internet Protocol
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
JSON-LD	JSON Linked Data
KB	Knowledge Base
LAN	Local Area Network
LE	Low Energy
LSDIS	Large Scale Distributed Information Systems
MAC	Media Access Control
MAXRAM	Maximum Random Access Memory
MBAN	Medical Body Area Network
MIPS	Microprocessor without Interlocked Pipeline Stages
MMI	Marine Metadata Interoperability
MQTT	Message Queue Telemetry Transport

NASA	National Aeronautics and Space Administration
NICTA	National ICT Australia Ltd
NoSQL	Not only SQL
NT	Node Type
O&M	Observations & Measurements
OGC	Open Geospatial Consortium
OpenGIS	Open Geospatial Consortium
OSI	Open Systems Interconnection
OSWA	Open Sensor Web Architecture
OUI	Organizationally Unique Identifier
OWL DL	Web Ontology Language Description Logic
OWL	Web Ontology Language
OWL-S	Web Ontology Language for Services
P2P	Peer-to-Peer
PDA	Personal Digital Assistant
PER	Packet Error Rate
PHD	Personal Health Device
PHY	Physical layer
PM	Project Manager
QoI	Quality of Information
QoS	Quality of Service
QUDT	Quantity, Unit, Dimension and Type
RAM	Random Access Memory
RDF	Resource Description Framework
ROM	Read-Only Memory
RTLS	Real Time Location Services
SAREF	Smart Applications REFerence model
SAREF4EHAW	SAREF ontology extension for the eHealth/Ageing Well domain
SCS	Sensor Collection Service
SDO	Sleep Domain Ontology
SEEK	Science Environment for Ecological Knowledge
SensorML	Sensor Model Language
SIG	Special Interest Group
SMS	Short Message Service
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOS	Sensor Observation Service
SOSA	Sensing, Observation, Sampling and Actuation
SOSA	Sensor, Observation, Sample, and Actuator
SPARQL	SPARQL Query Language
SPP	Scan Parameters Profile
SPS	Sensor Planning Service
SQL	Structured Query Language
SSN	Semantic Sensor Network
SUMO	Suggested Upper Merged Ontology
SWAMO	Sensor Web for Autonomous Mission Operations
SWE	Sensor Web Enablement
SWRL	Semantic Web Rule Language
TBD	To Be Defined
TC	Technical Committee
TCP	Transmission Control Protocol
TD	Thing Description
TR	Technical Report
TTL	Time To Live
UDDI	Universal Description, Discovery and Integration
UDP	User Datagram Protocol
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
US	United States
UUID	Universally Unique IDentifier
UWB	Ultra Wide Band

W3C	World Wide Web Consortium
WBAN	Wireless Body Area Network
WBANID	Wireless Body Area Network Identifier
WNS	Web Notification Service
WoT	Web of Things
WSDL	Web Services Description Language
WSN	Wireless Sensor Network
WSSN	Wireless Semantic Sensor Network
XML	eXtensible Markup Language
YAML	Yet Another Markup Language

4 Introduction

The present document gives the high level description of a unified data format providing one of the building blocks for heterogeneity management in Smart BANs. The present document will in particular define a unified description model with an extensible semantic metadata for Smart BAN entities related data as well as for monitoring and control information. The corresponding metadata management and indexing framework will also be investigated.

5 Ambit and induced constraints

The scope of clause 5 of the present document is to briefly investigate the initial TC SmartBAN use case requirements in order to point out their impact on the present document specifications and designs. The initial additional requirements induced by the present document scenario will also be listed.

Wireless Body Area Networks (WBANs) art expected to be operated in licensed frequency bands. Hence, the frequency spectrum of operation will be in the unregulated frequency bands for Industrial, Scientific and Medical (ISM) applications. If ISM and MBAN bands (US and European) with frequency between 2,3 GHz and 2,5 GHz are initially considered within TC SmartBAN, higher frequency bands (from 3,2 GHz to 10,2 GHz) will also be considered for allowing the support of Real Time Location Services (RTLS). Finally, WBANs are characterized by strong constraints in terms of low power, low latency, low Packet Error Rate (PER), reliability, QoS, coexistence and security. The initial technical requirements retained by TC SmartBAN for WBAN parameters are listed in Table 1.

Table 1

Parameter	Smart BAN Requirements
Coexistence/robustness	Good (low interference to other systems, high tolerance to interference)
Data rates (Sensor)	Nominally < 100 kbps/node (vital sign monitoring)
Transmission rate (PHY)	Up to 1 Mbps
Network topology	Star network
Power consumption (node)	TBD
QoS control	Priority based control and cross layer optimization. Emergency signal transmission supported.
Reliability	Robust to shadowing and multipath interference
Max. node capacity	up to 16 nodes (typically 8)
Range	< 1,5 m
Latency	< 125 ms (high sampling applications e.g. ECG, EEG)
Security / privacy	TBD

The initial ambit envisioned by TC SmartBAN contributors is a BAN network organized around a Hub and mainly following a star topology. The Hub plays the role of the BAN cluster head and also serves as an intermediary Gateway (GW) node allowing the interconnection of the BAN cluster with an healthcare local/remote monitoring and control center. This node, with extended memory and processing capacity (e.g. a smart phone), should be responsible for all the heavy processing management and control operations of the SmartBAN. In case of a multi hop routing strategy, the BAN shall be provided with at least a bridge/relay functionality that could be handled by the SmartBAN's Hub or within a dedicated SmartBAN device. This relay/bridge device offers enhanced performance and robustness (e.g. relay around hidden devices), as well as optimized SmartBAN solutions with enhanced connectivity (multi-radio) and routing (multi-hop) capabilities. In some global healthcare architectures, the BAN's Hub role may sometimes be handled by a cluster-external intermediary node called Central Control Unit (CCU) [i.2]. Finally, BAN discovery functionality and interworking shall be handled and shall thus be taken into consideration for the present document of the SmartBAN semantic open data model. Figure 1 gives a simple example of the considered SmartBAN end-to-end architecture.

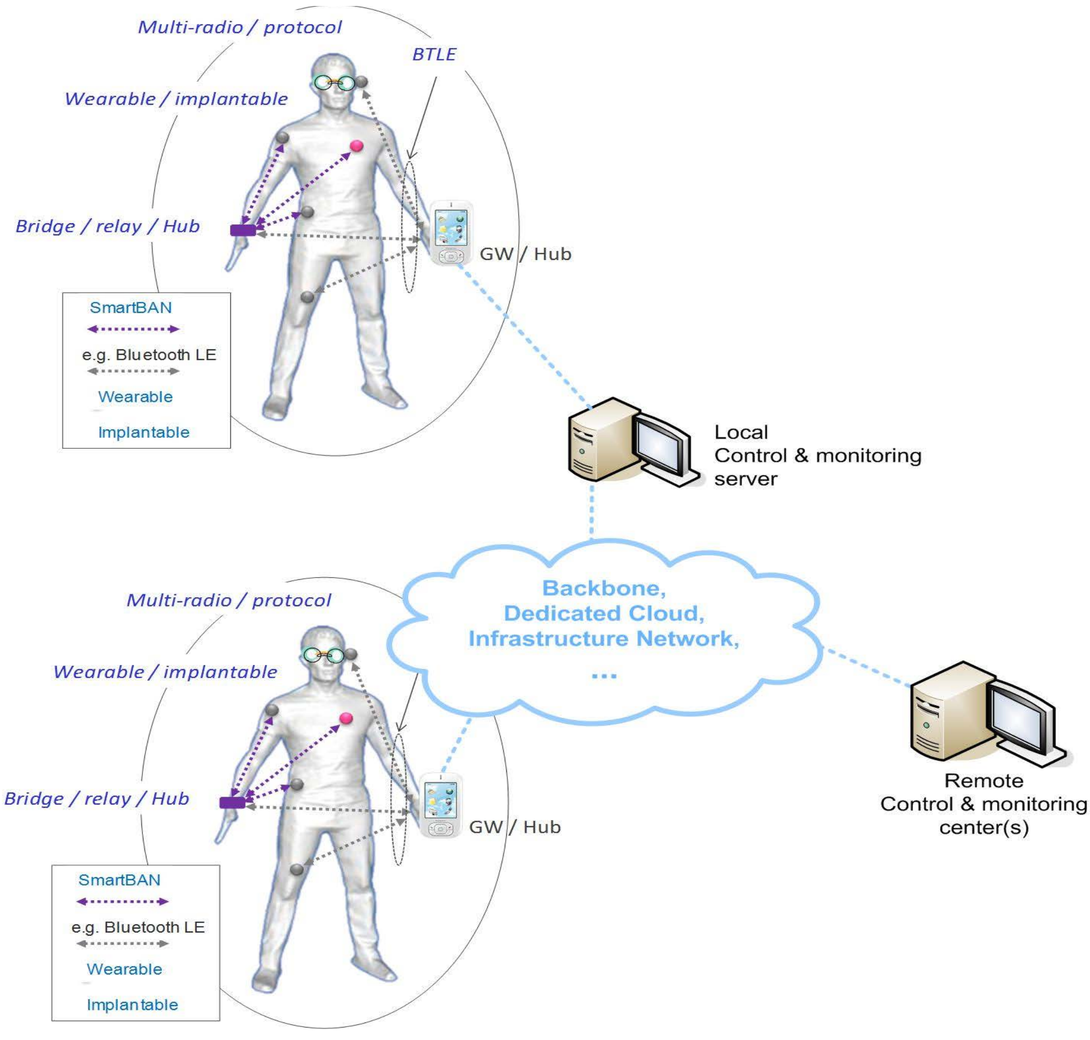


Figure 1: Example of considered SmartBAN end-to-end architecture

One main objective of the present document is the BAN heterogeneity management through the specification of a generic sensor and sensor data description and transfer format. This description format shall be as rich as possible to allow e.g. conflict resolution or similarity detection, but shall also be handled with low processing, low power and in quasi real time (e.g. latency < 125 ms and node addition or removal time < 3 s [i.1]). In that context, two scenarios shall a priori be envisioned:

- Or the proposed ontology is sufficiently light for allowing the sensor raw data pre-processing within the sensor itself.
- Or the pre-processing is deported to most powerful BAN's nodes. In that case, aforementioned nodes such as the BAN Hub or the BAN coordinator seem a priori good candidates for handling that pre-processing.

Furthermore, the ontology pre-processing and processing operations complexity strongly depend on the ontology format and storage format used (e.g. NoSQL [i.3] compared to SQL, or -YAML [i.4], JavaScript Object Notation [i.5], Extract Load Transfer strategies and Avro Schemas [i.6], etc. - compared to in particular XML, could significantly reduce the processing power required).

The aforementioned issues will be addressed in the present document.

6 SmartBAN open data model and ontology

6.1 SmartBAN information analysis

The objective of clause 6.1 of the present document is first to perform information analysis of the SmartBAN scenarios and to identify:

- the relevant related information sources, types, formats and owners;
- the information processing operations and functionality towards the SmartBAN related information.

Then, based on the analysis of existing information models already investigated in the SoA clause (see clause A.1 of the present document), the SmartBAN information meta-model (UML diagram) will be defined in clause 6.1 of the present document.

Dealing with tiny devices in WBAN shall lead towards the design of a modular model where some classes should be implemented in the sensor/actuator nodes depending on its resource's availability, and some others should be implemented and processed in more capable nodes like e.g. the SmartBAN' Hub or coordinator. Basically, the SmartBAN open data model shall be divided into three main parts:

- WBAN (SmartBAN or BAN cluster in the TC SmartBAN context);
- Nodes (i.e. Hub, sensors, actuators, etc.);
- Process and Measurements.

Figure 2 depicts the class diagram that shall be introduced for the WBAN/Smart BAN.

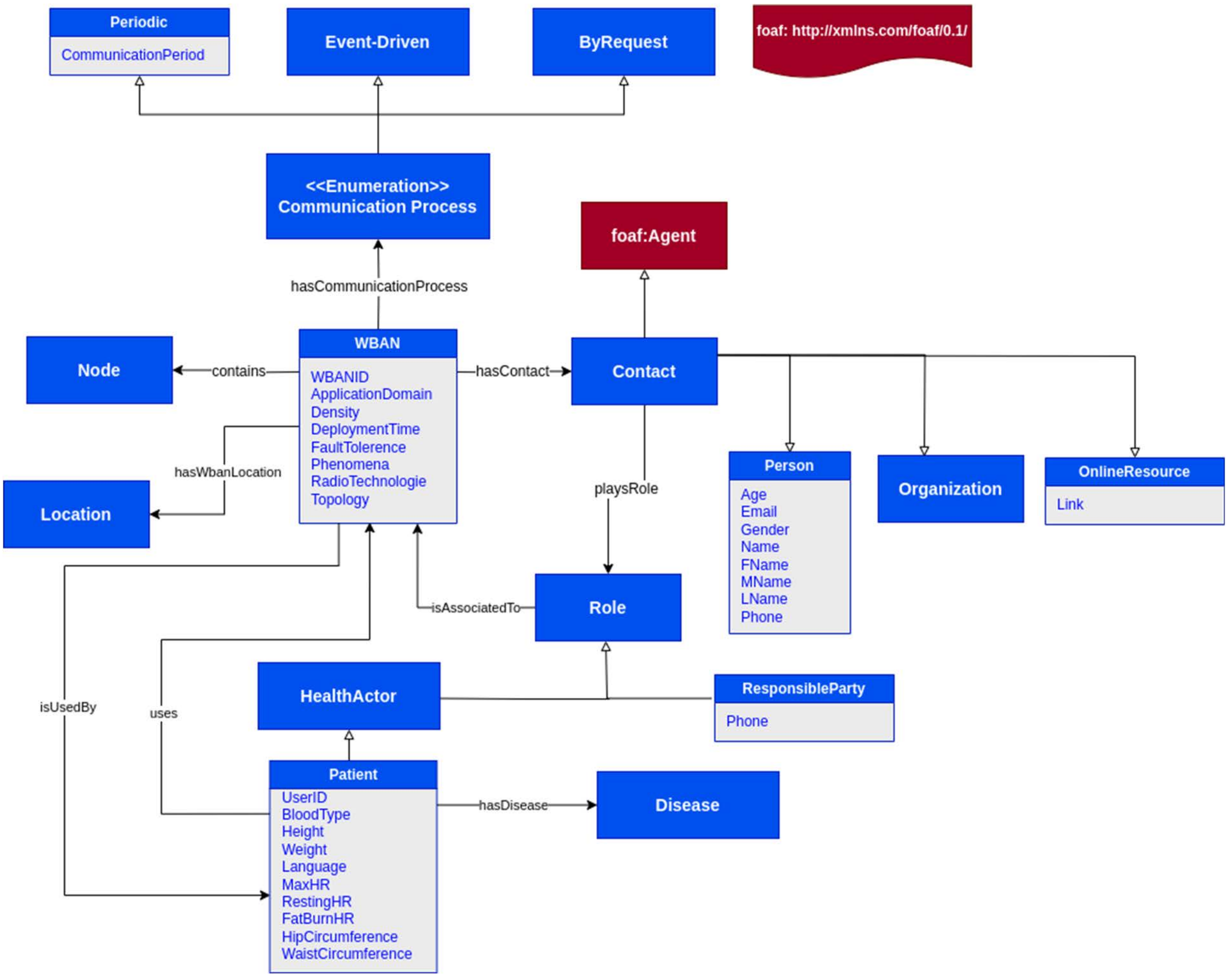


Figure 2: UML Class Diagram- WBAN

First, a WBAN shall be identified by its WBANID that shall be unique (e.g. the 8-bit BAN ID defined within the TC SmartBAN MAC TS [1]) and accessible by any authorized user. It is deployed on the human body or implanted in the human body. Thus, the location of the WBAN, when available and needed, shall be given relatively to the one of the human body can refer to the location of the patient wearing the WBAN. The WBAN will monitor a specific phenomenon (burned calories during exercises, glucose level, etc.) belonging to a specific domain of application (healthcare, telemedicine, assisted living, sport training, pervasive computing, safety and emergency, etc.). The WBAN is composed of certain number of nodes (WBAN's density). Theses nodes are distributed based on a physical topology (star in the special case of SmartBANs). The required lifetime and the fault tolerance differ from one application to another. Whereas WBAN for entertainment purposes should have a lifetime of weeks or few years, WBAN dedicated for assisted living or anomaly monitoring should last for many years. In addition a WBAN shall measure accurate value and shall require a small fault tolerance. This parameter is essential in the management of the networks where the fault tolerance shall be respected and may imply nodes reconfiguration where some nodes are considered down due to the inaccurate sensed data. The data within the WBAN will be exchanged using in particular SmartBAN radio technology defined in ETSI TS 103 326 [2] and ETSI TS 103 325 [1].

Each WBAN shall have contacts, and this assumption still holds for SmartBANs. In SensorML model, a contact can be a person described by his name, userID, email and phone number. A contact can also be an organization described by its name and address. In the FOAF (Friend of a Friend) data model, a foaf:Agent class is a central concept to describe people, their activities, and their relationships on the web. A foaf:Agent can be a foaf:Person (individuals, people) or a foaf:Organization (structured groups like companies, institutions). In our model, a contact shall be a foaf Agent. The agent shall be a person or an organization. A contact shall play a role. The role shall be:

- An online resource or document, and in that case, a reference (link) to the document is simply given as the class attribute.

- Or a Responsible Party which plays the role of the patient's reference. It should be an organization or a person. The creation of the responsible party is very useful for emergency cases. For example, if the WBAN is developed for the monitoring of elderly, in urgent cases, the system can generate a notification and send it to the responsible party's mobile phone.
- Or a health actor. The health actor shall be the patient or any other health reference like the doctor or the nurse. In the context of TC SmartBAN, the WBAN may have as contact one patient. The Continua® Alliance (see note), (see clause A.1.2.3 of the present document), describes the patient information in the user data service within the Bluetooth LE standard by patient's full name, weight, height, email, date of birth, gender, resting heart rate, waist circumference, hip circumference, fat burn heart rate lower limit, fat burn heart rate upper limit and language (these patient's data shall be added to the patient's attributes of the SmartBAN open data model). In order to enhance the management of electronic health data shared between medical systems, the diseases, blood type, allergy (a type of disease) or any other medical descriptions shall be added to this information. The patient shall also be identified by a unique ID that may be his user ID or social security number. Sharing these data among different biomedical systems allow any clinic or hospital to identify the patient and deduce his medical archive.

NOTE: Continua® is an example of a suitable product available commercially. This information is given for the convenience of users of the present document and does not constitute an endorsement by ETSI of this product.

Whereas the WBAN can be formed by many clusters, the present document is focused on the topology shown in Figure 2 where a SmartBAN shall be formed of only one cluster. The present document can be extended to multi cluster topology where the cluster shall be identified by the Cluster ID. If Bluetooth LE is used for wireless communication, the channel for each cluster shall be saved in addition to the number of nodes and the topology used within the cluster. Finally, the SmartBAN shall contain nodes.

In WBAN architecture, three types of communication shall be differentiated: Intra WBAN communication between the nodes and the Hub, Inter WBAN communication between the Hub and the internet and beyond WBAN communication with the medical servers and health care systems on e.g. a cloud. The inter WBAN communication shall obey to one of these three ways. The first type is the periodic communication where the nodes send data to the BAN coordinator (i.e. the cluster Hub) every *T* seconds. The second one is based on BAN coordinator requesting BAN nodes for triggering their reply. The third one is based on event-triggering. When certain events occur, the nodes send their data to the BAN coordinator. That is why the inter WBAN communication type shall be precise to enhance remote management. For example, if periodic exchange is used, and the hub did not receive data during T seconds, it should identify that this node is malfunctioning.

The description of the SmartBAN node's model is now presented. Figure 3 represents the UML class diagram that shall at least be specified for the SmartBAN's nodes. The node shall be a sensor, an actuator which acts according to data received from the sensors or through interaction with user, a node responsible of data routing or pre-processing like a BAN coordinator or relay, a cluster head (BAN's hub), a wearable, or others. As an example, an actuator equipped with a built-in reservoir and pump can administer the correct dose of insulin to give to diabetics based on the glucose level measurements. The node may be a cluster head (BAN's hub) which is responsible for collecting the data and for sending it outside the WBAN to a local or remote monitoring and control centre. In some cases, the coordinator is called aggregator, collector or body centre unit. In the SmartBAN model, the coordinator is called the Hub. In many cases especially in m-health applications, the Smart Phone or the PDA of the user will play this role. It will collect the data from the sensors via Bluetooth interfaces and send it to the medical servers via 3G/4G or wireless interfaces.

Moreover, each node shall be identified by the Node ID that may be the MAC address, the serial number or any other unique descriptor. A new class named "NodeType" and describing the physical characteristics of a node (as described in the SoA clause A.1.1.2 of the present document) has been introduced within the SmartBAN data model for redundancy avoidance. Each node shall have only one node's type. The survival properties, which describe the conditions to which a sensor can be exposed without causing lasting damage, shall also be added. When these properties occurred on a network, all data sent from this node should be considered as insignificant data and trigger the stopping of this node.

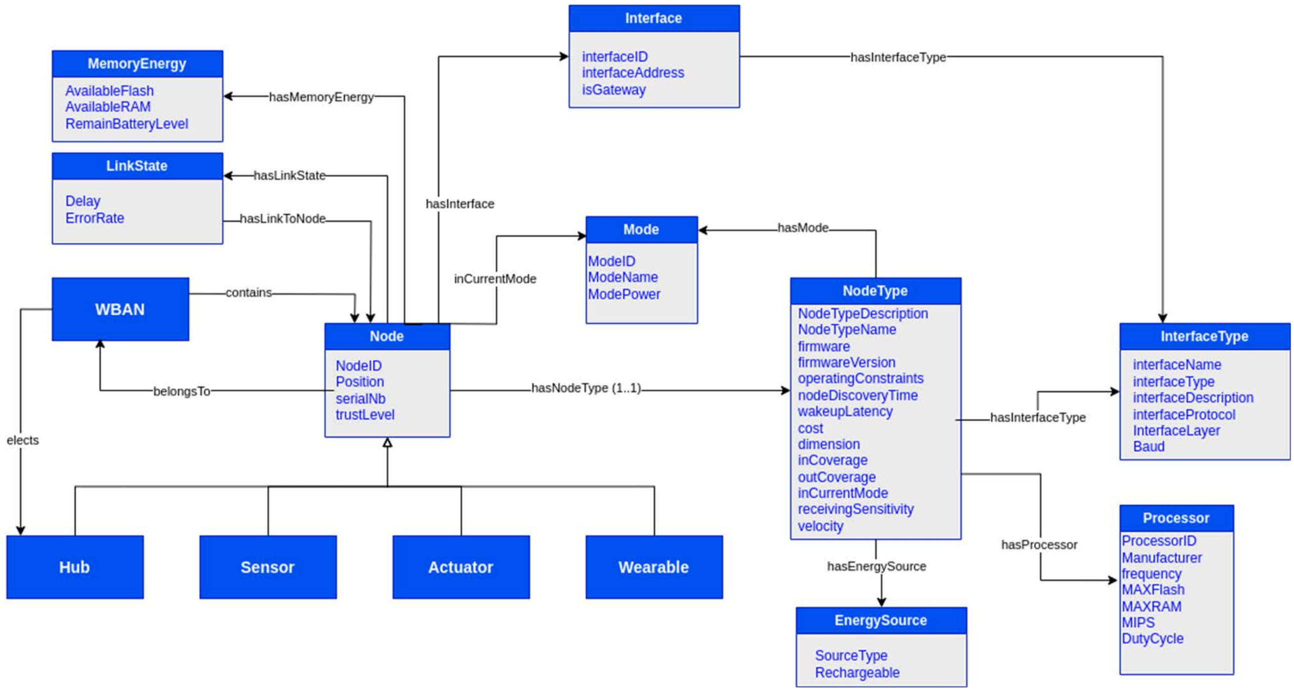


Figure 3: UML Class Diagram- Nodes

The mobility also enhances the performance of the sensor networks. Basically, the nodes in WBAN have mobility capability. Thus, the velocity of a mobile sensor, the energy consumption during its movement and the maximum distance that it can travel should be registered in the SmartBAN node model. This feature will help in collecting data. Many times it will be more efficient for the network that a mobile sensor travels over the network to collect data or send request or reconfigure certain nodes. Each node type has certain functioning modes (Sleep, active, down, transmission, reception, etc.). Keeping in mind the power management of the WBAN (SmartBAN low energy constraint), the power consumed in each mode is helpfull in estimating the lifetime of the WBAN and shall be registered in the SmartBAN node model.

EXAMPLE: If the communication within the WBAN is a periodic communication, it will be useful to switch the node from active mode to sleep mode during a period t (since no data will be sent during that period), thus permitting a reduction of power consumption on the node.

Each node type shall have wakeup latency. The wakeup latency is the time required by the sensor to generate a correct value once activated. Clearly, if the sensor reading is performed before the wakeup latency has elapsed, the acquired data is not valid. Furthermore, each node type shall have a processor identified by a processor ID. The processor shall have RAM, ROM and a flash memory. The node with higher memory resources has the priority to be elected as cluster head or a BAN coordinator. In addition, the MIPS, frequency and duty cycle shall be introduced in the SmartBAN node model because they reflect the speed of processing certain data.

While the nodes are typically powered by batteries with a limited lifetime, they should benefit from additional energy harvested from the external environment. Recently, many researches have been investigated in the domain of energy scavenging from on-body sources such body heat and body vibration. Where the source of energy for the same node can vary, the source of energy shall be revealed in the SmartBAN open data model.

Data transmissions within a WBAN are via interfaces. Each node type may have many interfaces (e.g. SmartBAN PHY & MAC TS, Bluetooth, UWB, IEEE 802.15.6 [i.7], serial interface, etc.). A new class "interface type" that describes the standard characteristics of the interface's protocol, the baud rate and the functional layer shall be added for redundancy avoidance. Regarding the interface, it shall be identified by the interface ID that may be the IP address or the unique address used in the communication protocol. The interface may have many addresses like MAC address, IP address or others. One shall precise if the interface is considered as gateway in order to help the monitoring and control server in finding how it can communicate with the WBAN from its LAN, backbone, or infrastructure side.

The dynamic characteristics of a node that vary during the WBAN lifetime are now presented. First, each node shall have only one current mode. For example, if a BAN coordinator has sent a request to collect certain information, only nodes who are in Active mode can reply. Moreover, the patient's information shall be always available to the physician. By tracking the current mode of each node, in addition to the features mentioned before, one may choose how the switching process shall be done. The node occupies certain position, it can be planted in the human body, on the body surface (head, waist, chest, wrist, etc.), or not in contact with the human body with a distance of few meters [i.1].

Moreover, security is a key point in WBAN management, that is why the Trust Level for each node shall be added within the SmartBAN data model.

The dynamic memory and energy components of the nodes shall be described in the "MemoEnergy" class. The remaining lifetime or memory may influence directly on the selection of the BAN Hub or coordinator, especially that the hub's death may cause the malfunctioning of the entire WBAN.

Finally, a crucial issue is the reliability of the transmission in order to guarantee that the monitored data is received correctly and in reasonable time. In general, the Hub is the node who can monitor other nodes to predict any failure. To assure the reliability, the link state for the path between nodes shall be added to the SmartBAN data model for indicating the last time the Hub has received a message from a node. If this time is too long, the Hub can therefore predict that this node is facing some trouble. In addition, it will keep track of the error rate of a node and its delay to consider a node as invalid and asks it to go to the inactive mode in order to reduce bandwidth consumption.

The third parts of the SmartBAN open data model shall be provided with the process classes presented in Figure 4.

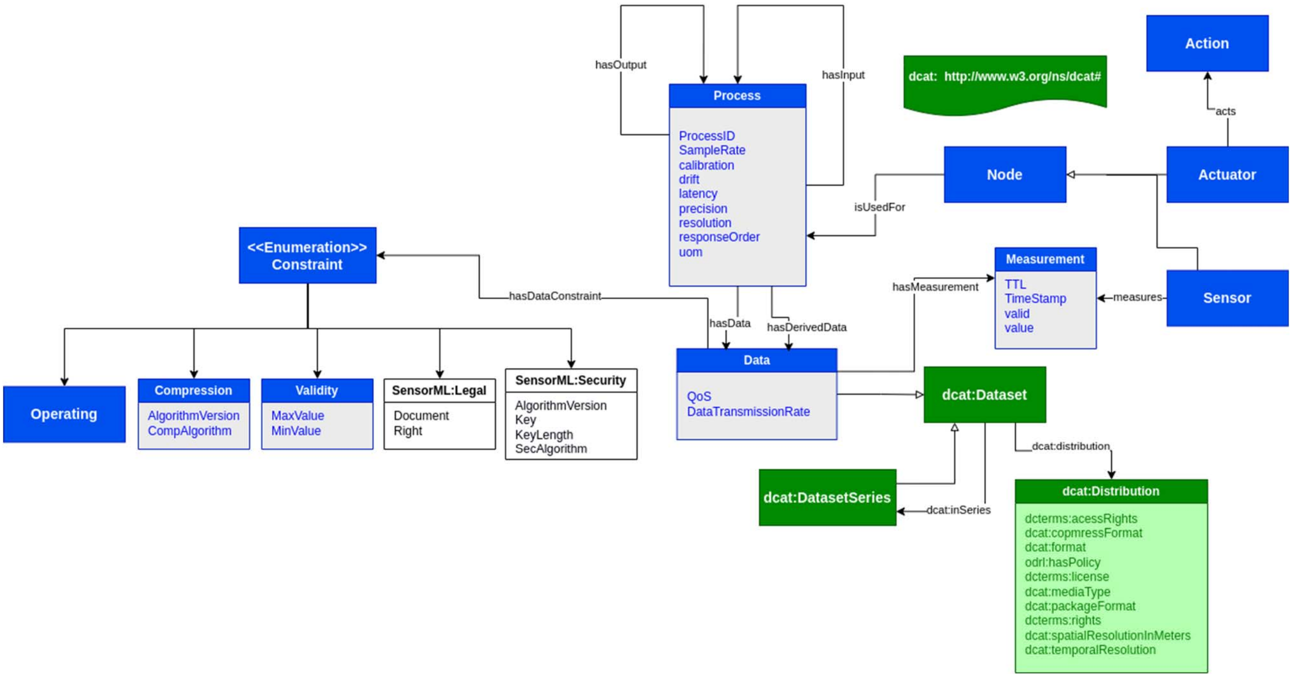


Figure 4: UML Class Diagram- Process & Measurements

Each node is used for certain process (temperature, blood pressure, insulin regulation, etc.). While sensors measure some measurements, actuators do certain actions. Based on the survey done in the survey clause A.1.1 of the present document, the following properties shall be added to the process class that shall be introduced to the SmartBAN open data model: calibration, precision, response order, drift, unit of measurement, latency, resolution and data rate. These parameters enhance the cooperation between different WBAN measuring the same process. For example, if one is monitoring the temperature, regardless the sensor's physical characteristics, the data rate should be around 120 bps with a precision of 8 bits. In addition, these parameters may help in the management of the access to the medium. If a process is characterized by a small latency, it should be given a higher priority to access the medium and send the data to the Hub. Moreover, the process shall have one or many data. For example, the tracking process shall include latitude, longitude and speed data. Many recent researches have been investigated to describe the electronic health data. As one can see in the survey clause A.1.2 of the present document, some models are dedicated for services discovery like GATT profiles, others focused on the interoperability between different medical centres and clinical centres. The present document is concentrated on the data sent within the BAN and sent form the Hub to an external medical control and monitoring server.

EDAM (EMBRACE Data and Methods) [i.8] is a data model of bioinformatics operations (tool, application, or workflow functions), types of data, topics (application domains), and data formats. The applications of EDAM are within organizing tools and data, finding suitable tools in catalogues, and integrating them into complex applications or workflows. EDAM includes 4 main sub-ontologies of concepts:

- ***Operation*:** A function or process performed by a tool; what is done, but not (typically) how or in what context.
- ***Data*:** A type of data in common use in bioinformatics.
- ***Topic*:** A general bioinformatics subject or category, such as a field of study, data, processing, analysis or technology.
- ***Format*:** A specific layout for encoding a specific type of data in a computer file or message.

EDAM may be used in the proposed SmartBAN data model, in particular in case where the biomedical sensors are used for bioinformatics data, for example in the analysis of mutations in cancer. In such application, the process shall have as data the data class in EDAM.

WBANs are widely used for remote patient monitoring and non-medical applications. That is why other existing models that describe the data types sensed by biomedical nodes have to be searched. As shown in the survey clause A.1.2.3 of the present document and in the reference model of the electronic healthcare, the data types include all the biomedical data formats. Whereas the process shall have one or many data, the data shall have one data format or type as described in the data value of the reference model. Moreover, three properties shall be added to the data class of the SmartBAN model: QoS, Data size and Data transmission. The data size reflects the precision and the resolution of the sensed data. In addition, WBANs are characterized by heterogeneous data that differs in their data rate, bandwidth, data types, accuracy, delay and others. An effective QoS mechanism is needed in order to deliver adequately these various information [i.1] and [i.2]. Beside QoS, data transmission should be unicast, multicast or broadcast. In general, it is considered as multicast when it is sent to group of nodes within a cluster and as broadcast when it is sent to the entire active nodes in WBAN. If the remaining lifetime of a node is crucial, it can ignore a broadcast or multicast data to save power.

In SmartBAN systems, data may be modelled as time series to represent measurements or observations that vary over time, such as sensor readings. This allows for an efficient and structured representation of temporal data, enabling analysis and real-time monitoring. Such time series data can be described using Data Catalog Vocabulary (DCAT) [i.52] data model proposed by W3C. dcat:Dataset captures metadata about the data collection as a whole. Specifically, the dataset are modelled as a dcat:DatasetSeries, which is particularly suited for representing a collection of distributions organized by temporal intervals. Each distribution can describe the data available for a specific time range and its corresponding serialization format, such as CSV or JSON. Temporal coverage can be defined using properties like dct:temporal, linking the series to the time period it spans, while dcat:Distribution provides access mechanisms, including file download URLs URL (Unique Resource Locator) or APIs (Application Programming Interface), ensuring interoperability and accessibility for SmartBAN applications.

In addition, the data shall have constraints. In SensorML, data can be constrained by security constraints based on the Security Banner Marking model of the Intelligence Community Information Security Marking (IC-ISM) Standard, valid Time that describes the validity time for certain process, and legal constraints which can be a private Act, intellectual Property, rights or copyrights. Those constraints shall be reflected in the SmartBAN data model. Moreover, in the GATT service profiles, each data is described by its minimum value and maximum value and those values shall be part of the Validity class. Furthermore, a data may have operational constraints. Consequently, the constraints shall at least be legal constraint, validity constraints (having minimum and maximum values), operational constraints, security constraint (security protocol and used key or certificate's reference), and compression constraint (compression algorithm).

Finally, a light class to store the measurements done by the sensors shall be added to the SmartBAN open data model. One of the main goals of this class is to be implemented within the sensor without affecting its performance permitting data aggregation and data value verification within the node before sending the data to the Hub. It shall include the sensed value, the timestamp and may include Time To Live (TTL) in order to assure the freshness of the medical data. In that way, if the connection to the hub is temporary unavailable, the sensor can check for the freshness of the sensed data, and if it is expired the data will not be sent to the Hub. Due to the limited resources of the sensors in terms of processor and memory, only this part of the model should be processed by the sensors allowing the conversion from raw data to aggregated data when it is necessary. For example, if only the average heart rate of a patient is needed to calculate the number of burned calories, this calculation may be done by the sensor and sent to the hub without degrading the performance of the node.

6.2 SmartBAN core ontologies

6.2.0 Introduction

One of the most important problems in electronic health is data heterogeneity management and interoperability between different systems. Two types of interoperability can be differentiated:

1) Syntactic and structural interoperability: the structure and provenance of information is understood by any clinical system.

2) Semantic Interoperability: having a common understanding of the information and data sources semantic such that they could be interpreted in a unique way by different clinical systems.

In order to ensure this SmartBAN interoperability and heterogeneity management, a dedicated open data model was already introduced and conceptualized in clause 6.1 of the present document (in the form of UML class diagrams). This open data model concept shall now be fully formalized and implemented through the use of ontology and this is the objective of clause 6.2 of the present document.

Therefore, a SmartBAN-dedicated ontology shall be formalized and developed based on the meta-model defined in clause 6.1. This BAN ontology, fully specified and presented in clause 6.2 of the present document, shall be designed in a way that existing healthcare and telemedicine data/information models and standards can be utilized through it (i.e. this SmartBAN ontology) for developing smarter control and monitoring operations as well as standardized eHealth services.

Ontology Web Language (OWL) has first been retained as description language for the formalization and the implementation of the computer interpretable SmartBAN-dedicated semantic ontology. This OWL-based ontology will allow the unique interpreting of different data, information and terms coming from different biomedical devices, applications and vendors. OWL DL, the lighter version of OWL, offers decidability by allowing consistency checking and automatic reasoning on the knowledge base using realistic computing resources in a reasonable time. Advanced rules may be assigned to the ontology to infer particular information based on the domain's requirements. Basic ontological components are classes, which are the sets of individuals with common characteristics, individuals which are specified objects in the domain and properties used to state relationships between individuals or from individuals to data values.

OWL DL offers the following options when defining a new class:

- subclass of;
- equivalent to;
- disjoint with;
- min cardinality;
- max cardinality.

Moreover, when creating an object property (which is the relation between two classes), the domain and the range shall be indicated and one can precise if it is:

- functional or inverse functional;
- transitive;
- inverse of;
- symmetric or asymmetric;
- reflexive or irreflexive.

Regarding the data property, the domain and the range (integer, string, anyURI, etc.) shall be precised. The data property may be functional, equivalent or disjoint from another data property.

The version of Protégé (4.3) has been used in the present document to formalize the SmartBAN semantic open data model and ontology for the following reasons:

- Mostly used.
- Support RDF/OWL structure.
- User friendly interface.
- Can import existing ontologies.
- Clear visualization of the classes and properties (OntoGraph plug-in).
- Support wide range of reasoners especially Pellet reasoner.
- Support SWRL language.
- Clear justification of the inferences.

OWL DL can be serialized in several formats, including RDF/XML, Turtle, JSON-LD (JavaScript Object Notation for Linked Data), and Manchester Syntax; The SmartBAN ontology is formalized in TTL due to its human readability, compactness, and wide support in RDF tooling, making it ideal for developing and reviewing ontologies.

Benefiting from the properties in OWL DL, the SmartBAN dedicated ontology shall be created as follows. Firstly, the SmartBAN domain shall be split and represented with three different sub-ontologies: WBAN, Nodes and Process. Those sub-ontologies shall fully mirror the three main classes of the SmartBAN open data model concept (see clause 6.1 of the present document). The distribution and description of the classes shall also mirror the one that has already been described in the UML class diagrams presented in clause 6.1 of the present document. Furthermore, when defining the properties of each class, one noted that many properties are variables described by their value and unit of measurement (like e.g. frequency, weight, height, speed, etc.). This is why a new class Quantity, capable of generically defining any kinds of units, shall be introduce into the SmartBAN data model.

QUDT was developed by NASA for that purpose and its last specification and version were released in April 2013 [3]. The last release was published in 12 December 2024. The already proposed models for managing hardware heterogeneity in wireless sensor network (see survey clause A.1.1.6 of the present document) also adopted QUDT ontology for units definition due to its wide adoption in many domains like e.g. e-commerce platform. Therefore, the Unit class from QUDT (Quantities, Units, Dimensions and Data Types) generic model [3] shall be adopted for the Quantity class. Figure 5 depicts a part of the hierarchy of the unit class. This hierarchy is also extensible for new units' type definitions.

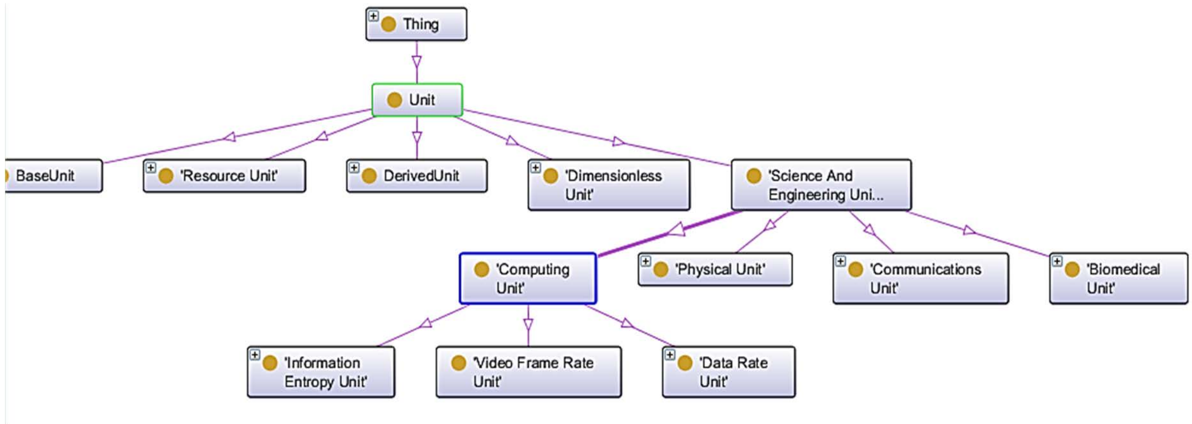


Figure 5: Part of the hierarchy of the QUDT unit class

The Quantity class of QUDT represents a specific measurement. It should be used to represent physiological property, expressed as a value combined with a unit (e.g. "98,6 °F" for body temperature or "75 bpm" for heart rate). A quantity kind refers to the abstract category of the physiological property being measured, such as body temperature, heart rate, or blood pressure, defining the type of measurement without specifying a value or unit. The quantity unit is the standardized reference used to express the magnitude of a quantity, ensuring consistency in measurements (e.g. "beats per minute" (bpm) for heart rate, "degrees Fahrenheit" (°F) for temperature). These concepts are essential for accurately representing and interpreting physiological data in WBANs.

In our Model, the Quantity class shall have as data property "hasvalue" of any type and has, as quantity, an individual of type Quantity. The Quantity has a quantity kind the general kind of the quantity, for example, temperature, length, height, etc. The quantity kind can be an qudt:quantitykind class defined in the QUDT model [3].

The node's location is essential in WBAN where the sensor's position should be: implanted or inside body, body surface or external. The node's location shall then be added to the SmartBAN model and ontology. In addition, the WBAN's location shall also be added to the model. Since the location is shared by more than one concept (person, organization, node, wban, etc.), an ontology "Location" shall be introduced in the SmartBAN model and ontology for that purpose. This ontology shall have as subclasses:

- Implant: which can be under the skin or inside the human body.
- Body surface: which has as data properties the position of the node relatively to the human bod.
- External: this subclass should be used to express the location of the WBAN's Hub, or nodes built in the PDA or Mobile of the patient, or to describe the location of the WBAN. This location may be given as coordinates for example using the latitude and longitude, or given by an address (City, country, building, etc.) or others.

The use of Location as independent ontology gives the user the ability to extend it by importing existing ontologies dedicated for locations. Some of these ontologies are e.g.

- Location.owl [i.9], an ontology to describe geographical and non geographical locations.
- The Places ontology [i.10], an ontology created by Smethurst, Styles and Scott for describing places of geographic interest.
- The W3C Basic Geo Vocabulary provides [i.53] a simple RDF namespace for representing spatially-located entities using properties such as latitude (geo:lat) and longitude (geo:long), based on the WGS84 coordinate reference system. At its core is the class geo:SpatialThing, used to describe anything with a geographic location, physical places, events, or conceptual entities.

Figure 6 shows the hierarchy of the Location class that shall be adopted for SmartBAN.

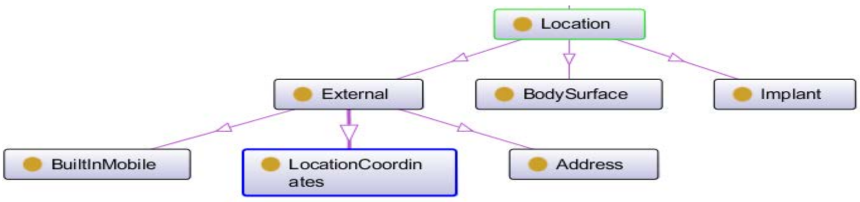


Figure 6: Classes hierarchy for Location ontology

The following clauses will now detail proposed SmartBAN ontology that shall be adopted, with all the classes introduced, their object's properties and their data properties.

To ensure that an ontology is FAIR (Findable, Accessible, Interoperable, and Reusable) compliant, it shall include rich and standardized annotation properties. These annotations provide critical metadata that enhance the ontology's usability and long-term value [i.54]. These annotation properties shall be reused from existing ontologies such as ***dcterms*** that refers to the Dublin Core Terms vocabulary [i.55], a widely used set of metadata properties for describing resources such as title, creator, date, and rights and ***vann*** that stands for the Vocabulary Annotation ontology [i.56], which provides properties for documenting and annotating vocabularies, including preferred namespace prefixes and namespace URIs. Consequently, all ontologies shall include:

- a resolvable and persistent identifier (owl:ontologyIRI, vann:preferredNamespacePrefix, vann:preferredNamespaceUri) which allows others to reliably locate and reference the ontology;
- descriptive properties (dcterms:description and rdfs:label) that help users and machines understand its purpose and scope;
- information about authorship and attribution (dcterms:creator, dcterms:publisher, dcterms:license) that emhances the reuse of the ontology;

- metadata related to maintenance and versioning (e.g. owl:versionInfo, owl:priorVersion, dcterms:created, dcterms:issued) that enable tracking of the ontology's evolution and provenance.

Together, these annotation properties make the ontology more discoverable, interpretable, and reusable by both humans and machine. Moreover, each ontology's concept shall have:

- rdfs:label that provides a concise, human-readable name for the term.
- rdfs:comment that offers a longer description or explanation about the term's meaning and use. It is advised to use well known medical standard to describe the used terms.

6.2.1 WBAN ontology

Classes:

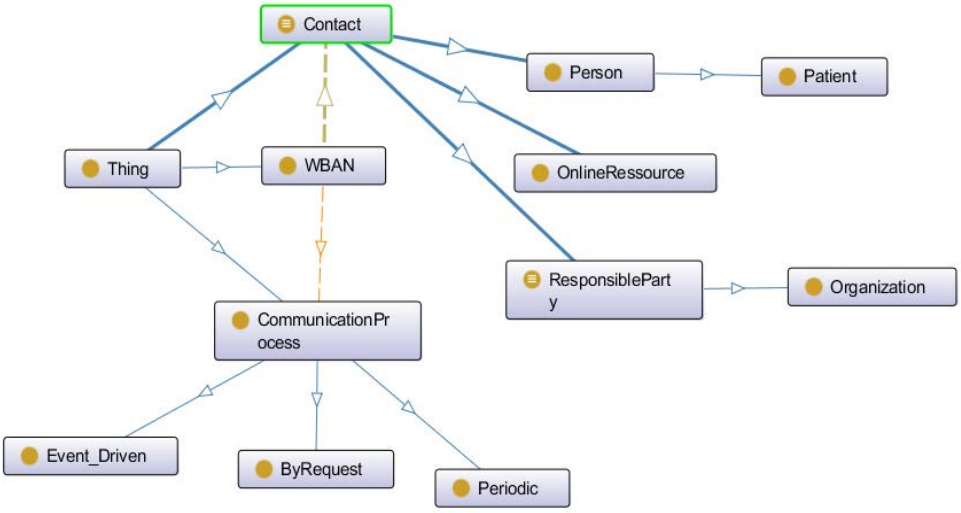


Figure 7: WBAN/SmartBAN Classes hierarchy

Object Properties:

Object property	Domain	Range
hasContact	WBAN	Contact
hasCommunicationProcess	WBAN	CommunicationProcess
hasWbanLocation	WBAN	Location
isAssociatedTo	Role	WBAN
playsRole	Contact	Role
uses	Patient	WBAN
isUsedBy	WBAN	Patient
hasHeight	Patient	Quantity (see note):Quantity
hasWeight	Patient	quantity:Quantity
hasHipCircumference	Patient	quantity:Quantity
hasRestingHeartRate	Patient	quantity:Quantity
hasMaxHeartRate	Patient	quantity:Quantity
hasFatBurnHeartRate	Patient	quantity:Quantity
hasWaistCircumference	Patient	quantity:Quantity
NOTE: Quantity: refers to the Quantity ontology of the SmartBAN Data Model.		

Data Properties:

Data Property	Domain	Range	Additional property
ApplicationDomain	WBAN	string	
BloodType	Patient	{"A+","A-","B+","B-","AB+","AB-","O+","O-"}	
Density	WBAN	int	
DeploymentTime	WBAN	dateTime	
FaultTolerence	WBAN	real	
Language	Person	string	
LifeTime	WBAN	duration	
Reference	OnlineResourse	anyURI	
Phenomena	WBAN	string	
RadioTechnology	WBAN	string	
Topology	WBAN	{"Adhoc","Star","Mesh", "P2P","Others"}	
UserID	Patient		
WBANID	WBAN	anyURI	Functional

6.2.2 Nodes ontology

Classes:

The nodes subclasses are depicted in Figure 8 and the nodes classes relationships are depicted in Figure 9.

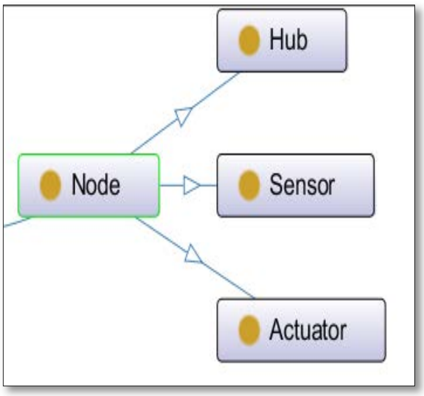


Figure 8: Node's subclasses

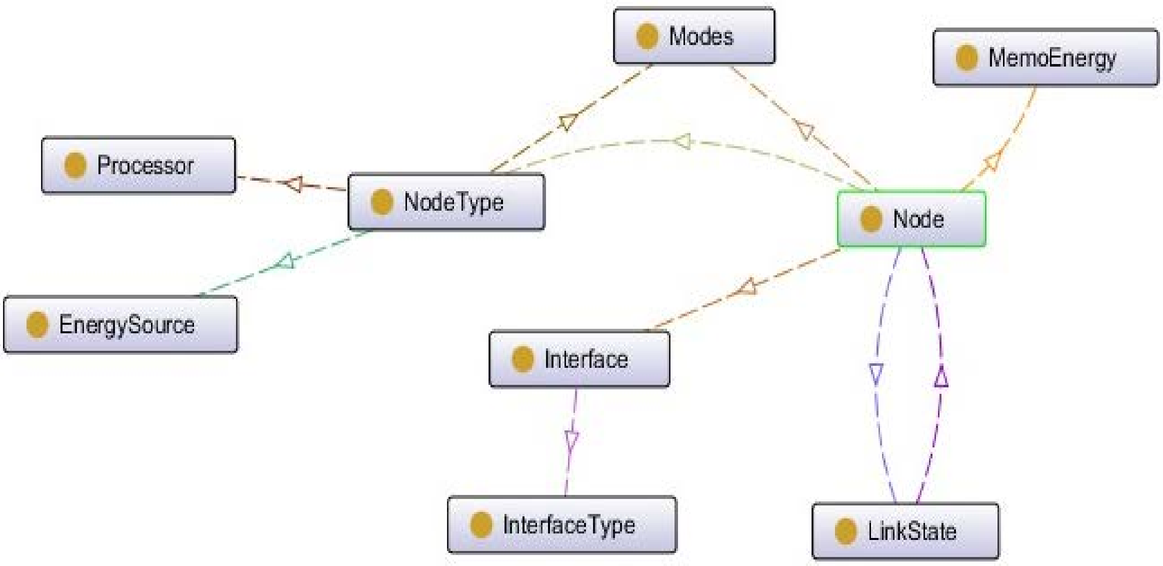


Figure 9: Nodes Classes relationships

Object Properties:

Object property	Domain	Range	Cardinality	
			min	max
cost	NodeType	quantity:Quantity		1
dimension	NodeType	quantity:Quantity		4
dutyCycle	Processor	quantity:Quantity		1
frequency	Processor	quantity:Quantity		1
baud	InterfaceType	quantity:Quantity		
hasAvailableFlash	MemoEnergy	quantity:Quantity		1
hasAvailableRAM	MemoEnergy	quantity:Quantity		1
hasEnergySrc	NodeType	EnergySource		
hasInterface	NodeType	Interface		
hasInterfaceType	Interface or NodeType	InterfaceType	1	1
hasLinkState	Node	LinkState		
hasLinkToNode	LinkState	Node		1
hasMAXFlash	Processor	quantity:Quantity		1
hasMAXRAM	Processor	quantity:Quantity		1
hasMemoEnergy	Node	MemoEnergy		1
hasModes	NodeType	Mode		
hasNodeType	Node	NodeType		1
hasProcessor	NodeType	Processor		
hasRemainBattery	MemoEnergy	quantity:Quantity		1
inCoverage	NodeType	quantity:Quantity		
inCurrentMode	Node	Mode	1	1
ModePower	Mode	quantity:Quantity		
outCoverage	NodeType	quantity:Quantity		
position	Node	Loc (see note):Location		
receivingSensitivity	NodeType	quantity:Quantity		1
velocity	NodeType	quantity:Quantity		1
wakeupLatency	NodeType	quantity:Quantity		1
NOTE: loc: refers to the Location ontology of the SmartBAN Data Model.				

Data Properties:

Data Property	Domain	Range	Additional property
interfaceAddress	Interface	string	
interfaceType	InterfaceType	string	
addressType		string	SubProperty of interfaceAddress
addressValue			SubProperty of interfaceAddress
delay	LinkState	duration	
nodeDiscoveryTime	NodeType	duration	
MIPS	Processor	int	
errorRate	LinkState	float	
firmwareVersion	NodeType	string	
firmware	NodeType	string	
interfaceDescription	InterfaceType	string	
interfaceID	Interface	anyURI	Functional
interfaceLayer	InterfaceType	string	
interfaceName	InterfaceType	string	
interfaceProtocol	InterfaceType	string	
isGateway	Interface	boolean	
modeID	Mode	anyURI	Functional
modeName	Mode	string	
nodeID	Node	anyURI	Functional
nodeTypeDesciption	NodeType	string	
nodeTypeName	NodeType	string	
operatingConstraints	NodeType	string	
processorID	Processor	anyURI	Functional
rechargeable	EnergySource	boolean	
serailNb	Node	string	
sourceType	EnergySource	string	
trustLevel	Node	int	

6.2.3 Process ontology

Classes:

The process classes and relations are depicted in Figure 10.

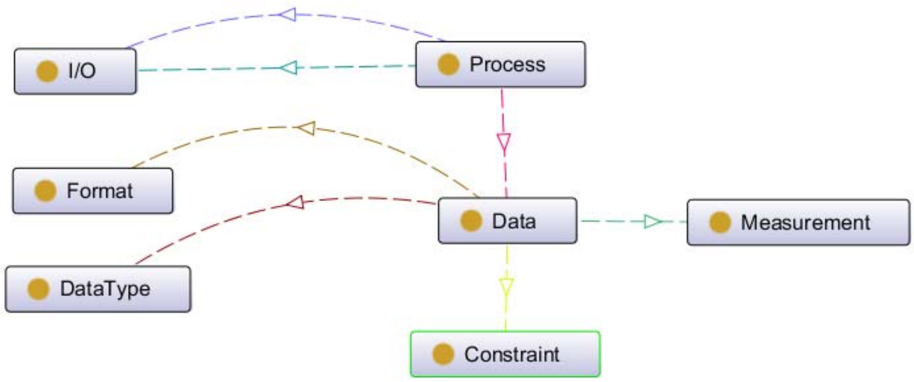


Figure 10: Process classes and relations

Object Properties:

Object property	Domain	Range	Cardinality	
			min	max
calibration	Process	quantity:Quantity		1
acts	node:Actuator	Action		
drift	Process	quantity:Quantity		
measures	node:Sensor	Measurement		
isUsedFor	node:Node	Process		
hasConstraint	Data	Constraint		
hasData	Process	Data		
hasDataType	Data	DataType		1
hasDerivedData	Process	Data		
hasInput	Process	Process		
hasMeasurement	Data	Measurement		
hasOutput	Process	Process		
latency	Process	quantity:Quantity		1
precision	Process	quantity:Quantity		1
resolution	Process	quantity:Quantity		1
responseOrder	Process	quantity:Quantity		1
sampleRate	Process	quantity:Quantity		1
uom	Process	quantity:Unit		1

Data Properties:

Data Property	Domain	Range	Additional property
AlgorithmVersion	Compression or Security	string	
CompAlgorithm	Compression	string	
Key	Security	string	
KeyLength	Security	int	
MaxValue	Validity		
MinValue	Validity		
ProcessID	Process	anyURI	Functional
QOS	Data	int	
SecAlgorithm	Security	string	
TimeStamp	Measurement	dateTimeStamp	
TTL	Measurement	int	
valid	Measurement	boolean	
value	quantity:Quantity		

The valid data property shall be used to determine if the measurement is valid or invalid. For example:

- If the TTL is zero, which means that the measurement is not fresh.
- When the value of a measurement exceeds the maximum range.
- When the measurement is captured before the wakeup latency.
- Etc.

In all these cases, the measurement should be invalid and not considered as correct value for further treatments.

6.2.4 Ontofull ontology

Ontofull ontology (see clause B.4 of the present document for the full OWL-DL specification in turtle format) shall include six direct imported ontologies: wban.owl (see annex B of the present document for the full OWL-DL specification in turtle format), nodes.ttl (see clause B.2 of the present document for the full OWL-DL specification in turtle format), process.ttl (see clause B.3 of the present document for the full OWL-DL specification in turtle format) and quantity.ttl (see clause B.4 which contains the quantity classes), and location.ttl (see clause B.5 of the present document for the full OWL-DL specification in turtle format).

Object Properties:

Object property	Domain	Range	Cardinality		Additional property
			min	max	
BelongTo	Node	WBAN		1	inverse of Contains
Contains	WBAN	Node			inverse of BelongTo
DoAction	Actuator	Action			
ElectHub	WBAN	Hub		1	
isMeasuredBy	Measurement	Sensor		1	inverse of measures
Measures	Sensor	Measurement	1		inverse of isMeasuredBy
usedFor	Node	Process	1		

Rules:

Additional rules shall be added to the ontology based on the application domains of WBANs/SmartBANs. These rules are defined using Semantic Web Rule language SWRL [i.11] in the present document.

- Rule 1: When a node measures some measurements having certain Data belonging to a process A, one should assume that this node is used for process A. the rule is:
    - Node(?N), measures(?N, ?M), hasData(?P, ?D), hasMesurement(?D, ?M) -> usedFor(?N, ?P).
- Rule 2: When a WBAN/Smart BAN elects a Hub, one shall infer that this WBAN/Smart BAN contains the Hub:
    - WBAN(?W), ElectHub(?W, ?S) -> Contains(?W, ?S).
- Rule 3: If the WBAN/Smart BAN contains a sensor built in the mobile, and the WBAN/Smart BAN elects this mobile as Hub, one should assert that the sensor has the same node type as the mobile:
    - WBAN(?W), Position(?N, BuiltInMobile), hasNodeType(?S, ?NT), Contains(?W, ?N), ElectHub(?W, ?S) -> hasNodeType(?N, ?NT).
- Rule 4: If the TTL of a measurement is equal to zero, the measurement should be invalid:
    - Measurement(?M), TTL(?M, ?T), equal(?T, 0) -> valid(?M, false).
- Rule 5: In the context of SmartBAN, the hub shall observe the link state of the nodes belonging to the same WBAN/Smart BAN of a certain patient. Thus, if a Hub has link state to a node, and this Hub belongs to a WBAN/Smart BAN, the node should belong to the same WBAN/Smart BAN:
    - WBAN(?W), hasLinkState(?S, ?L), hasLinkToNode(?L, ?N), Contains(?W, ?S) -> Contains(?W, ?N).

- WBAN(?W), hasLinkState(?S, ?L), hasLinkToNode(?L, ?N), Contains(?W, ?N) -> Contains(?W, ?S).

The users of the SmartBAN ontology may define additional rules related to its application using the SWRL language. The general rules applicable for all types of WBAN/Smart BAN applications have only been defined/specified in the present document.

6.2.5 OWL-DL formalization of SmartBAN core ontologies

This clause presents the OWL-DL formalization that shall have TC SmartBAN core ontologies. Since all the ontology classes, objects and attributes were already fully detailed in the previous clauses of the present document, their corresponding text XML format will only be listed in Annex B of the present document.

6.3 Semantic interoperability and upper level model for SmartBAN

The objective of clause 6.3 of the present document is mainly to provide a short analysis of SmartBAN semantic interoperability management strategies. The corresponding relevant upper level concepts, fulfilling the SmartBAN system level requirements (see ETSI TR 103 394 [i.57]) and that have to be carried out for providing SmartBAN semantic interoperability handling, will also be listed below.

In the context of SmartBAN, interoperability management is handled via an ontology-based strategy and is fully relying on the SmartBAN reference model and core ontologies already described in clause 6 of the present document. Semantic interoperability thus consists on modelling semantically working interactions between entities (e.g. medical devices) and systems that have to interoperate, at all levels and in particular at data, device, network, system and application levels. Based on the analysis of existing interoperability models already investigated in clauses A.1.3 and A.1.4 of the present document, the SmartBAN reference model and associated modular ontologies shall be extended with service level concepts, service modular ontology and associated enablers. Semantic Web and WoT(to some extend) strategies are in particular retained (see clauses A.1.3 and A.1.4 of the present document) and the corresponding service model is derived enabling the SmartBAN Reference Model extension with an upper ontology at service level. This allows, in particular, full semantic interoperability management (application level included), medical device discovery, data sharing (application level), cross domain use cases handling (e.g. patient on the move, healthy lifestyles for citizens, emergency situation detection and warning, elderly at home, etc.). Indeed, the use of sensor as a service (whatever a medical, an automotive, a home- or a city-related one) that is described/represented through a common/generic reference model, can be viewed as a generic service. The SmartBAN service model extension is fully described, specified and formalized within clause 6.4 of the present document.

6.4 SmartBAN open service model and ontology

6.4.0 Introduction

One of the most important problems in electronic health is data interoperability between different systems. Two types of interoperability can be differentiated:

1) Syntactic and structural interoperability: the structure and provenance of information is understood by any clinical system.

2) Semantic Interoperability: having a common understanding of the information and data sources semantic such that they could be interpreted in a unique way by different clinical systems.

A already mentioned in clause 6.4 of the present document and for ensuring SmartBAN full interoperability management (semantic and cross-domain included), a SmartBAN dedicated open service model shall be introduced for extending the SmartBAN reference model and its core ontologies with an upper ontology. This open service model concept shall now be fully specified and formalized, through the use of ontology and this is the objective of clause 6.5 of the present document.

Based on the existing semantic solutions for services 'description (see clause A.1 of the present document), a SmartBAN Service Ontology is specified and formalized by re-using the general structure of OWL-S ontology (see clause A.1.3 of the present document). The SmartBAN Service Ontology is thus composed of three main classes (i.e. concepts):

- SmartBAN ServiceProfile: It describes the service and enables the auto-discovery of services.
- SmartBAN ServiceProcess: It describes how the service is realized based on the IOPE (Input, Output, Precondition and Effect) mechanism.
- SmartBAN ServiceGrounding: It describes how the service can be invoked.

In addition, a general SmartBAN Service model that combines all the aforementioned Service Ontology classes (i.e. concepts) and includes general and domain rules is also introduced and defined in clause 6.4.4of the present document.

The SmartBAN Service Ontology is also designed using the modularization principle that concerns the structuring of a wide concept into multiple, simpler and self-designed sub-concepts that can be specified and handled independently. In fact, the reason behind using modular ontology is not only coming from OWL-S but also from the necessity for implementing one part of the ontology in constraint nodes. The Smart Service ontology is thus divided into sub-ontologies (modules) that can be separately treated and reasoned in specific nodes (embedded analytics and edge computing included). Moreover, many applications require customization and personalization, which will be easier using modular ontologies. The remaining clauses of clause 6.4 of the present document are detailing all the aforementioned SmartBAN Service model sub-ontologies, i.e. ServiceProfile, ServiceProcess, ServiceGrounding and Service ontologies.

6.4.1 SmartBAN ServiceProfile ontology

6.4.1.0 Introduction

The integration SmartBAN in WoT imposes that the BAN can be discovered by end users/software. At that end, services offered by SmartBAN should be described in a semantic manner. This is the aim of SmartBAN 'ServiceProfile' ontology. It fully describes the non-functional characteristics of a service. This information is used to identify the best service that suits the need of the service consumer. Thus, it will help to resolve the problem of bugs or corruption introduced by some services and automatically switch to another service having same functionality.

6.4.1.1 Service Profile Model

First of all, a service profile is described by its name and description (reproduced form OWL-S). Because a WBAN can cover a wide range of application domain having different requirements, constraints and characteristics, the domain experts were left the opportunity to classify the services based on their own criteria. This is modeled by the relation "hasServiceClassification" between the "ServiceProfile" class and the domain ontology.

Because the SmartBAN service profile model in dedicated for BAN, each service are categorized based on a service category that reflects the observed feature. The service category is divided into three sub-categories:

- Body: divided into general and medical. This category groups all services related to human body. It includes general services (acceleration, speed, position, status, location, fall detection, etc.), and medical services (temperature, heart rate, blood pressure, calories burned, etc.).
- Ambient: includes weather services like ambient temperature, pressure, humidity, light intensity, wind speed, etc.
- Object: services related to objects like the velocity, acceleration, temperature, etc.

This categorization is mainly used when a client requests a service. In addition, this semantic description will eliminate the imprecise meaning of certain parameters like body temperature and human temperature. It is modeled by the relation "hasServiceCategory". In a BAN, services are basically:

- data collected from sensors;
- commands that should be executed by actuators;

- alerts that inform the user about urgent cases;
- notifications that notify the user about a change or an update, as well as remind the user about an event that he/she assigned.

Knowing the type of service permits the users (e.g. caregivers or patients) to choose which service they want to use.

EXAMPLE: If a user is interested in monitoring his blood pressure, he/she can choose a monitoring service where:

- the data are periodically collected and displayed; or
- a notification is sent when his/her blood pressure is changing; or
- an alert is sent when his/her blood pressure is abnormal.

Thus, the "ServiceCategory" class is used to differ from two services related to the same observed data. In addition, this class is very effective for service provider/developer because it allows them to use the adequate service category based on their users' needs. If the client is requesting a fall detection service that sends him/her a notification, when the service discovery agent is searching for available services, the system will search for a service which has as service category "fall detection" and as service type "notification".

For dealing with various types of data (periodic, urgent, time sensitive, etc.), the following concepts are also integrated into the "ServiceProfile" model:

- Quality of Service (QoS), indicating the quality of the provided service; and
- Quality of Information (QoI), indicating the quality of the information delivered by the service.

QoI parameters are calculated from the QoS parameters using certain formulas that are also saved in the SmartBAN modular service ontology. QoS and QoI parameters permit to choose the adequate service between different services having same category and type. For example, if two thermometer sensors are gathering the body temperature, but the accuracy of one differs from the other, the QoI will permit to select the more accurate thermometer to retrieve the body temperature.

Furthermore, the service constraints (security, legal, time validity, etc.) were also added to the SmartBAN "ServiceProfile" model. These constraints are modeled by the "Constraints" class of SmartBAN "Process" sub-ontology (see clause 6.2.3 of the present document).

Finally, like for OWL-S, the service profile classes (i.e. concept) is associated to a service process. In OWL-S, a service profile has contact information. Or, in the SmartBAN "WBAN" sub-ontology, each BAN has a contact. Therefore, the contact information is not added to the SmartBAN service profile ontology, but the WBAN ontology is rather linked to the service ontology by the following relation: WBAN "isEngaged" in Service. Accordingly, the contact information can be deduced based on a SWRL rule that is explained in clause 6.4.4 of the present document. Figure 11 depicts the class diagram of "MyOntoServiceProfile" ontology.

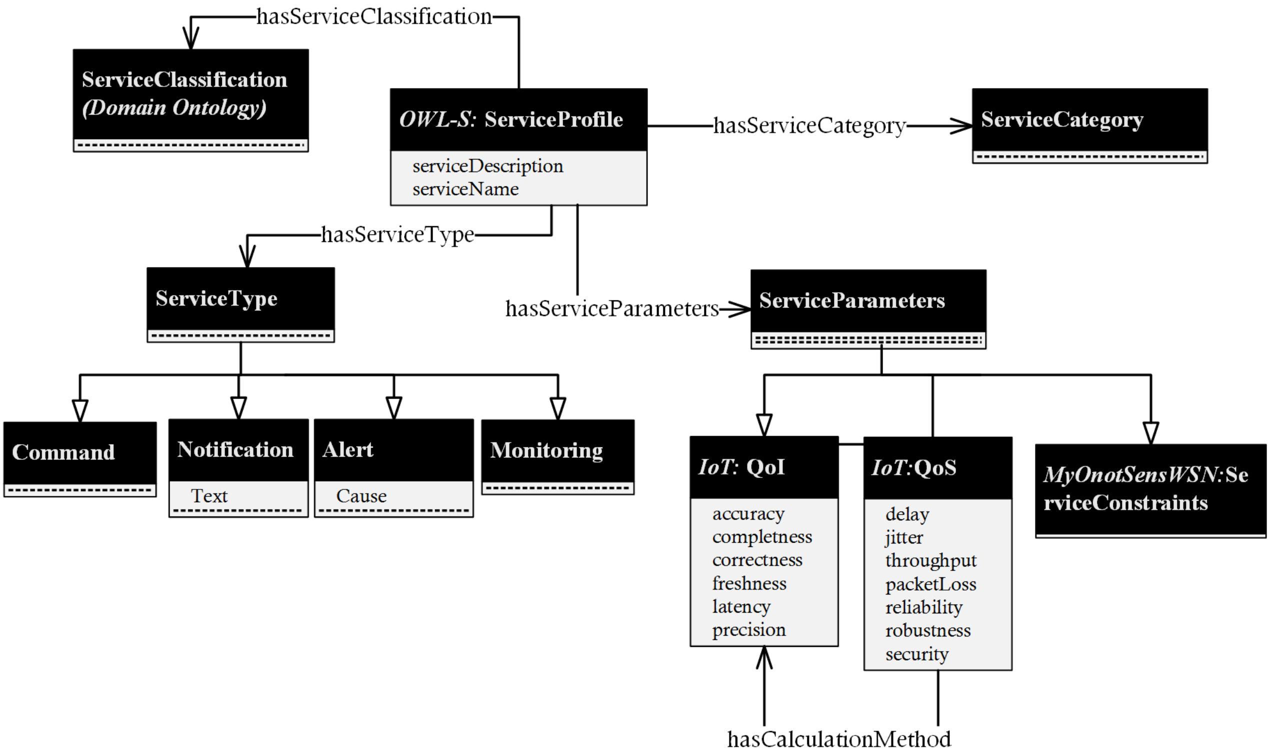


Figure 11: ServiceProfile Class Diagram of the SmartBAN Service model

6.4.1.2 Service Profile Ontology Description

Clause 6.4.1.2 of the present document details the object properties and data properties of SmartBAN "ServiceProfile" ontology. The full description of the ontology is given in clause B.5.

Object Properties:

Object property	Domain	Range	Additional description
hasCalculationMethod	QoS	QoI	
hasServiceCategory	ServiceProfile	ServiceCategory	
hasServiceClassification	ServiceProfile	Any imported ontology	
hasServiceParameter	ServiceProfile	ServiceParameter	
hasQoS	ServiceProfile	QoS	subProperty of *hasServiceParameter*
hasQoI	ServiceProfile	QoI	subProperty of *hasServiceParameter*
hasConstraints	ServiceProfile	ServiceComstraints	subProperty of *hasServiceParameter*
hasServiceProcess	ServiceProfile	MyOntoServiceProcess: ServiceProcess	
hasServiceType	ServiceProfile	ServiceType	

Data Properties:

Data Property	Domain	Range	Additional property
accuracy	QoI		
completeness	QoI		
correctness correctness	QoI		
freshness	QoI		
latency	QoI		
precision	QoI		
delay	QoS		
jitter	QoS		
throughput	QoS		
packetLoss	QoS		
reliabilty	QoS		
robustness	QoS		

Data Property	Domain	Range	Additional property
security	QoS		
serviceName	ServiceProfile	anyURI	Functional
serviceDescription	ServiceProfile	String	
Cause	Alert	String	
code	ServiceType		
CommandDescr	Command	String	
Text	Notification	String	

6.4.2 SmartBAN ServiceProcess ontology

6.4.2.0 Introduction

The aim goal of SmartBAN service process sub-ontology is to describe how the service that is executed, as well as how a user (e.g. a caregiver) can interact with the service. As for OWL-S, SmatBAN "ServiceProcess" ontology is based on the IOPE principle.

The same service profile can have many service process models. For example, if a fall detection service is provided, it can be based on the calculation of the linear acceleration or based on the angular and linear acceleration. In this case, two different process models are combined into the same service profile. Moreover, when a user requests a service, he should know what the required input parameters are, as well as if there is a pre-condition or not. This can be deduced from the SmartBAN service process model. Note that the user can be a software, an application, a service or a end user (patients, caregivers or helpers). The detail of the service process model is given in clause 6.4.2.1 of the present document.

6.4.2.1 Service Process Model

Within OWL-S, a process is defined as an atomic process, or as a composite process expanded into one or more simple processes. The same idea is transposed for SmartBAN "ServiceProcess" model and WBAN.

Basically in WBAN, a trivial service is a service providing the data collected by a sensor. Within the SmartBAN "Process" sub-ontology (see clause 6.2.3 of the present document), each sensor is used in a process. Thus, this process can be considered as the trivial service offered by the BAN (like temperature, acceleration, blood pressure, etc.). Consequently, each process is an atomic service process within the SmartBAN "ServiceProcess" model.

In order to enhance the semantic description of the composite processing, the "ProcessRelation" class (concept) is added to the SmartBAN "ServiceProcess" model. This class models the relation between two (or more) simple processes expanding from the same composite process. The relation could be an order (first, before, etc.), parallel, or a condition between two simple processes.

Re-using the "Method" class from OWL-S, the relation "hasCalculationMethod" between the service process class and the method class is also added to the SmartBAN "ServiceProcess" model. For that purpose, sub-classes are added to the class method where a method can be an aggregation (maximum, minimum, etc.), or a specific formula (least square, etc.).

As aforementioned, the SmartBAN service process has input parameters, output parameters, precondition(s), and result (s). The "Optional" property is also added to the parameter class as Boolean. Within the SmartBAN "ServiceProcess" model, each service process can have:

- Input parameters (same as OWL-S), e.g. the patient being monitored.
- Output parameters (same as OWL-S), e.g. the observed value.
- Pre-condition (same as OWL-S), e.g. the user permission to access the monitored data.
- Effect. For SmartBAN, the effect can e.g. be a display, notification, sound effect, command, action, etc.
- Post-condition, a condition to be satisfied when the process is executed.

EXAMPLE: If a patient's relative can monitor the patient's heart rate, then the service is the monitoring of the heart rate. In that case:

- the inputs are the time of the requested service and the requestor identity to verify the permission (in case of using a smartphone for monitoring, it can be e.g. his access code);
- the precondition is that the heart rate sensor is enabled in order to retrieve data;
- the effect is a display on the mobile's screen of the relative; and
- the post condition could be a confirmation from the patient that this relative can check his heart rate, although he already has the permission.

Likewise, OWL-S defined for each process participants using the object property "hasParticipant". This object property has two sub-properties: "hasClient" and "performedBy". For BAN, an application generally has end users (e.g. helpers or caregivers) or software as client and is performed by a node. But what is really essential for BAN is also to know the producer of the data. For that purpose, the concept of Entity is introduced for the SmartBAN "ServiceProcess" model. This entity can be a physical entity (person, node or object), or virtual entity (software, social person or application). Therefore, the object property "hasParticipant" is added between the service process class and the Entity class. This object property has two sub-object properties: "hasConsumer" and "hasProducer". While Figure 12 depicts the service process class diagram, Figure 13 depicts the sub-classes of the Entity class.

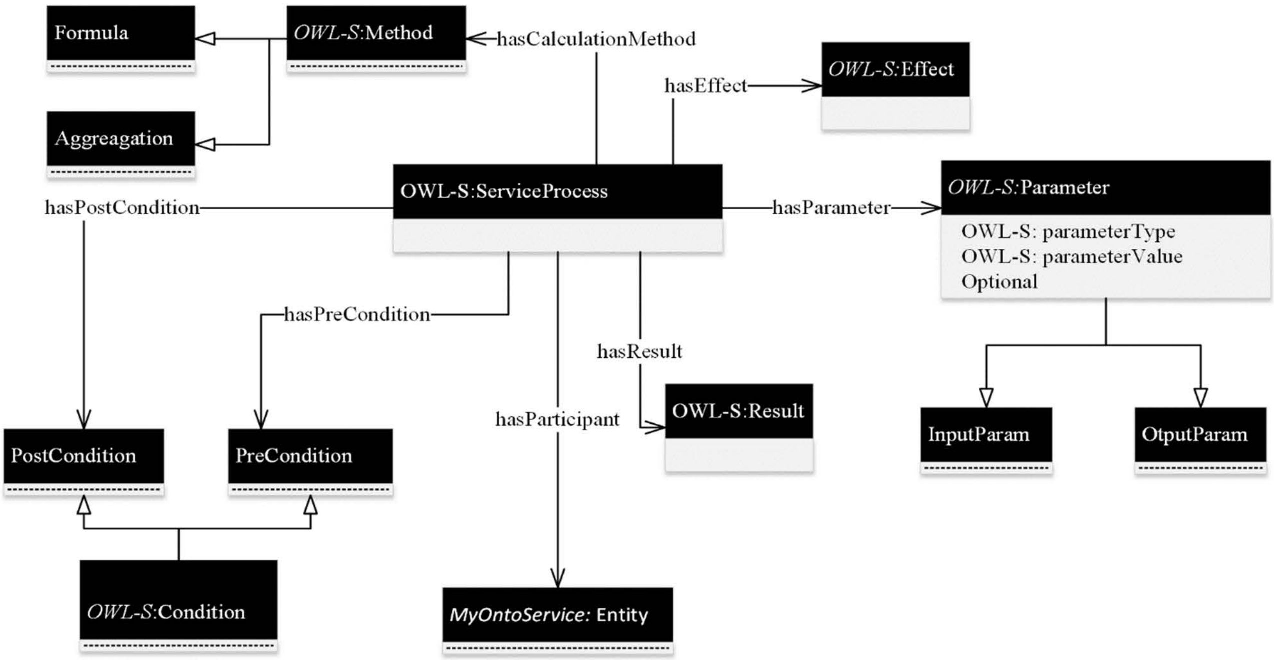


Figure 12: SmartBAN Service Process Class Diagram

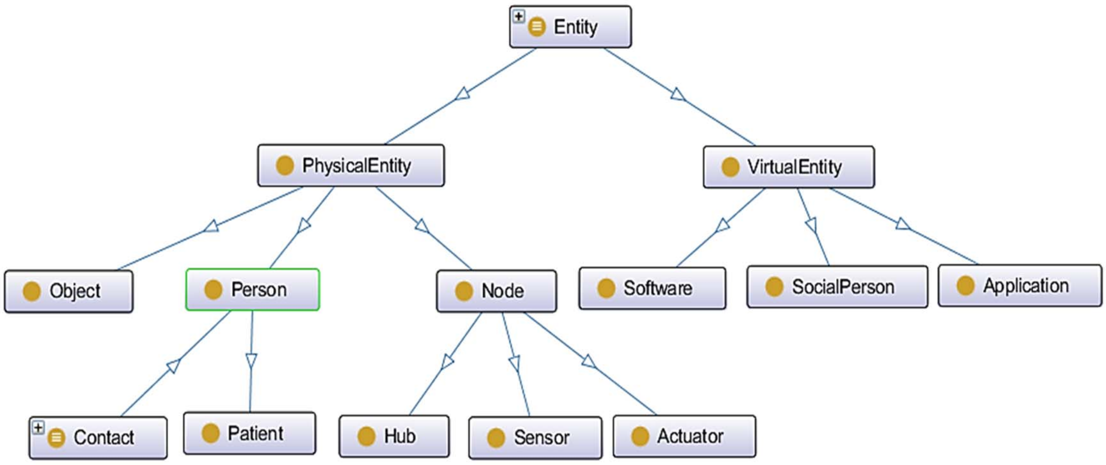


Figure 13: Entity sub-classes of the SmartBAN "ServiceProcess" model

6.4.2.2 Service Process Ontology Description

The "Atomic" class of the SmartBAN "ServiceProcess" ontology is equivalent to the "Process" class of SmartBAN "Process" sub-ontology (see clause 6.2.3 of the present document). In addition, the "ServiceProcess" class is also an abstract class that can be a composite, simple or atomic class.

Object Properties:

Object property	Domain	Range	Additional Description
hasPostCondition	QoS	QoI	
hasPreCondition			
Collapses	Simple	Composite	Inverse Of: expandsTo
expandsTo	Composite	Simple	Inverse Of: collapses
hasCalculationMethod	ServiceProcess	Method	
hasParameter	ServiceProcess	Parameter	
hasInputParam	ServiceProcess	InputParam	Sub-property Of: hasParameter
hasOutputParam	ServiceProcess	OutputParam	Sub-property Of: hasParameter
hasProcessRelation	Composite	ProcessRelation	
hasResult	ServiceProcess	Result	
hasEffect	ServiceProcess	Effect	
isAssociatedWith	ServiceProcess	(Domain Ontology)	
isRelatedTo	ProcessRelation	Simple	
isAfter			Sub-Propety Of: isRelatedTo
isBefore			Sub-Propety Of: isRelatedTo
isInParallel			Sub-Propety Of: isRelatedTo
hasParticipant	ServiceProcess	*MyOntoService:* Entity	
hasProducer			Sub-Property Of: hasParticipant
hasConsumer			Sub-Property Of: hasParticipant

Data Properties:

Data Property	Domain	Range
Optional	Parameter	boolean
parameterType	Parameter	
parameterValue	Parameter	
RelationType	ProcessRelation	string

6.4.3 SmartBAN ServiceGrounding ontology

6.4.3.0 Introduction

The SmartBAN service grounding ontology models how the service is invoked, as well as how to map semantic data to required data for transmission. This sub-ontology permits to find a service based on the technology enabled on the user (e.g. patient, caregiver, helper) side.

a) As defined in OWL-S, each SmartBAN service grounding has a determined port number, reference and protocol version. OWL-S restricted the grounding sub classes to the SOAP technique for services. It offered the possibility to add REST techniques. With the evolution of web services techniques new protocols appeared for data transmission like, e.g. MQTT or CoAP dedicated for constraint devices, JSON and JSON-LD for non-XML data transmission, HL7 messages for biomedical data, Bluetooth LE messages for Bluetooth devices, etc. For that reason:the sub-class hierarchy of the grounding class is extended as depicted in Figure 14; and

b) the service profile is linked to two classes: the "MessgeElements" and "MessageOptions" classes as depicted in Figure 15. These classes describe the optional/required fields with its characteristics in order to form a message based on one of the service grounding class.

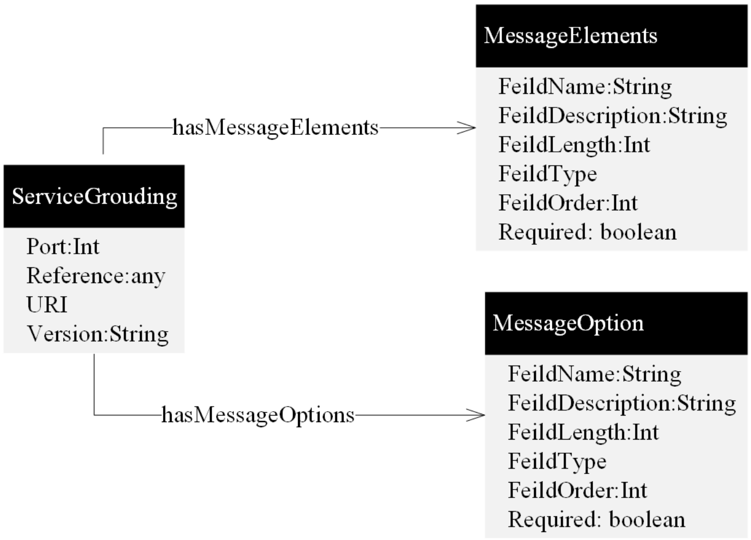


Figure 14: SmartBAN ServiceGrounding class diagram

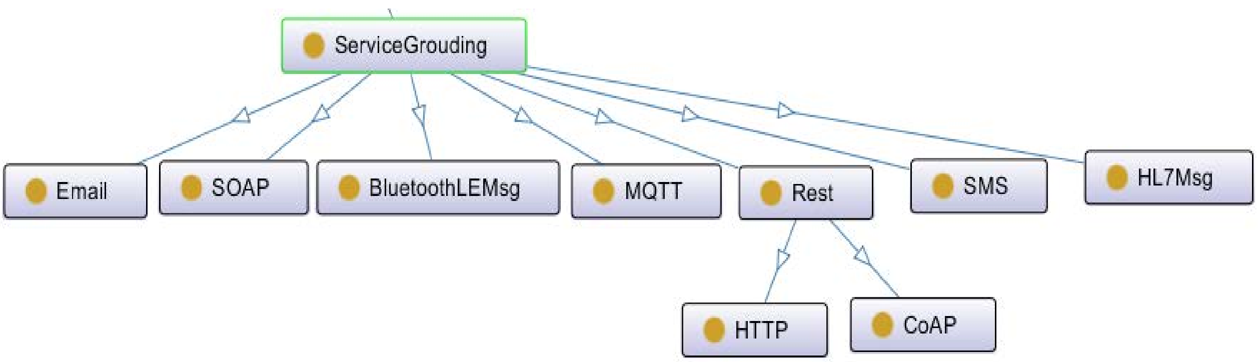


Figure 15: SmartBAN ServiceGrounding sub-classes

6.4.3.1 Service Grounding ontology description

Object Properties:

Object property	Domain	Range
hasMessageElements	ServiceGrounding	MessageElements
hasMessageOptions	ServiceGrounding	MessageOption

Data Properties:

Data Property	Domain	Range
FeildDescription	MessageElements or MessageOption	string
FeildLenght	MessageElements or MessageOption	integer
FeildName	MessageElements or MessageOption	string
FeildOrder	MessageElements or MessageOption	integer
FeildType	MessageElements or MessageOption	
Required	MessageElements or MessageOption	boolean
Port	ServiceGrounding	int
Reference	ServiceGrounding	anyURI
Version	ServiceGrounding	string

6.4.4 SmartBAN Service ontology

SmartBAN Service ontology is the global service ontology that directly imports the three aforementioned sub-ontologies. It contains the Service class described by the service's name and service description (same as in OWL-S, see clause A.1.3 of the present document). In addition, it also contains the object properties that link SmartBAN "ServiceProfile" (see clause 6.4.1 of the present document) , "ServiceProcess" (see clause 6.4.2 of the present document) and "ServiceGrounding" (see clause 6.4.3 of the present document) sub-ontologies. Figure 16 depicts the class diagram of the MyOntoService ontology.

Although a service may be available for the end users, the owner of a service (e.g. patients, caregivers, helpers) can choose to disable that service. That's why the object property "enable" is added between the Person class and the Service class, as depicted in Figure 16.

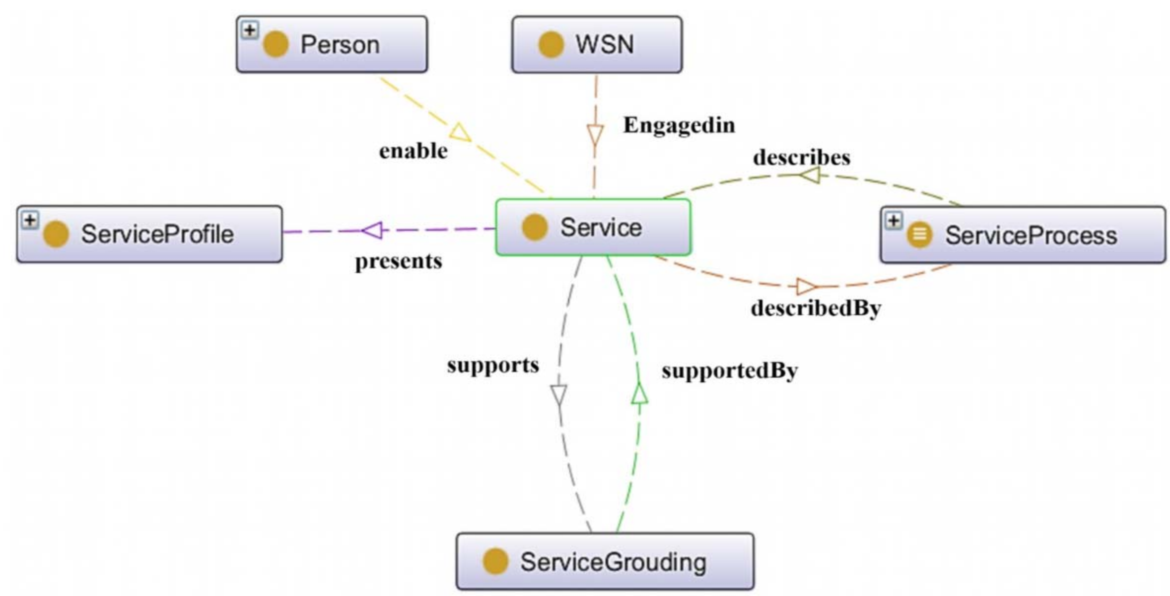


Figure 16: SmartBAN Service Classes schema

To give more expressiveness to the proposed SmartBAN Service ontology, SWRL rules are added. Firstly, the rules related to the trivial service offered by each BAN, which is the process modeling the observed value of a sensor, were added. The corresponding set of rules is the following:

1) Each process is considered as a service:

(Rule 3.1) Process(?P) -> Service(?P)

2) Each atomic process has as producer the sensor:

(Rule 3.2) Sensor(?S), Atomic(?a), usedFor(?S, ?a) -> hasProducer(?a, ?S)

3) The service ID of the process is the same as the process ID:

(Rule 3.3) Process(?P), ServiceProfile(?P), ProcessID(?P, ?PID) -> ServiceID(?P, ?PID)

4) If a process has constraints the service profile should have the same constraints:

(Rule 3.4-a) Process(?P), Security(?L), ServiceProfile(?P), hasConstraint(?D, ?L), hasData(?P, ?D) -> hasConstraint(?P, ?L)

(Rule 3.4-b) Legal(?L), Process(?P), ServiceProfile(?P), hasConstraint(?D, ?L), hasData(?P, ?D) -> hasConstraint(?P, ?L)

Secondly, when a sensor in a BAN is used in a process, that BAN is surely engaged in the corresponding service (the one described by that process). This leads to the following rule:

(Rule 3.5) ServiceProfile(?P), BelongToCluster(?S, ?C), usedFor(?S, ?P), belongTo(?C, ?W) -> Engagedin(?W, ?P)

Thirdly, if two simple processes are related by a sequence relation, if one simple server is before the other, assuredly the second one is after. This leads to the following rule:

(Rule 3.6) Simple(?P1), Simple(?P2), isAfter(?P1, ?P2) -> isBefore(?P2, ?P1)

Finally, (Rule 3.7) is added for the "isRelativeof" property:

(Rule 3.7) isRelativeof(?P1, ?P2) -> isRelativeof(?P2, ?P1)

More rules can also be added depending on the domain of application. A fully detailed example is given in clause 6.5 of the present document.

6.4.5 OWL-DL formalization of SmartBAN Service ontologies

This clause presents the OWL-DL formalization that shall have TC SmartBAN Service ontology. Since all the ontology classes, objects and attributes were already fully detailed in the previous clauses of the present document, their corresponding text XML format will only be listed in Annex B of the present document.

6.5 SmartBAN ontologies pre-validation

6.5.1 Introduction

The objective of clause 6.5 of the present document is to present the first pre-validation tests of the SmartBAN ontologies that have been conducted within TC SmartBAN. Mainly based on symbolic validation methods, those tests will demonstrate the a priori validity of the proposed SmartBAN ontology.

To pre-validate the ontology, the example of a WBAN The example of a WBAN for monitoring and assisting runners is described in detail. The WBAN is equipped with a heart beat sensor and GPS receiver has been implemented for calculating the calories burned of a runner during exercises. A common used sensor has been used for heart beat measurements.

6.5.2 WBAN for runners

6.5.2.1 WBAN Description

For testing purposes, two WBANs Runner_Fadi and Runner_Lina have been created as depicted in Figure 17. Because the WBANID is a functional property, each WBAN can have only one WBANID. That is why the MAC address of the mobile where the application is downloaded was chosen as WBANID, knowing that the application can be downloaded once on the mobile. The WBAN has contact the runner identified by his email address. In order to calculate the burned calories the runner should be described by his age, gender, height and weight. The WBAN contains the GPS and the heart beat sensors and elects the Mobile as Hub. Figure 18 depicts a part of Runner_Lina's WBAN ontology instance.

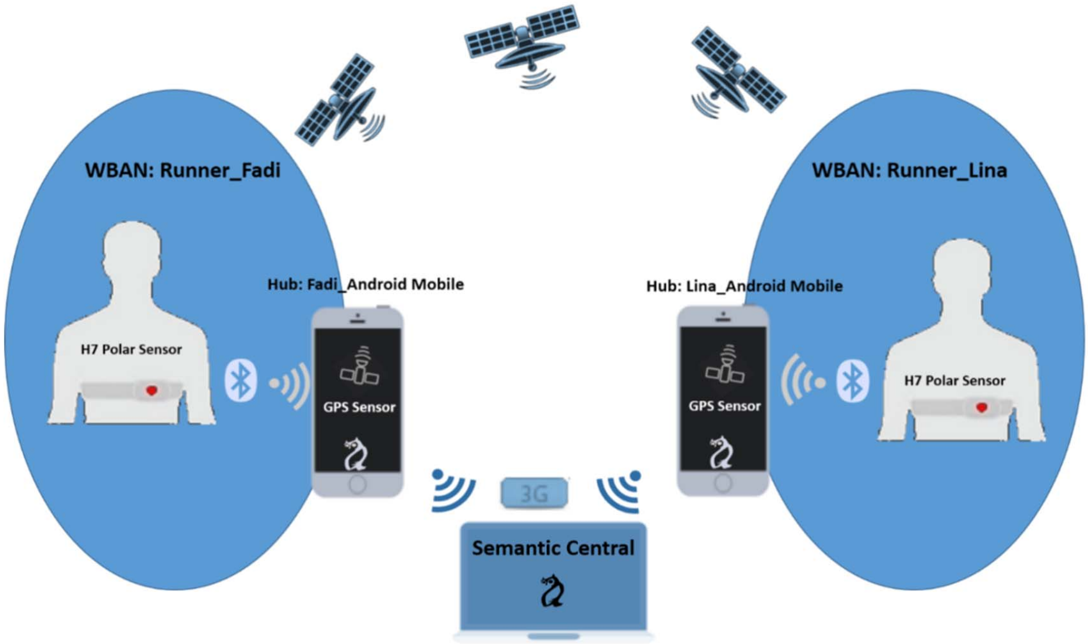


Figure 17: High-level architecture of the retained test scenario

The object property, "Contains Lina_AndroidMob" is inferred based on rule 2 defined in clause 6.2.4 of the present document..

The mobile phone is equipped with a Bluetooth LE interface and a 3G interface which connects the gateway of the WBAN to the internet. The GPS sensor is built in the mobile and used for tracking process. The common used heart beat sensor communicates with the mobile via the Bluetooth LE interface. This heart beat sensor is identified by the Serial number retrieved from its GATT general attributes (see clause A.1.2.3 of the present document.) and is used for the heart beat process. The heart beat sensor has the waist as position (individual of type BodySurface, the subclass of the location class).

While the tracking process has the meter (m) as unit of measurement, the heart beat process has beats per minute (bpm) as unit. The tracking process has two data, the latitude and longitude, and an additional derived data which is the speed calculated based on Haversine formula. The heart beat process has the HR data. Each data has measurements measured by a sensor.

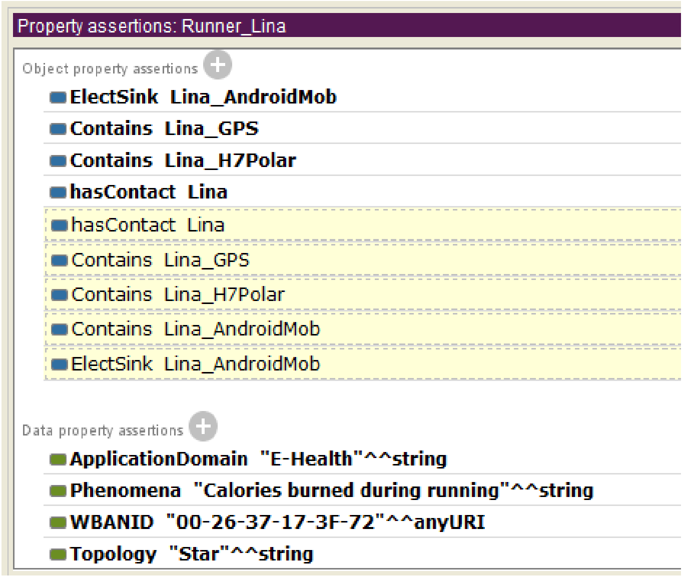


Figure 18: Part of Runner_Lina's WBAN ontology instance

To encourage user to run on higher speed, the maximum heart rate is calculated based on the following formula (see [i.12]):

Maximum heart rate= 208 - (0,7 × Age).

6.5.2.2 SWRL rules

The SWRL rule is added:

Patient(?P), MaxHR(?P, ?Q), Age(?P, ?Age), multiply(?x, 0,7, ?Age), subtract(?y, 208, ?x) -> hasValue(?Q, ?y).

If the heart beat sensor measures a value greater than the maximum heart rate, a notification will be sent to the user and the measurement will be considered as invalid in order to stop the running process. To do so, the following SWRL rule is added:

WBAN(?W); MaxHR(?P, ?Q); hasContact(?W, ?P); Contains(?W, ?S); measures(?S, ?Mes); usedFor(?S, HeartBeat); value(?Mes, ?measure); hasValue(?Q, ?max); lessThan(?max, ?measure) -> valid(?Mes, false).

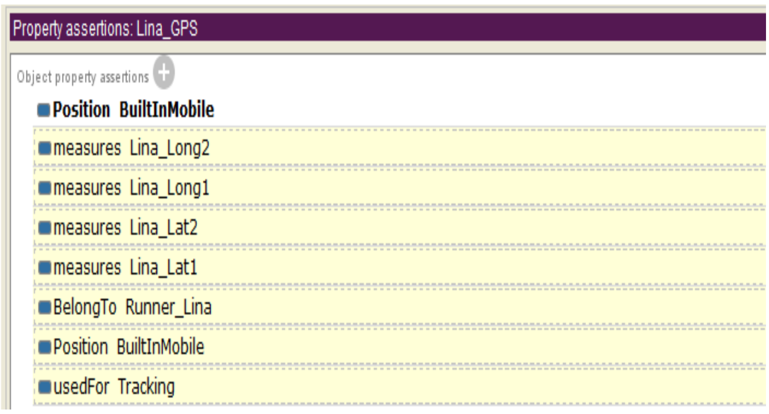


Figure 19: GPS Object properties

a) Figure 19 shows the inferred object properties of Lina_GPS sensor BelongTo Runner_Lina: inferred from the inverse property Contains (Lina_WBAN Contains Lina_GPS).

b) Measures: inferred from the inverse property is MeasuredBy (e.g. Lina_Long2 is MeasuredBy Lina_GPS).

c) usedFor: inferred from Rule 1 already defined in clause 6.2.4 of the present document.

The maximum heart rate for Lina should be 187. Thus, if the heart beat sensor measures a value greater than this maximum, the measurement should be invalid. Figure 20 shows an invalid measurement.

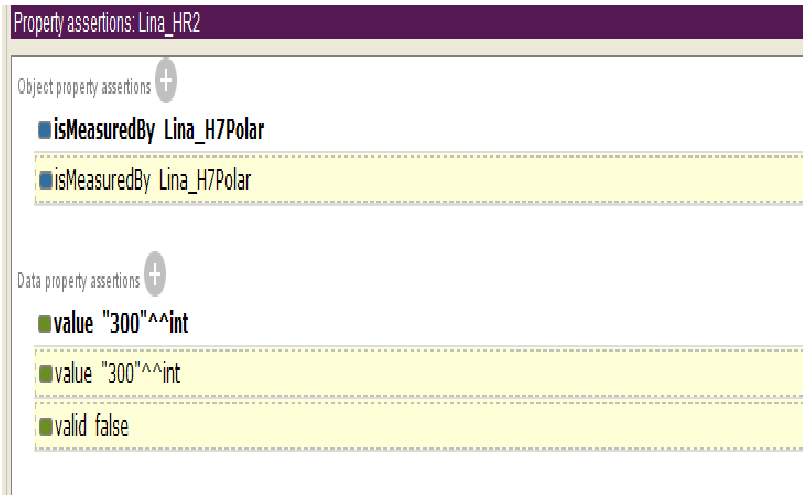


Figure 20: Invalid measurement caused by a value greater than the maximum limit

Using Pellet reasoner, the consistency of the SmartBAN ontology was verified. Pellet reasoner also pointed out that the SmartBAN ontology was well classified. The inferred value was retrieved. More rules and verification can be done to assure the well functionality of the proposed ontology.

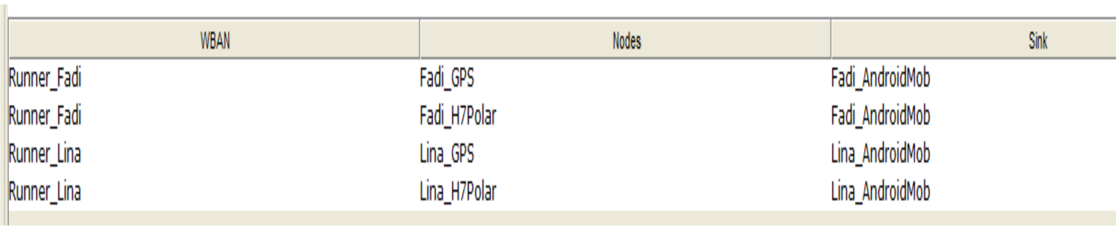
6.5.2.3 SPARQL queries

To retrieve the information, the SPARQL engine implemented in Protégé has been used. Unfortunately, this engine cannot retrieve inferred data.

Here is the example of the written query for node discovery:

```
SELECT ?WBAN ?Nodes ?Hub
    WHERE { ?WBAN full:Contains ?Nodes.
?WBAN full:ElectHub?Hub.
}
```

The result of the query is given in Figure 21.



WBAN	Nodes	Sink
Runner_Fadi	Fadi_GPS	Fadi_AndroidMob
Runner_Fadi	Fadi_H7Polar	Fadi_AndroidMob
Runner_Lina	Lina_GPS	Lina_AndroidMob
Runner_Lina	Lina_H7Polar	Lina_AndroidMob

Figure 21: Node discovery SPARQL result

7 SmartBAN modular ontologies mapping with of interest reference ontologies

7.1 Introduction

The objective of clause 7 of the present document is to address the mapping between SmartBAN modular ontologies and others of interest reference ontologies for smart BANs.

7.2 Main reference ontologies to be considered

7.2.1 Semantic Sensor Network (SSN/SOSA) ontology

Semantic Sensor Network (SSN) ontology is the most important ontology that describes sensors and observations, and related concepts [i.13]. However , it does not model domain concepts, time, locations, etc. Developed by W3C Semantic Sensor Network Incubator Group, SSN also includes a core ontology called 'Sensor, Observation, Sample, and Actuator' (SOSA) for its elementary classes and properties. It is briefly described in clause A.1.1.4 of the present document.

7.2.2 oneM2M ontology

oneM2M ontology [i.14] describes key classes, relations and properties that are necessary to enable semantic functionalities and interoperability between applications. This ontology is based on a Thing that has properties through ThingProperty. In oneM2M, a Device is a subclass of Thing that is designed to accomplish a Functionality represented in the network by a Service. To communicate over the network a Service uses an Operation.

7.2.3 Smart Applications REFerence model (SAREF) ontology

SAREF ontology is a reference ontology for IoT created in close interaction with the industry during a study requested by the European Commission in 2015 and subsequently transferred into an ETSI Technical Specification [i.15]. It contains core concepts that are common to several IoT domains, like e.g. a Device made to accomplish a task by performing one or more functions in a defined space. Dedicated SAREF extensions can also be created in order to handle specific data elements for a certain domain. As a reference ontology, SAREF serves as the means to connect the extensions in different domains.

7.2.4 SAREF ontology extension for the eHealth/Ageing Well domain (SAREF4EHAW)

SAREF4EHAW (see https://saref.etsi.org/saref4ehaw/v1.1.1/, [i.16]) is an extension of SAREF ontology [i.15] for the eHealth/Ageing-well (EHAW) vertical. SAREF4EHAW extension has been specified and formalized by investigating EHAW domain related resources, as reported in ETSI TR 103 509 [i.17], such as: potential stakeholders, standardization initiatives, alliances/associations, European projects, EC directives, existing ontologies and data repositories. It therefore both:

- Allows the implementation of a meaningful limited set of typical EHAW related use cases already identified in [i.17], i.e.:
  - use case 1 "monitoring and support of healthy lifestyles for citizens";
  - use case 2 "Early Warning System (EWS) and Cardiovascular Accidents detection".
- Fulfils the EHAW related requirements provided in [i.17], mainly the ontological ones that were mostly taken as input for the ontology specification.

SAREF4EHAW mainly reuses the following existing ontologies:

- SAREF [i.15];
- SmartBAN (V1.1.1 of the present document [i.58]);
- SAREF4ENVI (see https://saref.etsi.org/saref4envi/v1.1.2/) [i.60]; and
- SSN [i.13].

For embedded semantic analytics purposes, SAREF4EHAW has also been designed using the modularity principle and can thus be mainly described by the following self-contained knowledge modules: HealthActor, Ban, HealthDevice, Function (measured data related concepts included) and Service.

7.3 SmartBAN core ontologies mapping with main reference ontologies

7.3.0 Introduction

The objective of clause 7.3 of the present document is to describe the mapping of SmartBAN modular ontologies (SmartBAN Reference model) with main reference ontologies (i.e. those aforementioned in clause 7.3 of the present document).

7.3.1 Mapping of classes

Table 2 presents the mapping of SmartBAN modular ontologies classes (i.e. ontology concepts) with those of main reference ontologies already aforementioned in clause 7.2 of the present document. This class mapping is of the type owl:equivalentClass.

Table 2: Class mapping between SmartBAN Reference Model and oneM2M base ontology, SAREF, SOSA, SAREF4EHAW

SAREF classes	oneM2M classes	SOSA classes	SmartBAN classes	SAREF4EHAW classes
	Thing			
Device	Device		Node	HealthDevice, a saref:Device
	InterworkedDevice			
		Plateform		
		System		
			Ban	Ban
			Contact	Contact
				HealthActor, a BAN contact is an HealthActor
Sensor		Sensor	Sensor	HealthSensor
Actuator		Actuator	Actuator	HealthActuator
				HealthWearable
			Hub	BanHub
Service	Service		Service	Service
A service exposes one or more functions (*functionalities, Function class*)	Function, a service exposes one or more functions (*functionalities, Function class*)		ServiceProfile	Serviceprofile
	Operation		ServiceGrounding	ServiceGrounding
			ServiceProcess	ServiceProcess
Function	Function	Procedure	Process	Function
SensingFunction	MeasuringFunction	Observation	MeasurementProcess	MeasurementFunction
ActuatingFunction	ControllingFunction	Actuation	Action, a Process	ActuatingFunction
Command	Command			Command
A service process has data input (hasInput data property)	OperationInput	Input	Input	Input
A service process has output input (hasOutput data property)	OperationOutput	Output	Output	Output
Property		Property		
FeatureOfInterest		FeatureOfInterest		
State	Aspect			
Task				
			Mode	Mode
			DeviceType	DeviceCharacteristic
			Processor	ComputingPower
			EnergySource	PowerSource
			Interface	Interface

SAREF classes	oneM2M classes	SOSA classes	SmartBAN classes	SAREF4EHAW classes
			Operating, Data object property	OperatingConstraint, HealthDevice object property
			Data	Data
			Constraints	DataConstraint
Measurement		Observation	Measurement	Measurement
				TimeSeriesMeasurement
				FrequencyMeasurement
UnitOfMeasure	Metadata concept could be used *(no direct equivalence, Metadata contains data like units*)		Uom (*Process level*)	UnitOfMeasure (*Measurement level*)
				MeasurementCollectionSession
State, the state in which a device can be found				
	Variable, and entity that stores some data			

7.3.2 Mapping of object properties

Table 3 presents the mapping of SmartBAN modular ontologies object properties with those of main reference ontologies already aforementioned in clause 7.2 of the present document. This object properties mapping is of the type owl:equivalentProperty.

Table 3: Object properties mapping between SmartBAN Reference Model and oneM2M base ontology, SAREF, SOSA, SAREF4EHAW

SAREF Object Property	oneM2M Object Property	SOSA Object Property	SmartBAN Object Property	SAREF4EHAW Object Property
consistOf (*Device level*)	consistOf (*Device level*)			consistOf (*Device level*)
offers (*Device level*)	hasService (*Device level*)		engagedIn (*Node level*)	Offers (*Device level*)
hasFunction (*Device level*)	hasFunction (*Device level*)	Implements (Entity level)	hasProcess (*Node level*)	hasFunction (*Device level*)
			hasInterface (*Node level*)	hasInterface (*Device level*)
			hasdeviceType (*Node level*)	hasdeviceCharacteristic (*Device level*)
				hasOperatingConstraint (*HealthDevice level*)
			hasConstraints (*Data level*)	hasConstraint (*Data level*)
			InCurrentMode (*Node level*)	InCurrentMode (*HealthDevice level*)
				isAttachedTo (*HealthDevice level*)
				hasParticipant (*MeasurementCollectionSession level*)
				usesBAN (*HealthActor level*)
				hasHub (*BAN level*)
			contains (*BAN level*)	contains (*BAN level*)
			hasResponsibleParty (*BAN level*)	hasResponsibleParty (*BAN level*)

SAREF Object Property	oneM2M Object Property	SOSA Object Property	SmartBAN Object Property	SAREF4EHAW Object Property
			hasContact (*BAN level*)	hasContact (*BAN level*)
			hasProcessor (*DeviceType level*)	hasComputingPower (*Devicecharacteristic level*)
			hasEnergySrc (*DeviceType level*)	hasPowerSource (*Devicecharacteristic level*)
hasCommand (*Function level*)	hasCommand (*Function level*)			hasCommand (*Function level*)
hasMeasurement (*FeatureOfInterest level*)			hasMeasurements (*Data level*)	hasMeasurement (*Data level*)
				hasTimeSeriesMeasurement (*Data level*)
represents (*Service level*)	exposesFunction (*Service level*)		presents (*Service level*)	presents (*Service level*)
	hasOperation (*Service level*)		supports (*Service level*)	supports (*Service level*)
			discribedBy (*Service level*)	isDiscribedBy (*Service level*)
makesMeasurement (*Device level*)		observes (*Sensor level*)	measures (*Sensor level*)	makesMeasurement (*Device level*)

Annex A (informative): Background and SoA

A.0 Introduction

This annex presents the SoA, i.e. existing documents, works and architectures that have to be investigated for identifying:

- specifications that should be reused and directly standardized;
- closely related frameworks from which the present document should be inspired; and
- missing parts that have to be fully specified in the present document;

for coming up with the reference specifications of the Smart BANs' open data model. This model is one of the building blocks providing solutions for heterogeneity management in Smart BANs.

A.1 Existing sensor/actuator and BAN data representation

A.1.0 Introduction

In modern medical services and eHealth environments, patients' care information is shared among multiple stakeholders and is also stored in a distributed way for enhancing the management of patient's care. This information, plus data range from patient's medical records, is not only stored in repositories located in patients' care places, but can also be merged with a variety of additional information related to medical research, pharmaceutical products, and information stored within social networks of healthcare interest groups. In that context, a common understanding is needed to mask the heterogeneity of this information. Ontologies represent one technology that can contribute to this goal.

Therefore, the scope of clause A.1 of the present document is to investigate the wireless sensor and the eHealth domains in order to list and evaluate the existing sensor and sensor data representation models in the context of TC SmartBAN. The common used models and corresponding ontologies will be identified, compared and confronted to Smart BAN requirements in order to select the one that have to be adopted or to make known the gaps that have to be addressed by TC Smart BAN Work Item 1.1.

If a SmartBAN dedicated ontology has to be specified and qualified in the present document, common used ontology validation methods will also be presented for that purpose in that SoA/survey in clause A.1.3 of the present document.

A.1.1 Wireless Sensor Networks (WSNs) common used data representation models and ontologies

A.1.1.0 Introduction

In clause A.1.1 of the present document, an overview of existing data representation formats and ontologies of WSN sensors/actuators will be presented. Those ontologies will also be confronted to Smart BAN and eHealth requirements in order to eventually identify the gaps that have to be addressed by TC Smart BAN Work Item 1.1.

A.1.1.1 OGC's Observations & Measurements (O&M) and SensorModel Language (SensorML)

A.1.1.1.1 Observations and Measurements (O&M)

"Observations and Measurements" (O&M [i.18]) model provides XML encoding extensions for the description of sensor observations (i.e. sensor data) and their management. It was proposed by the Open Geospatial Consortium (OGC or OpenGIS) in year 2000 and published as an ISO standard [i.19]. O&M is specified and maintained as a standalone component under the OGC Sensor Web Enablement (SWE) initiative [i.20]. It is presented below.

O&M defines the observation as an act associated with a discrete time period through which a number, term or other symbol, is assigned to a phenomenon. The phenomenon is a *property* of an identifiable object, which is the 'feature of interest' of the observation. The 'feature of interest' is a representation of the observation target, being the real-world object regarding which the observation is made. The observation uses a *procedure*, which is often a device or sensor but may also be a process chain, a human observer, an algorithm, a computation or a simulator. The *procedure* is used to generate the result. The *observedProperty* identifies or describes the phenomenon for which the observation result provides an estimate of its value. Figure A.1 shows the basic observation model defined in O&M.

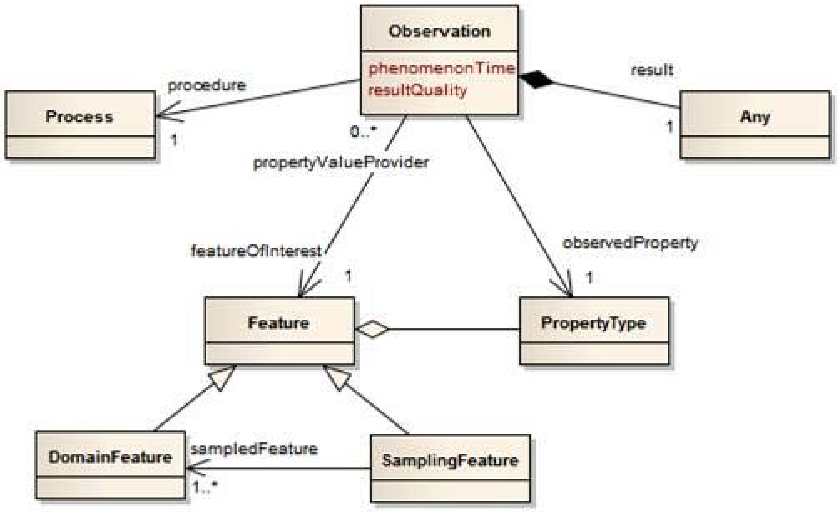


Figure A.1: Basic observation model

An *observation* has a *samplingTime*, which is the time that the result applies to the 'feature-of-interest'. In addition, an *observation* has a *resultTime*, which is the time when the procedure associated with the observation act was applied. An observation *parameter* is a general event-specific parameter that will typically be used to record environmental parameters, or event-specific sampling parameters that are not tightly bound to either the 'feature-of-interest' or the procedure. An *observation* may have both metadata and an indication of the event-specific *resultQuality*. Figure A.2 summarizes the characteristics of an *observation*.

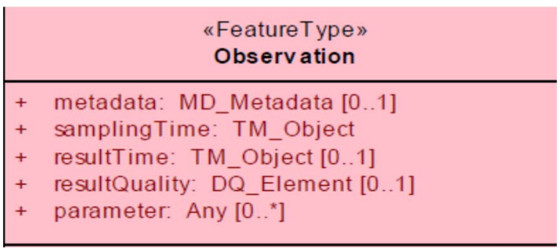


Figure A.2: Observation's properties

An observation could also have different feature's types. Figure A.3 shows the different feature types of an observation and its results.

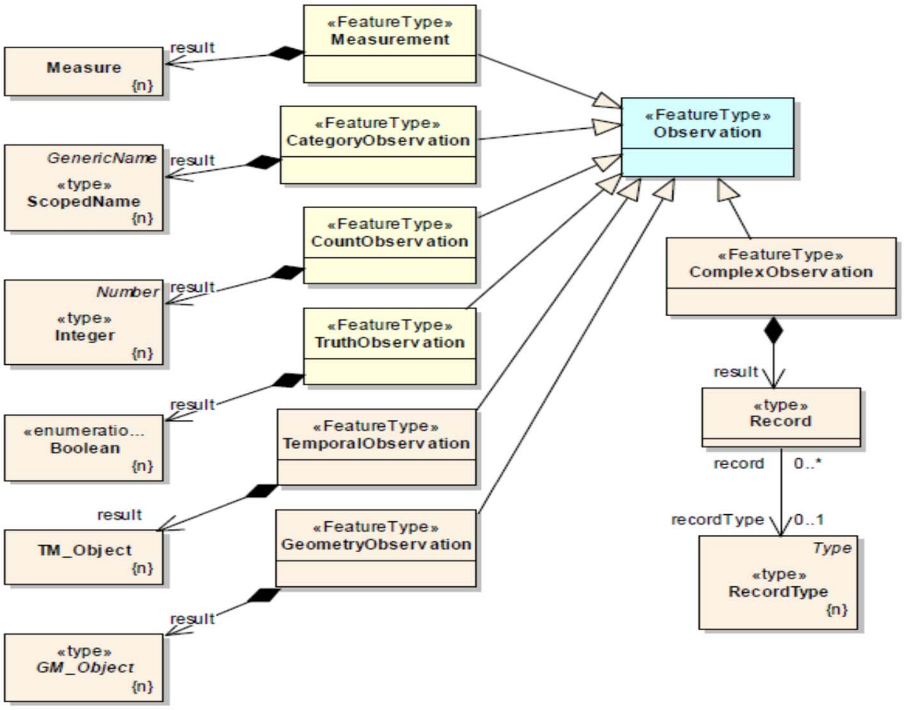


Figure A.3: Specialization of observation by result type

Moreover, there are sensors that are capable of doing these observations and measurements. That is why OGC defined the SensorML language in order to provide all the attributes required for processing, registering, and assessing the quality of measurements from sensor systems. The details of SensorML are provided in clause A.1.1.1.2 of the present document.

A.1.1.1.2 Sensor Model Language (SensorML)

SensorML is an XML model and encoding extensions for sensor and sensor data description/processing that has been proposed by the OGC in 2007 [i.21]. It is specified and maintained as a standalone component under the OGC Sensor Web Enablement (SWE) initiative [i.20]. Even if SensorML was initially dedicated for the earth observation domain, it actually serves as basic language for the formalization of many of the existing sensor ontologies. It is described below.

SensorML was defined for handling two possible roles. The first one is to describe the procedure by which an existing observation was obtained. The second one is to provide processing chains from which SensorML-enabled software could derive on-demand new data from existing observations.

In SensorML, all components are modelled as *processes*. This includes components, normally viewed as hardware (including transducers and actuators) and processors, sensors and platforms. *Processes* take input and generate output through the application of an algorithm defined by a method and parameter values. Figure A.4 shows the conceptual model of SensorML.

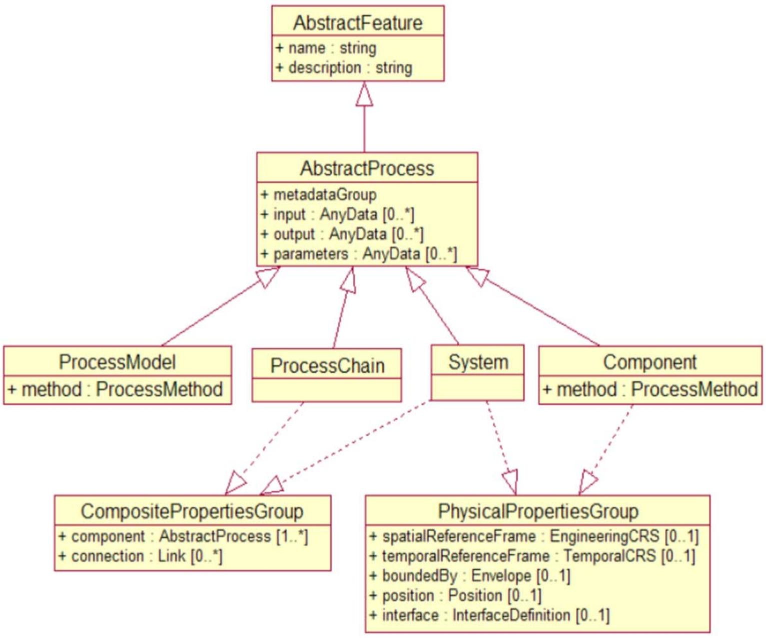


Figure A.4: Conceptual model of SensorML

Within SensorML, all *processes* are derived from an *AbstractProcess* which is itself derived from an *AbstractFeature*. *ProcessModel*, presented in Figure A.4, is used to define more or less atomic pure processes that are expected to be used within more complex process chains. A process chain is defined by:

- the description of all of the *processes* in the chain; and then
- the description of the appropriate connections between components.

In addition to the description of purely mathematical processes, SensorML also describes the physical processes deriving from the abstract *PhysicalProcess*. This abstract *PhysicalProcess* have *spatialReferenceFrame* and *temporalReferenceFrame* properties. Those properties define any spatial or temporal Coordinate Reference Systems (CRS) that might be used for interrelating internal components within the process, as well as any reference frames that might allow this component to be related to other components within a process chain or system. *PhysicalProcess* may also have zero or more interfaces over which commands and data can flow based on particular physical connections and particular protocols. The interface property contains an *InterfaceDefinition* which is based on the Open Systems Interconnection (OSI) reference network layer model, as presented in Figure A.5.

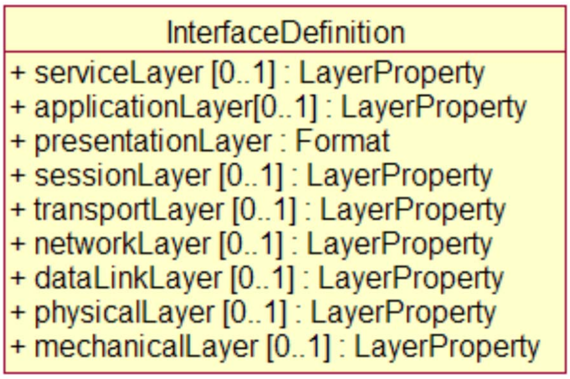


Figure A.5: Interface definition in SensorML

Within SensorML, any physical process can be modelled as a *Component*, if it either cannot be subdivided into smaller sub processes, or if one chooses to treat it as a single indivisible process. A *System* is a physical equivalent of a *ProcessChain*. A *System* may include several physical and non-physical processes that all act for providing a certain set of *System* outputs, based on the *System* inputs and parameters.

A.1.1.2 Existing WSN ontologies

In the last decade, lots of ontologies were proposed for describing WSNs. In [i.22], the authors state a summary of most important ontologies used. They were themselves inspired by [i.23] where a detailed analysis of sensor ontologies has been provided, based on two main topics: 'Sensor' and 'Observation'. This analysis is summarized in the table shown in Figure A.6.

		sensor								physical					observation					domain			
ontology	base concepts	sensor hierarchy	identity & manufacturing	contacting & software	deployment	configuration	history	components	action & process	location	power supply	platform	dimension, weight, etc.	operating conditions	data/observation	accuracy	frequency	response model	field of view/sensing	units of measurement	feature/quality	sampled medium	time
MMI	sensor (system) & process		✓		✓	✓		✓	✓			✓	✓	✓		✓	✓	✓			✓	✓	
CSIRO	sensor & process	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		
OOSTethys	component, system & process							✓	✓			✓			✓						✓	✓	
CESN	sensor	✓			✓					✓					✓						✓		✓
SWAMO	agent, process & sensor			✓				✓	✓	✓		✓			✓				✓	✓	✓		✓
Kim	sensor					✓				✓					✓	✓	✓		✓		✓		
OntoSensor	component & sensor	✓				✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓
Eid	sensor		✓			✓				✓	✓				✓	✓	✓	✓	✓				
Matheus	system & sensor	✓		✓					✓	✓					✓		✓						
Avancha	sensor			✓						✓	✓				✓	✓	✓	✓	✓	✓	✓	✓	
ISTAR																							

Figure A.6: Comparison of the existing ontologies based on Sensor & Observation

Figure A.6 shows that CSIRO ontology is the ontology which fully describes the sensor and its physical features. However it does not describe the data transferred between sensors.

Table A.1 summarizes the comparison done in [i.22] where the authors have added 'data' as a third topic for comparing the existing WSN ontologies.

Table A.1: Comparison between existing ontology based on Sensor, Observation & Data

	Sensor	Observation	Data								
Ontologies	**8 facets**	**5 facets**	**Data**	**Data Stream**	**Sensing Process**	**Communicating Process**	**Transforming Process**	**States**	**Data Quality**	**Acquisition Policy**	**Communication Policy**
SSN Ontology	**8/8**	**4/5**	*		*					*	
CESN Ontology	**2/8**	**1/4**	*		*						
CSIRO Ontology	**8/8**	**4/5**	*		*					*	
Sensei O&M Ontology	**Not available**	**Not available**	*		*						
OOSTethys ontology	**2/8**	**2/5**	*		*						
MMI Ontology	**5/8**	**Not available**			*					*	
SWAMO Ontology	**3/8**	**2/5**			*						
SEEK Ontology	**Not available**	**Not available**	*		*						
SDO Ontology	**2/8**	**2/5**	*		*					*	
SeRes O&M Ontology	**Not Available**	**Not available**	*		*						
OntoSensor Ontology	**5/8**	**5/5**	*		*					*	

A detailed analysis of the comparisons, presented in Table A.1, shows that SSN ontology seems to be the best ontology for describing sensors, contrary to OntoSensor Ontology that seems to be best one for describing observations and data. Clauses A.1.1.3 and A.1.1.4 of the present document will detail those two common used ontologies.

A.1.1.3 OntoSensor ontology

OntoSensor is stemming from a combination of SensorML defined by OGC, IEEE Suggested Upper Merged Ontology (SUMO) and the International Organization for Standardization [i.25]. Its latest update was released in 2008. The main objective of OntoSensor is to provide an ontological description for data observation of sensors. Figure A.7 shows the different parts of OntoSensor Ontology.

OntoSensor ontology [i.24] supports data discovery, processing and analysis of sensor measurements, geo-location of observed values (measured data), performance characteristics (e.g. accuracy, threshold, etc.), explicit description of the process by which an observation was obtained (i.e. its lineage). It also provides an executable process chain for deriving new data products on demand.

Although OntoSensor illustrates a semantic approach to sensor description and provides an extensive knowledge model, there is no distinctive data description model to facilitate interoperable data representation for sensors observation and data. This is however quite mandatory for Smart BANs.

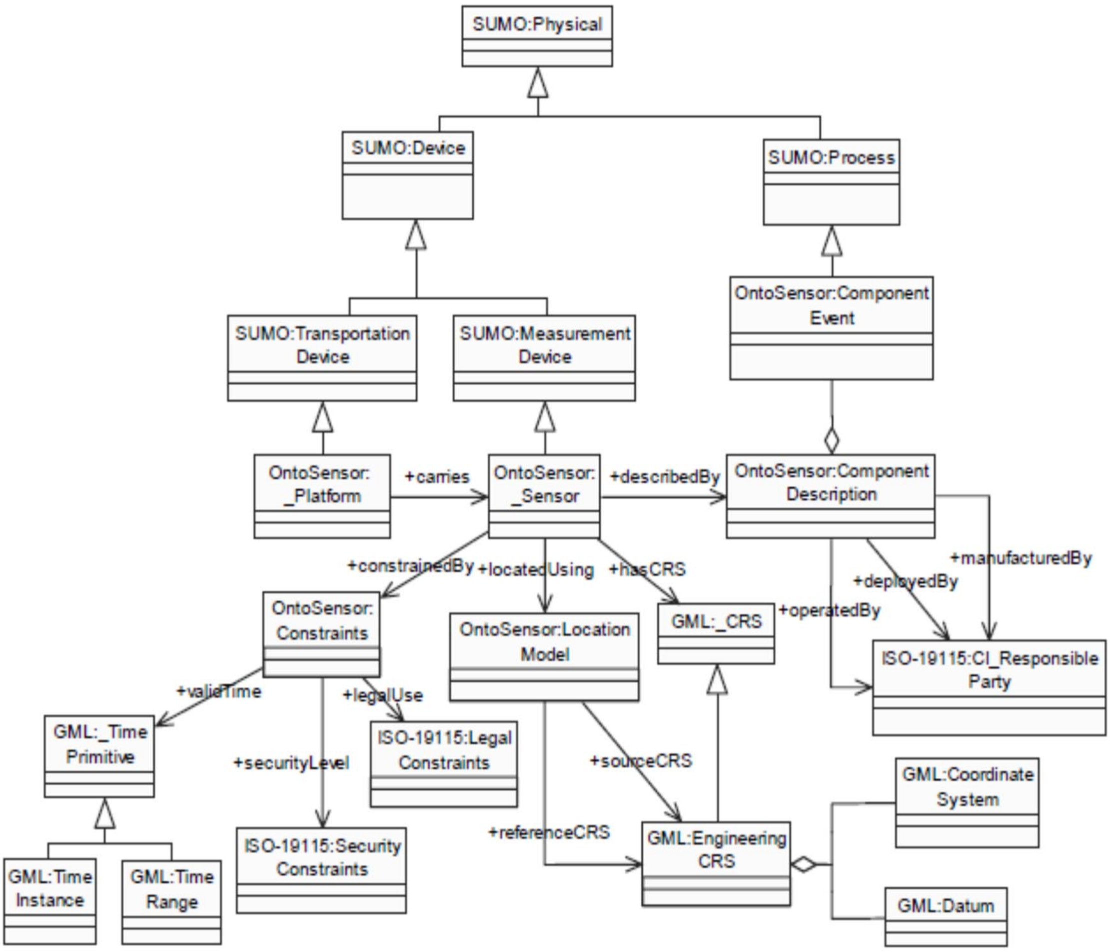


Figure A.7: OntoSensor ontology

A.1.1.4 Semantic Sensor Network ontology (SSN)

The most important ontology that includes sensor, observation and data description is Semantic Sensor Network ontology (SSN) [i.26]. Developed by W3C Semantic Sensor Network Incubator Group, this ontology describes the capabilities of sensors, the measurement processes used and the resultant observations. SSN can also be aligned with other ontologies that describe e.g. observed phenomena. This ontology covers large parts of the SensorML and O&M standards from the OGC, omitting calibrations as well as process descriptions and data types, which were deemed not sensor specific (if required it can always be included from other ontologies). However, SSN does not provide facilities for abstraction, categorization, and reasoning offered by semantic technologies.

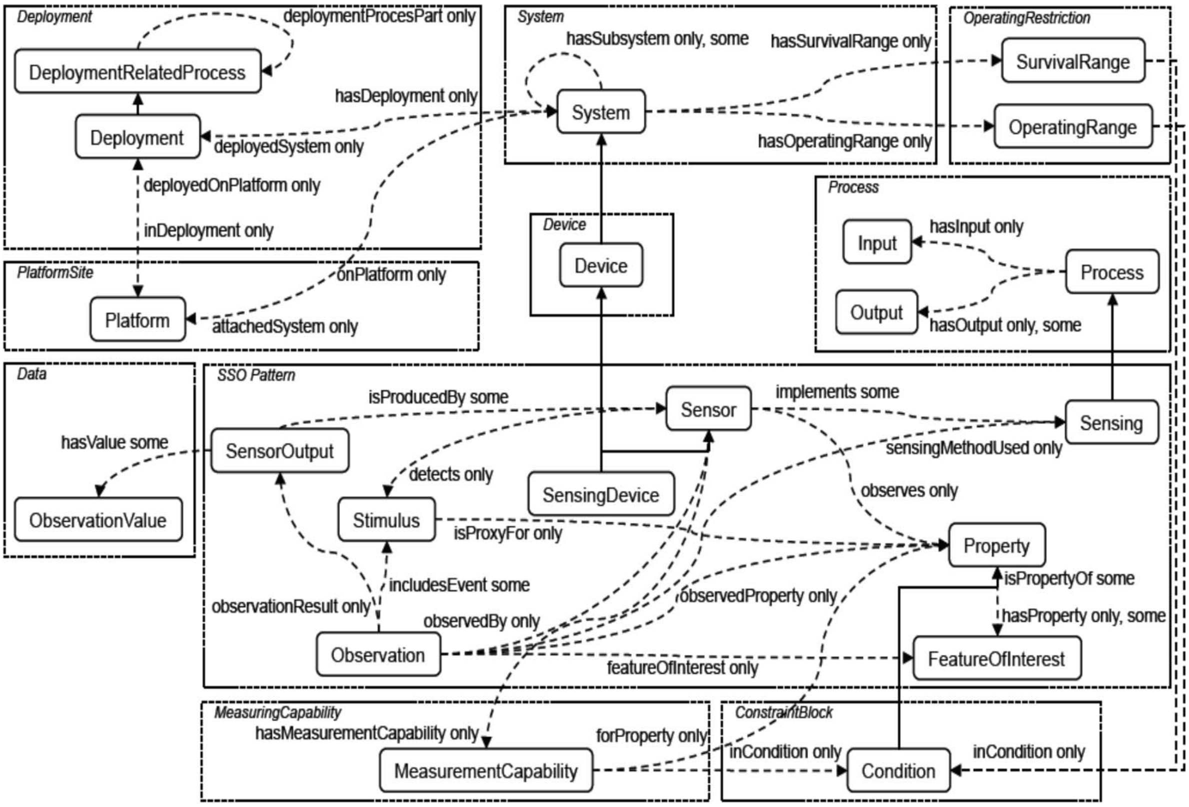


Figure A.8: SSN ontology

Figure A.8 shows that SSN is focusing on the deployment of the wireless sensor network, the description of the sensors and the phenomena it is measuring. The ontology can be used for a focus on any (or a combination) of a number of perspectives:

- A sensor perspective, with a focus on what is sensed, and how it is sensed.
- A data or observation perspective, with a focus on observations and related metadata.
- A system perspective, with a focus on systems of sensors.
- Or a feature and property perspective, with a focus on features, their properties, and what can sense those properties.

However, SSN ontology does not focus on the communication process. In order to overcome this limitation, W3C members introduced, in the eleventh International Semantic Web Conference, an extension of SSN ontology called Wireless Semantic Sensor Network ontology (WSSN). This extension is presented in clause A.1.1.5 of the present document.

A.1.1.5 WSSN ontology

WSSN was proposed by W3C to extend SSN with three new concepts: the communication process, the data stream and the state. The "Communicating class" is added to the ontology for describing the sensor communication process. In addition to that, a "Communication Output class" is added for describing data transmitted by the sensor (the communicating device). The resulting data stream, if a communication occurs, is defined as a set of communications while the acquisition data stream is defined as a set of observations. A communicated data can be equal to a data acquired by sensor or to a data generated from a set of data acquired by sensors using, for example, some aggregation procedure.

The WSN node and the observed phenomenon pass through several states. In the WSSN ontology [i.22], the state of any entity is represented by the "State class".

A.1.1.6 Semantic Web Based Architecture for Managing Hardware Heterogeneity

A new semantic ontology was proposed in 2011 in [i.27] by Nikolić, Penca, Segedinac and Konjović for managing hardware heterogeneity in wireless sensor network. The proposed architecture, "Semantic Web Based Architecture for Managing Hardware Heterogeneity", consists of three segments:

- application;
- middleware; and
- WSN.

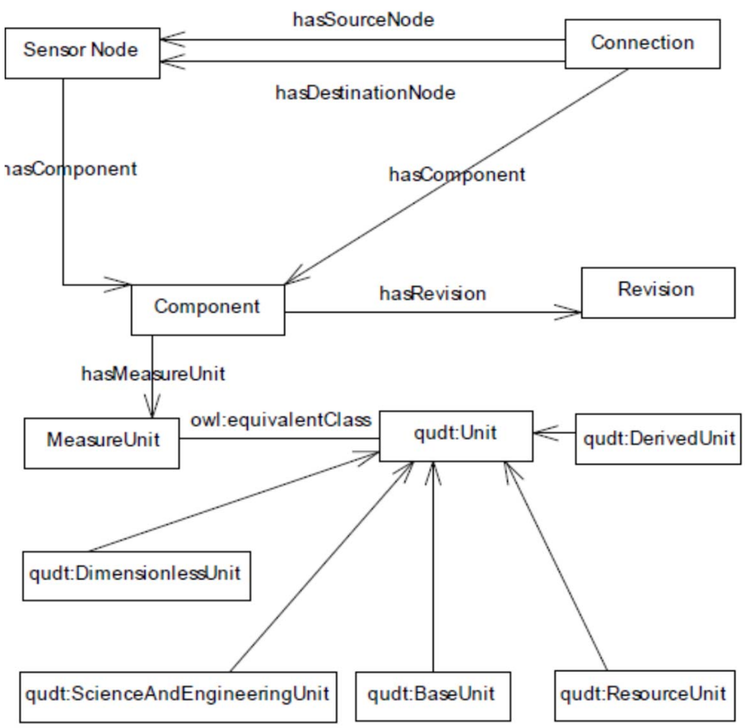


Figure A.9: WSN in Semantic Web Based Architecture

Figure A.9 shows that this ontology describes the sensor, its physical features (dimension, platform.), the connection and the data measured. It uses the same features described in SensorML to describe a node property, a component and the data. In addition, it includes the following additional features:

- Data & Data Stream. The ontology uses DataTime and Value fields in the "ReadingSensorstotable" to describe the data stream.
- Communication Process & Policy. The ontology includes Radio component that describes the RadioType and RadioFrequency.
- Acquisition Policy. This field indicates the interval of data acquisition.
- Data Quality. However, this field is not covered by the ontology.

Moreover, "Semantic Web Based Architecture for Managing Hardware Heterogeneity" ontology includes a Processing Component composed of Memory Component, not mentioned in the ontologies cited previously. Figure A.10 shows the Component structure in this ontology.

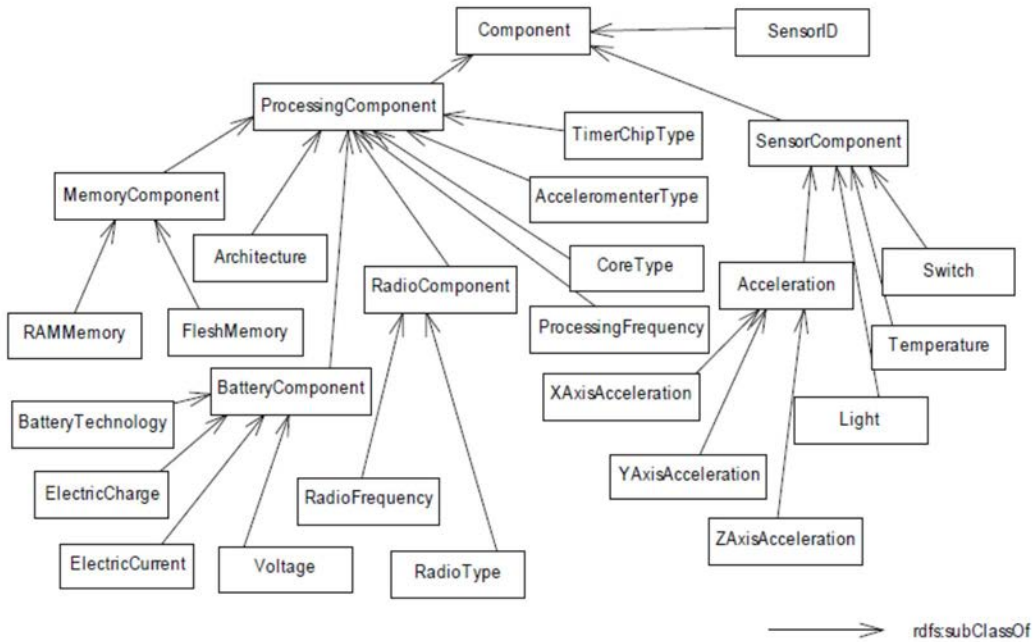


Figure A.10: Component Structure in Semantic Web Based Architecture

The Processing component presented in Figure A.10 may be very useful for the management of the Wireless sensor Network.

A.1.2 Proposed sensor ontologies in the eHealth sector

A.1.2.1 CEN TC 251 Health Informatics

CEN/TC251 [i.28] has been founded in 1990. Its objective is the standardization in the field of health Information and Communications Technology (ICT) to achieve compatibility and interoperability between independent systems and to enable modularity.

CEN/TC251 [i.28] has developed ISO 13606 [i.29] for Electronic health record communication which follows a dual model methodology. The CEN/TC251 ISO 13606 [i.29] norm is composed of five parts that are summarized below.

Part 1: Reference Model

The Reference Model formalizes the global characteristics of health record components, i.e. how they are aggregated, and the context information required to meet ethical, legal and provenance requirements.

Entry, Composition (which is the set of information committed to one Electronic Health Record- EHR - result) and elements of each EHR describe the clinical data related to the patient, the diseases and the collected data. The cluster allows the specification of nested multi-part data structures such as time series, list or tables.

Each element has one Data_Value. The Data_Value of the reference model describes all the types of data that can be found in the biomedical environment. Figure A.11 summarizes the Data_Value types.

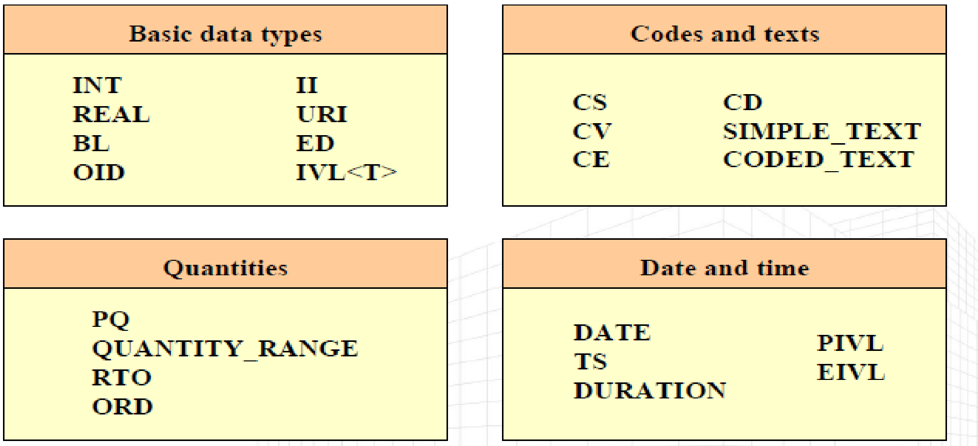


Figure A.11: Data_Value in Reference Model

Part 2: Archetype Model

An archetype is expressed in the form of constraints on the Reference Model. These constraints can consist in defining relevant attribute values, optionality and multiplicity of attributes and objects, data types and value ranges that data instances may take, and finally binding to clinical terminologies or ontologies. The archetype model will group the data collected from the ontology to respect the adequate format and repartition.

Part 3: Reference archetypes and term lists

The term lists contain a list of biomedical terms. They can be used as data dictionary for the service layer.

Part 4: Security

Health records have to be created, processed and managed in ways that: guarantee the confidentiality of their contents and legitimate control by a patient in their usage.

Part 5: Interface specification

The interface specification defines the computational view point of each interface as the payload to be communicated.

A.1.2.2 IEEE^{TM} reference models for medical device communications

ISO/IEEE 11073 [i.30] Personal Health Device (PHD) standards are a group of standards addressing the interoperability of Personal Health Devices (PHDs) such as weighing scales, blood pressure monitors, blood glucose monitors, etc. The standards draw upon earlier ISO/IEEE 11073 [i.30] standard works, but differ from those earlier works due to an emphasis on devices for personal use (rather than hospital use) and the use of a simplified communication model. The central core of the standard is the so-called Domain Information Model. Objects containing vital-sign data representations and their relationships are defined in this model. Objects for additional services around vital signs data objects, are also defined. The ISO/IEEE 11073 [i.30] model is summarized in Figure A.12.

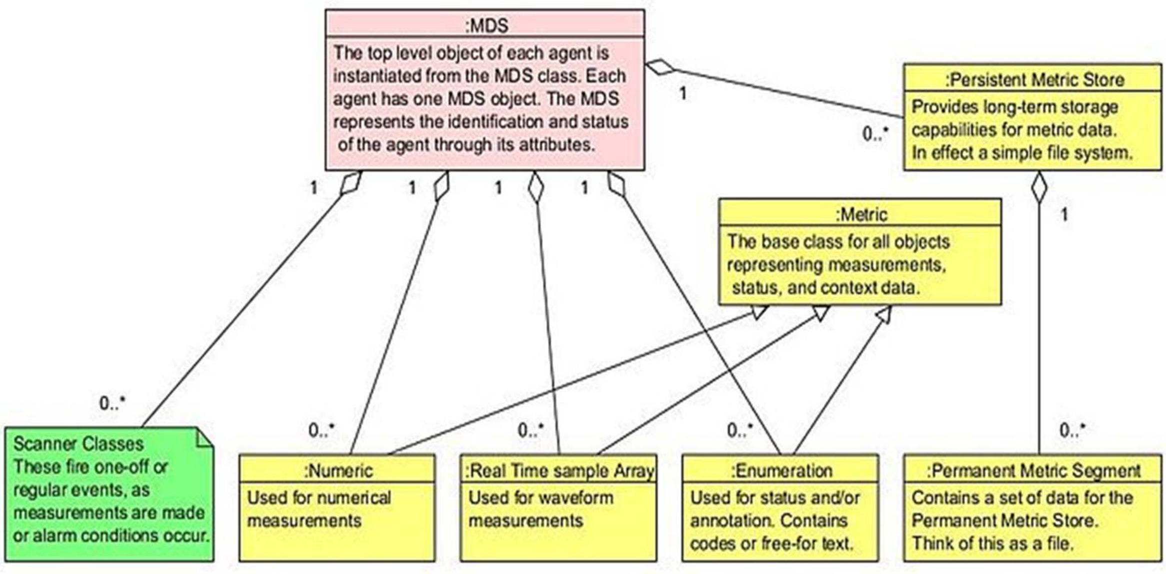


Figure A.12: ISO/IEEE 11073 [i.30] model summary

Numeric objects relate to the physiological parameters and have as attributes the mechanism to obtain an observed value and its status such as units and the timestamp. The Real Time and Enumeration attributes are used to represent a group of data. The Scanner Class determines the way the data is transmitted from sender to destination. It provides mechanisms to reduce message overhead by communicating the message format at configuration time and then sending pure data during runtime. The Storage Class, called PM-Store, provides a facility to store data for longer periods of time and also provides mechanisms to transfer portions of data that are needed by the receiver. Within the ISO/IEEE 11073 [i.30] standard, device specialization sub-standards have been described.

A.1.2.3 Bluetooth LE (Low Energy) profiles for medical devices proposed by Continua Alliance

Bluetooth LE (Low Energy) [i.31] was proposed and specified by the Bluetooth SIG for providing the present document with lower energy working process, thus allowing its usage in button cells and energy harvesting applications (like e.g. eHealth or Internet of Things applications). For that purpose, Bluetooth SIG members fully redesigned the physical and MAC layer of Bluetooth, as well as its protocol stack. At the upper layer level, they introduced a simplified device and application profile called GATT (Generic Attribute Profile), provided with the corresponding discovery and access protocol called Attribute Protocol. All the profiles that will be introduced within Bluetooth LE will be derived from the generic GATT one.

Within GATT, all the data are exposed as attributes. An attribute has:

- A unique identifier called UUID.
- A type.
- A value, with fix or variable length (from 0 to 512 octets).
- And a permission. The permission is used for setting the attribute's access rights for reading and or writing.

A device is characterize by the following attributes:

- Manufacturer Name, a String representing the name of the manufacturer of the device.
- Model Number, a String representing the model number that is assigned by the device vendor.
- Serial Number, a String representing the serial number of a particular instance of the device.
- Hardware Revision, a String representing the hardware revision for the hardware within the device.
- Firmware Revision, a String representing the firmware revision for the firmware within the device.

- Software Revision, a String representing the software revision for the software within the device.
- System ID, a structure containing an Organizationally Unique Identifier (OUI) followed by a manufacturer-defined identifier. This ID is unique for each individual instance of the product.
- Regulatory Certification Data List, representing regulatory and certification information for the product in a list defined within ISO/IEEE 11073-20601 [i.30].

Within the GATT specification, units are given using the international standards for the measurement of physical quantities, such as e.g. meter for length or kilogram for mass. The type of the units is determined in the 'Format Types' attribute.

The "Attribute Protocol" is a client/server based protocol providing attribute exchange functionalities between devices. GATT and "Application Protocol" upper layer modules are also associated with "Generic Access Profile" (GAP) and "Access Manager" modules that provide secure device discovery, access and connection functionalities. The corresponding device discovery, access and connection profiles are specified within the GAP module.

Based on that Bluetooth LE standard, the Continua alliance [i.31] specified new device/service profiles dedicated to medical devices and eHealth applications. It defines 13 different profiles that determine how services can be used to enable an application or use case. Some of these profiles are dedicated for notification, while some others are dedicated for measurement and for client server communication. These profiles are detailed below.

The HID Over GATT Profile (HOGP) defines the way a Bluetooth LE device can support Human Interface Device (HID; e.g. Heart Rate Monitor, Glucose Meter, Changed Oximeter, Mobile Phone) services using the Generic Attribute Profile.

The 'Measurement Profiles' are used to determine how a Connector could connect and interact with the sensor for being used in health care applications. It includes blood pressure, cycling power, cycling speed and cadence, glucose, temperature, heart rate, location and navigation, proximity, running speed and cadence.

The Time profile is used to get the date and time attributes.

The Alert Notification Profile (ANP) enables a client device to receive different types of alerts and event information. For example, the Phone Alert Status profile is used for obtaining the phone alert status exposed by a phone.

The Scan Parameters Profile (SPP) is used for providing devices with information assisting them in the management of their connection idle timeout and advertising parameters in order to optimize their power consumption and/or reconnection latency.

A.1.2.4 ASTM E31

ASTM - Committee E31 [i.33] Committee on Healthcare Informatics of American Standards for Testing and Materials (ASTM) develops standards related to architecture, content, storage, security, confidentiality, functionality, communication of healthcare information and decision making (including patient-specific information and knowledge). ASTM E31 has also developed some standards for defining health vocabularies, Electronic Health Record (EHR) system management (including interactions between different parties), as well as for ensuring security and privacy of Healthcare systems.

Four messaging standards where created:

- ASTM E1238-97 standard specification [i.34] for transferring clinical observations between independent computer systems (Withdrawn 2002).
- ASTM 1394-97 standard specification [i.35] for coded values used within the EHR. ASTM 1394-97 [i.35] allows the representation of different biomedical signals, with the cooperation of Health Industry Level 7 (HL7), for supporting multichannel signals (Withdrawn 2002).
- ASTM E1467-94 (2000) standard specification [i.36] for transferring digital neurophysiological data between independent computer systems (Withdrawn 2004). It defines the message structure exchanged between clinical instruments (Withdrawn 2004).

- ASTM E2369-12 standard specification [i.37] for the Continuity of Care Record (CCR). This recent specification gives a particular and detailed description of data (and its organization into structured textual forms) that are dedicated for the communication between healthcare enterprises and that may not be interoperable otherwise. Thes specification [i.37] details the abstraction of data contained in the EHR attributes. These data are documented in practice ASTM E1384-07 [i.38] and are organized and composed into a textual view for their transfer to other care settings that may not utilize a structured electronic representation of patient data. The textual form is used in order to provide a familiar cognitive format for information exchange between practitioners that are currently operating using traditional unstructured care settings.

A.1.3 Existing Semantic Web Service Architectures

A.1.3.1 OWL-S

OWL-S [i.40], the semantic markup for web services, developed by DARPA DAML program using OWL languages aims to describe semantic web services. It permits the automatic web service:

- discovery;
- invocation;
- composition;
- interoperation; and
- execution monitoring.

The OWL-S ontology includes mainly three sub-ontologies as shown in Figure A.13 and are the following: the service profile ontology that describes what the service does; the process model ontology that describes how the service is used; and the grounding ontology that describes how to interact with the service.

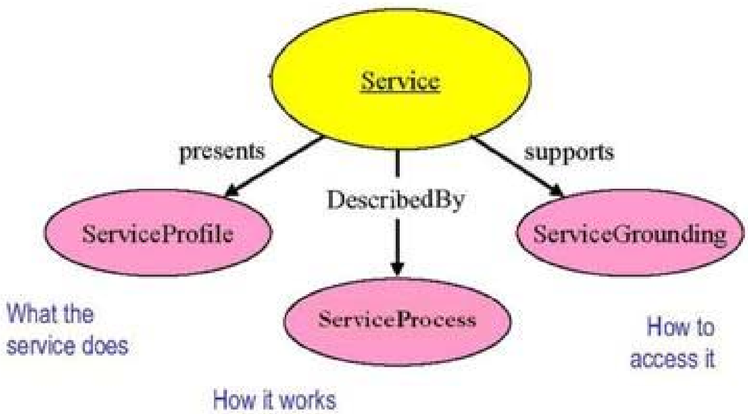


Figure A.13: Top Level of the Service ontology

Service Profile

The main aim of the service profile is to provide the automatic service discovery. On one hand, when a producer wants to publish a server, it should describe the service based on the service profile ontology. On the other hand, the requester will discover the capabilities of available servers based on the service Profile.

The service profile includes three different aspects:

- Functional aspect that determines the functionality of a service. The "hasProcess" object property, connects a "Profile" to a "Process" that has inputs, outputs, preconditions and effects produced during the execution of a service.
- Classification aspect that describes the type of service. It may precise the service's type based on a predefined Service Category.

- Non-functional aspect that makes the difference between services offering same service but in different manners. The non-functional parameters can be the QoS, privacy, security, etc.

Service Model

The service model contains a sub-class, "Process", that indicates to the consumer what he should do to invoke a service. The process could be atomic, composite (composed of atomic processes), or simple (used to provide abstracted views of atomic or composite processes).

While an atomic process has a single interaction between consumer and producer, a composite process has a set of processes linked together by a control flow (sequence, parallelism, etc.) and a data flow that describes how the data is interchanged between the processes.

Each process has:

- Participants: the client requesting the process and the server performing the process.
- Parameters: the input, output and local parameters.
- Pre-condition: a condition that the client should satisfy in order to invoke the service. This condition could be an expression, a SWRL rule or others.
- Result: obtained after the execution of the process. The result has a type and a value.

Service Grounding

The service grounding defines how abstract information can be realized by concrete information exchanges. It details the used transport protocol, port number and the message format. Actually, the service grounding class contains only the WSDL sub-class used to exchange SOAP messages.

Figure A.14 depicts a typical OWL-S interaction Model. A service provider advertise a service described by a service model and a service profile. A requester queries for a service. The OWL-S matchmaker verifies if there is an available service that matches the query, if yes, it provides the requestor by the URI of the available services. The requestor, then, choses an adequate server and retrieves the service model in order to invoke the service.

More researches can be investigated for the OWL-S matchmaker and matching algorithm.

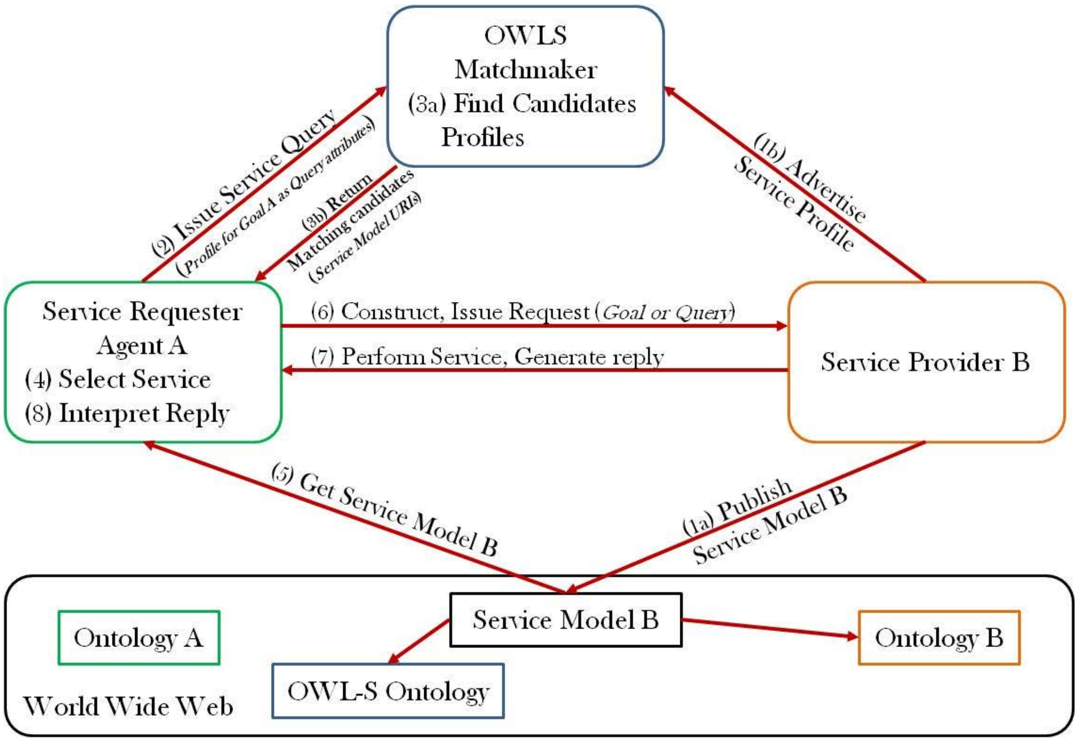


Figure A.14: Typical OWL-S Semantic Web Service interaction model

A.1.4 Existing generic service layers and enablers for in particular WSNs and WBANs

A.1.4.1 Service Oriented Architecture for WSN

In [i.41] S. Lan et al. present a SOA (Service Oriented Architecture) for WSNs. This architecture is depicted in Figure A.15.

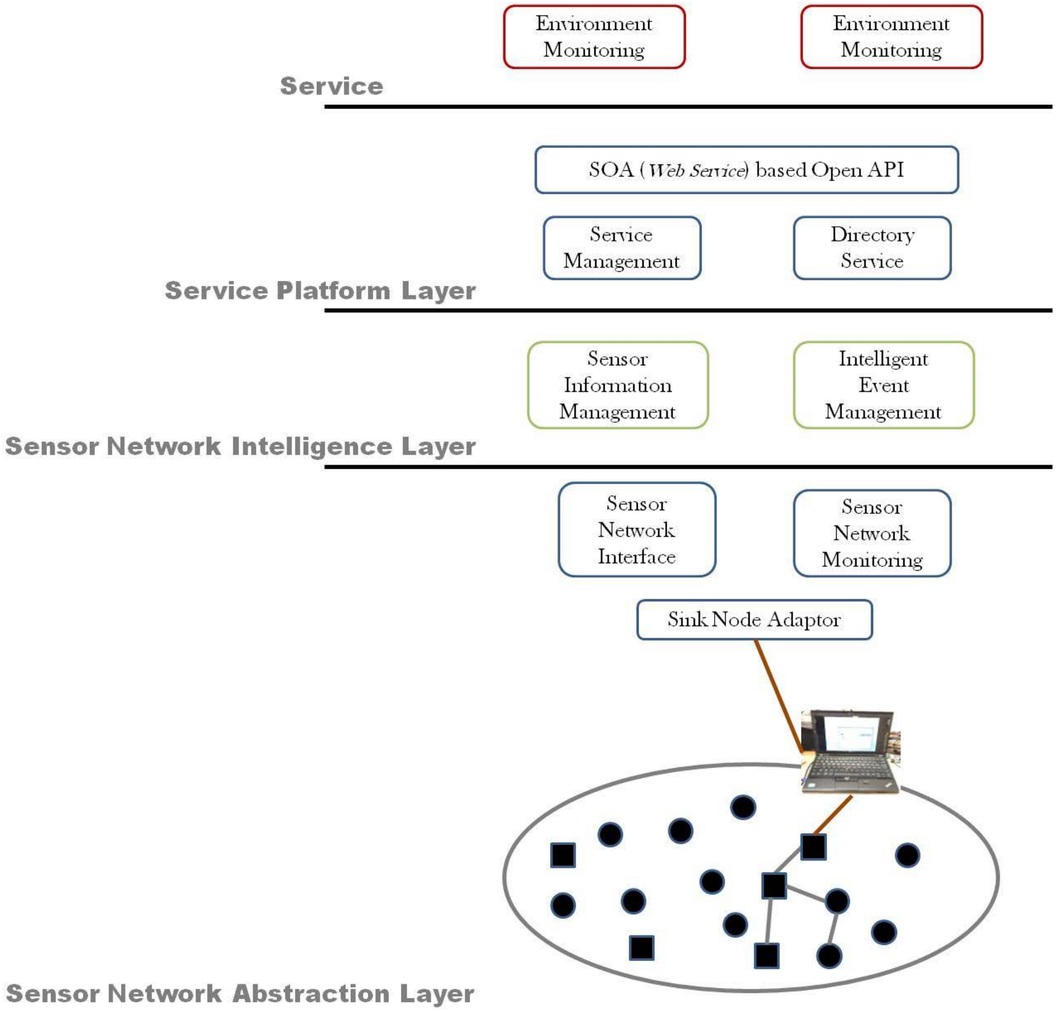


Figure A.15: Service Oriented architecture for WSN

The architecture depicted in Figure A.15 is split into three main layers:

- The Service Platform Layer which provides an application interface for various services such as the discovery of components or sensor network.
- Sensor Network Intelligence Layer which provides intelligent sensing data processing, intelligent event processing, and context information processing.
- Sensor Network Abstraction Layer which provides wireless information infrastructure abstraction and sensor network monitoring.

In [i.42] Avilés-López, Garcia-Macias and Antonio propose a service oriented architecture for WSNs they called TinySOA. This architecture is presented in Figure A.16.

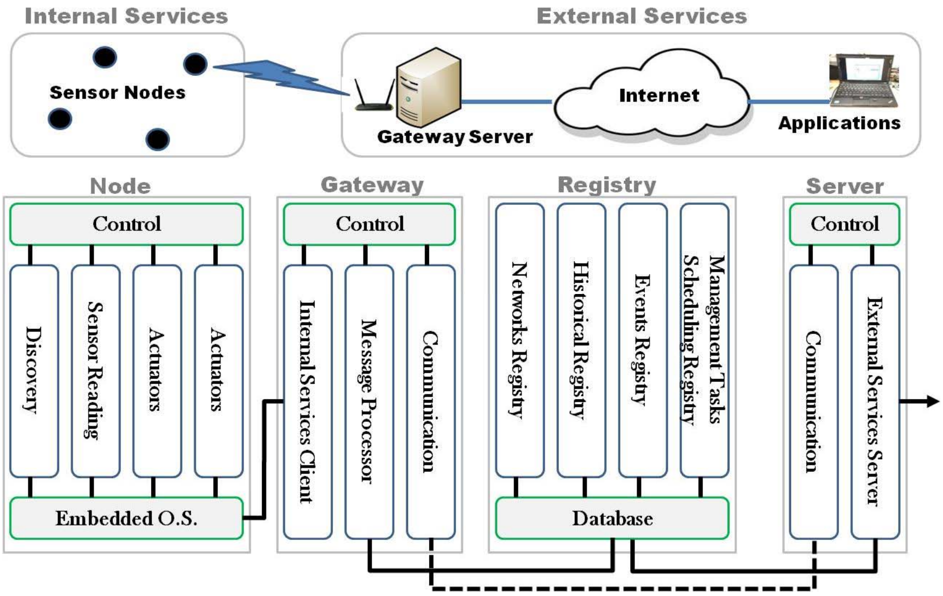


Figure A.16: TinySOA architecture

As depicted in Figure A.16, the Node encompasses all the functionality of sensing nodes and networks. The gateway receives, at startup, the registration messages with the single capabilities of each node. This gateway then aggregates the received node capabilities and deduces the services that can be provided by the full sensor network. All information about the infrastructure is stored in a registry. The server acts as a Web Services' provider, abstracting each available WSN as a separate web service. This provides an common interface for consulting services offered by each of the networks, for checking the registry and for registering/consulting events and maintenance tasks. Additionally, an information web service, in which clients can retrieve how many different network web services are available and how to access them, is set up.

TinyVisor was also implemented in [i.42] for testing purposes. At start up TinyVisor asks for the of a TinySOA server. This server URL have is provided for locating information and network web services. Once connected to the server, TyniVisor proceeds to open the information web service and sends a request for retrieving the number of currently available network web services. An interactive dialog box is then displayed. This dialog box shows the information related to the available WSNs including the name, description, and web service URL. It is then possible to select the network that is going to be used for monitoring and visualization. Once this network is selected, the information regarding its nodes and the corresponding sensed data can be visualized. This visualization is doable either in data mode graph mode, or topology mode. Figure A.17 presents the callable functions, from any program, through the web services interface to access WSNs.

Network Information
getNetworkId()
getNetworkName()
getNetworkDescription()
getTimeWindow()
getNodesList()
getActuatorsList()
getSensorsList()

Tasks Management
getTasksList(limit)
getTaskById(id)
addTask(type, value, target, time, recursive, event)
modifyTask(id, type, value, target, time, recursive, event)
deleteTask(id)

Events Management
getEventsList(limit)
getEventById(id)
addEvent(name, criteria)
modifyEvent(id, name, criteria)
deleteEvent(id)

Readings Management
getReadingsToTime(time, limit)
getReadings(from, to, sensor, limit)
getLastReadings(sensor)
getStatistics(statistic, from, to, sensor)

Figure A.17: TinySOA functions that can be called from any program via web services interface to access WSNs

In [i.43] Delicato et al. propose another Service Oriented Architecture for WSNs. This architecture is depicted in Figure A.18.

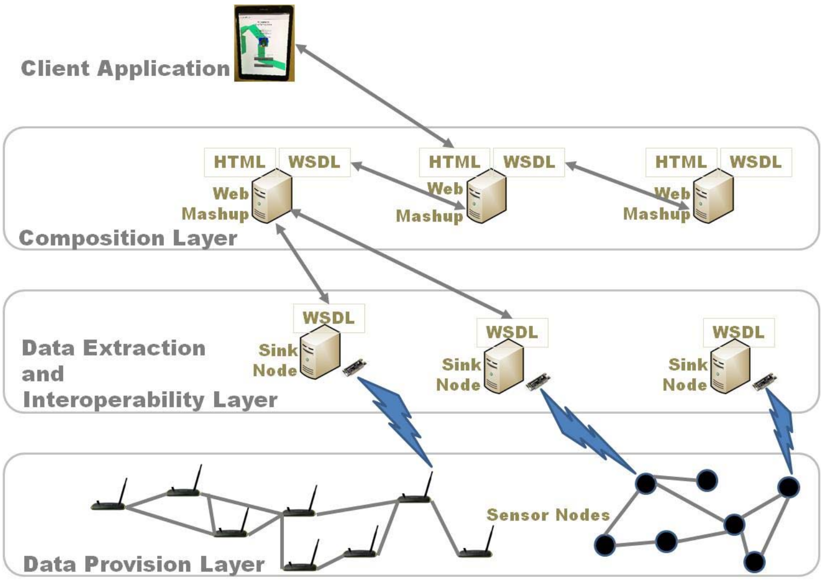


Figure A.18: Service Oriented Architecture for WSN proposed by F.C. Delicato et al.

As shown in Figure A.18, a sensor node can play the role of a sensing device, a router, and/or a data aggregator. The Sink node provides the collected information to other architectural components and can also play the role of a data aggregator. The Composition Layer presented in Figure A.14 consists of Web Mashups. This layer provides value-added services through the composition of data extracted from different WSNs using the sink node common interface. Such Mashups allow different levels of visualization and processing of a WSN. Web Mashups are published and discovered using the Mashup Catalogue that plays the role of service registry. This Mashup Catalogue can be implemented as a regular UDDI (Universal Description, Discovery, and Integration) register [i.44]. The communication between Sink nodes and the sensor network is accomplished using a WSN specific data dissemination protocol. The exchanged message are formatted using XML. HTTP binding is used to encapsulate the message exchange between the Sink and sensor nodes. Figure A.19 depicts the communication stack between sensor, sink and Mashups.

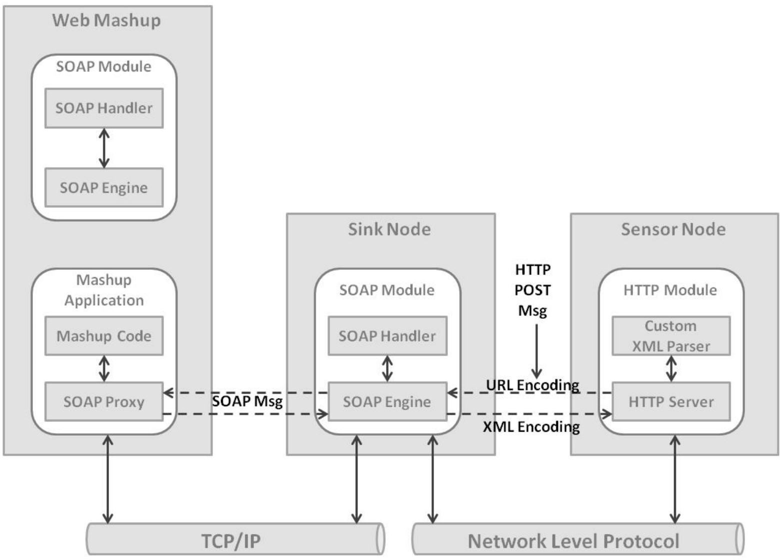


Figure A.19: Communication Stack proposed by F.C. Delicato et al.

In [i.44] Ghobakhlou et al. presents and XML-based architecture using the OGC SWE protocols and encodings. To overcome the overhead caused by XML exchange format, they proposed to interlink both OGC SWE semantics for sensor descriptions and observations (OGC Sensor Observation Service, SOS) and sensor configuration and planning (OGC Sensor Planning Service, SPS) with the open Message Queue Telemetry Transport (MQTT) protocol. A mapping of the complex and generic SWE semantics within MQTT implementation was developed.

In [i.45] Khedo et al. presented MiSense, a service oriented component based middleware architecture for WSN. MiSense aims to provide an abstraction layer between applications and network layer in WSN. it relies on the publish/subscribe service mechanism. MiSense involves four layers:

- The communication layer that is responsible for subscription and notification events.
- The resource management layer that manages the access control to the required resources. This access control is required for executing an application process.
- The common service layer which provides the common services for a WSN (data aggregation, topology management and routing).
- And the Domain Layer where specific services, like e.g. data fusion, are defined.

In [i.46] Singh et al. have suggested the use, for a WSN, of a flexible service oriented network architecture which supports service discovery, service abstraction, QoS and transparency, as well as interoperability among services. The main players in this architecture are:

- Service Provider: provides the service to application/users. The service provider can compose new services based on predefine, precompiled and pre-composed methods for matching requirements of Service User.
- Service Broker: matches the required service to a pre-defined existing services using protocol graph generator. It translates the messages to the adequate format and transmits it to the requestor.
- Service User: requests the service. It can be a parametric user which knows the technical details of the architecture, or a non-parametric user ignoring the details and requesting common services like monitoring.

A.1.4.2 Semantic SOA for WSN

In [i.47] Amato et al. presented an innovative architecture for risk management that relies on services. The proposed architecture allows the collection and the aggregation of raw data in order to create end-users application using web services. Figure A.20 depicts the general view of the proposed architecture.

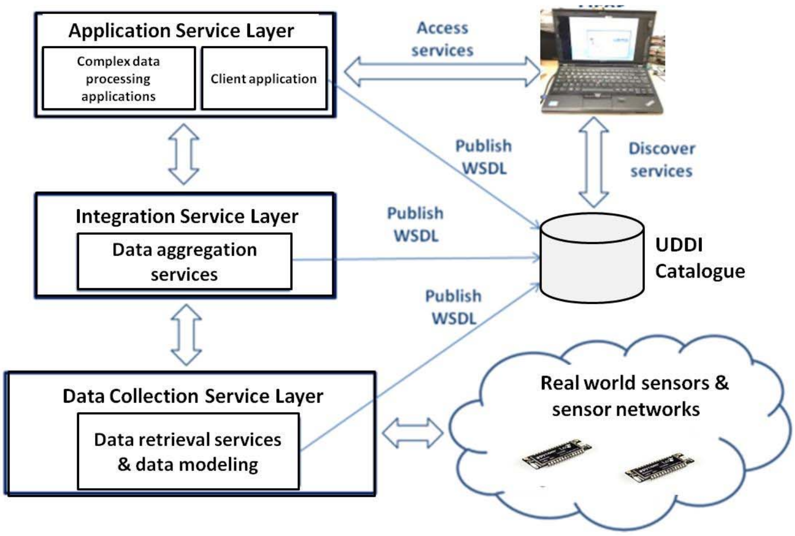


Figure A.20: Semantic SOA Overview Architecture proposed by F. Amato et al.

As shown in Figure A.20, the Data Collection Service Layer interacts directly with the WSN for collecting the raw data. The Data Collection Service Layer then enriches these raw data with ontological modelling techniques. This layer thus offers the service of converting data into XML/RDF data model. The Integration Service Layer aggregates and clusterizes data according to a defined data model. Then, this Integration Service layer delivers it to the Application Service Layer. Finally, the application service layer invokes the service which is accessible through WSDL standard.

Each service has its own security policy and it is published in a public registry. To integrate data from heterogeneous source, the wrapper-mediator paradigm has been used. Each wrapper explores and monitors the local sensor network and sends to the mediator an appropriate description of the related information according to a common data model. Figure A.21 summarizes the wrapper-mediator functionality within the Collection and Integration Services.

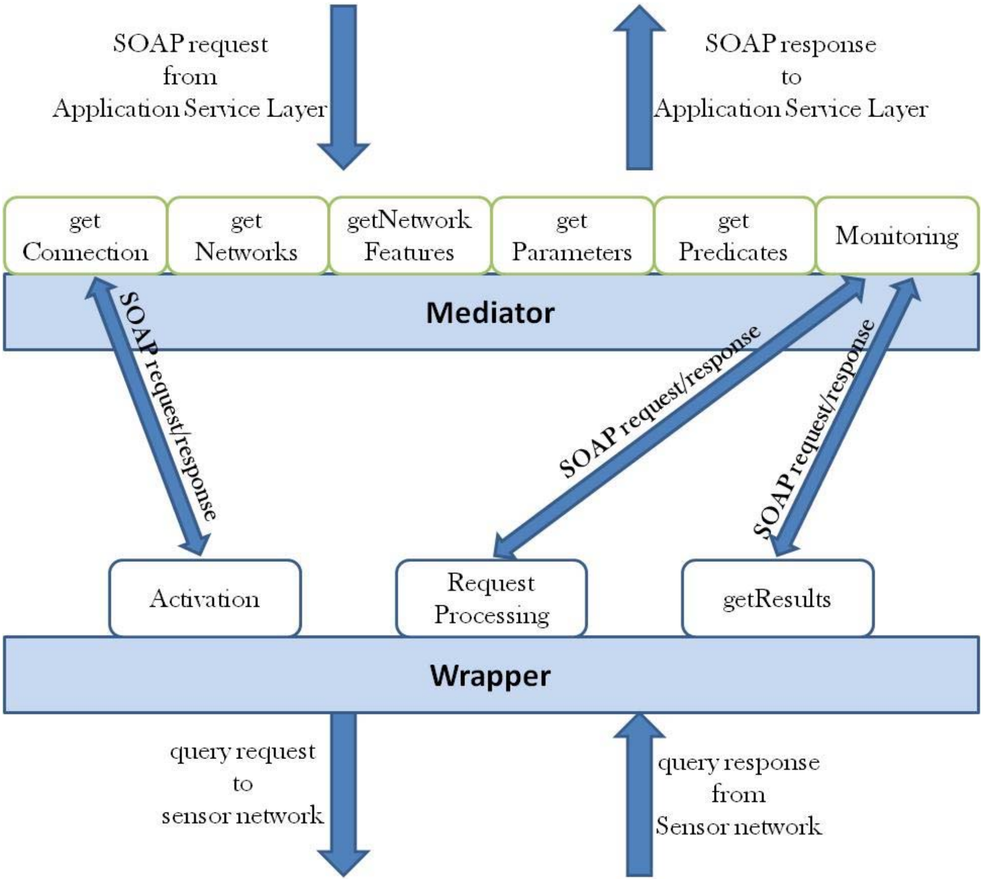


Figure A.21: Collection and Integration Services

For the discovery and the registration of a specific wrapper, the getConnection function is used. It then activates the service provided by the wrapper module. The getNetworks gives as output the list of all networks connected to the system, their topology, and basic information such as network identifiers, number of sensors and maximum depth for each network. The getNetworkFeatures takes as input parameter the network identifier and returns: the type of system, the middleware, the number of sensors, the maximum depth, the IP address, the registration time, the communication ports, as well as information about the base station (frequency, communication link). The getParameters returns the sensor parameters, like the IDs (i.e. network, cluster and sensor identifiers), the free memory, the voltage and the channel quality. The getPredicates gives all the physical variable measured by a sensor (identified by its sensor identifier). Finally, the monitoring task queries a sensor, or a sensor network, invokes both the requestProcessing and the getResults services for getting the results related to a specific query.

In conclusion, to add semantic meanings to the sensed data, the sensors add-on (data aggregator) retrieves samples from sensor systems and converts them into an XML format, according to defined translation rules. The XML-RDF Wrapper translates the file into an RDF document, according to a set of sensor domain ontologies. Finally, the O&M Data modeler retrieves an RDF document and converts it in an O&M standard XML document.

A.1.4.3 Open Sensor Web Architecture (OSWA)

OSWA [i.48] is an extension of SWE proposed by National ICT Australia Ltd (NICTA) University of Melbourne. Its main aim is to combine SOA with SWE. OSWA defines four different layers as shown in Figure A.22.

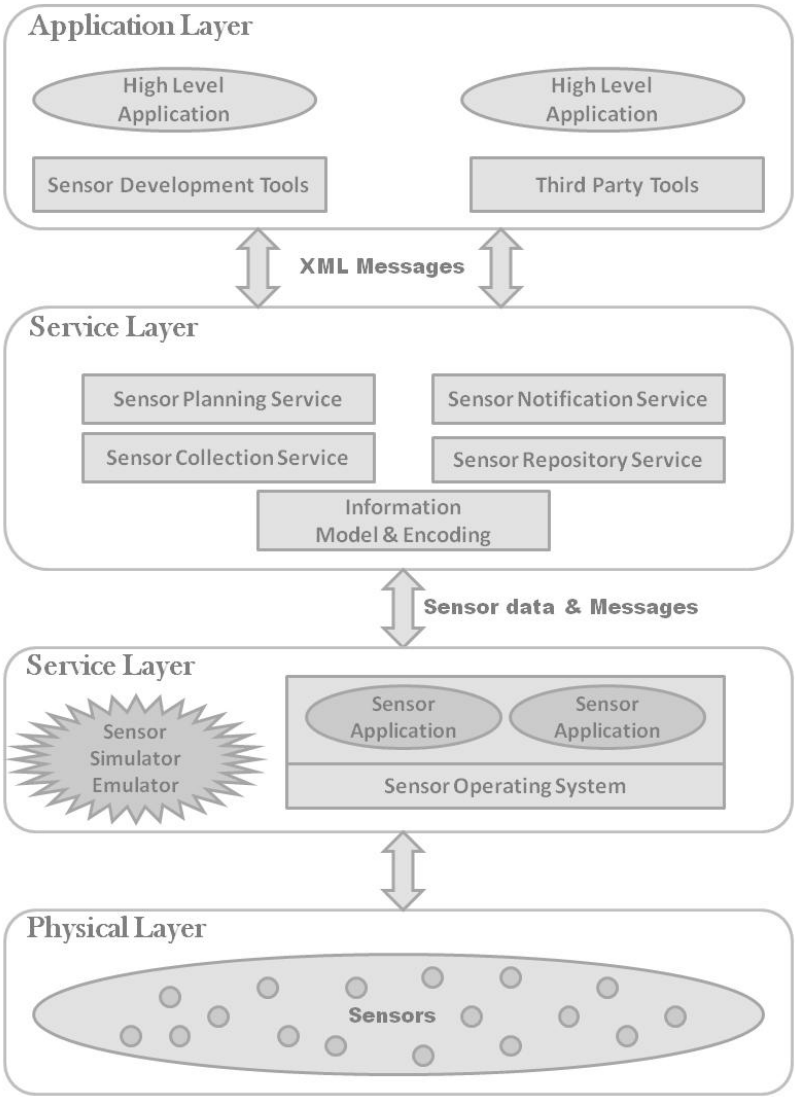


Figure A.22: OSWA layers

As shown in Figure A.22, the OSWA service layer reuses the services described in SWE and specifies how these services can be implemented in real scenarios. It uses WSDL and UDDI for service discovery. SCS provides interface to both streaming data and query based sensor applications that are built on top of TinyOS and TinyDB respectively. Only two features of SPS were implemented:

- getFeasibility; and
- submitRequest.

The WNS offered two features: registerUser and doNotification. For user registration a JDBC-based UserAccountManager has been implemented. Moreover, the email protocol is used to notify clients. Sensor Repository Service is deployed as a web service that can be accessed via SOAP messages.

A.1.4.4 W3C Web of Things (WoT) ontology: Thing Description

W3C [i.59] is in particular working on the Web of Thing and is actually specifying an formalizing a dedicated semantic reference model and associated ontology called TD ontology (Thing Description). This ontology, a key building block of a global WoT environment, is still under design, which means that all the concepts around a Thing and WoT environment are not yet fixed and may be subject to major revisions including concepts removal. Figure A.23 depicts the current High Level view of the W3C TD ontology.

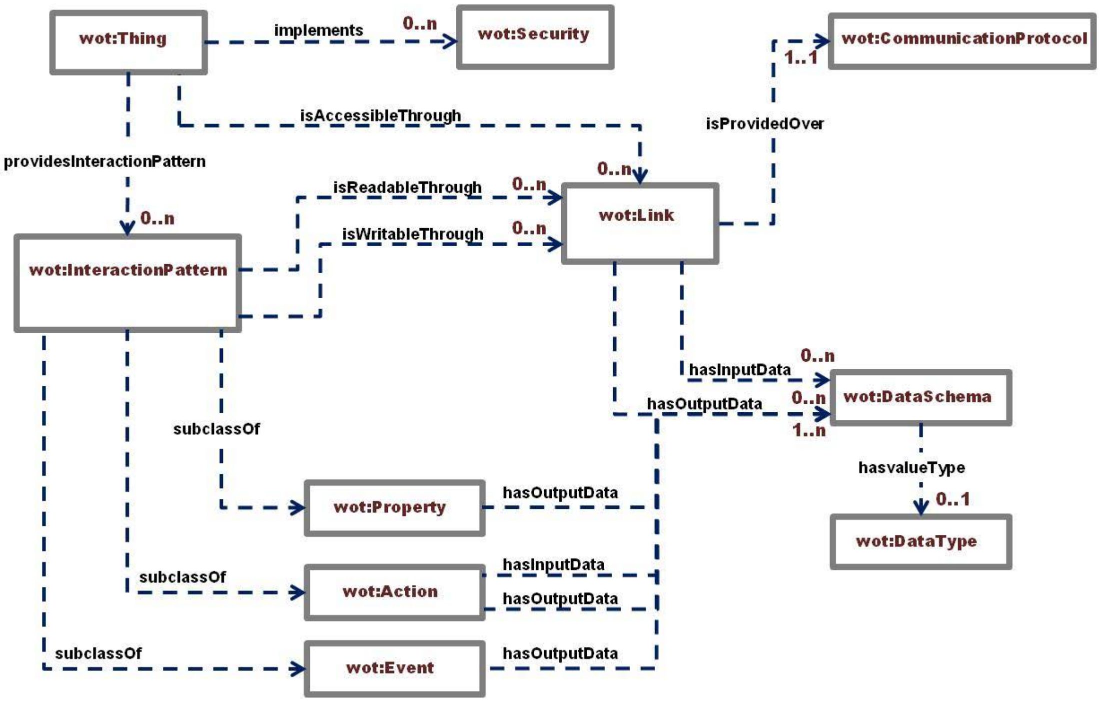


Figure A.23: Current High Level view of W3C TD ontology

As shown in , the main concept (class) introduced for DT ontology is *Thing*, i.e. the virtual description of a physical object/thing. Within DT model, a Thing is described via the following concepts, properties and attributes (see Figure A.23):

- Thing. A Think has a name (String) and a baseURI (an URI as unic ID). A Think implements security mecanisms that are described through a Security class (actually an empty generic container template that should model any security paradigms that a thing has to follow/implement). A Thing is accessible through a *Link* that materializes the possible interactions with a Thing, mainly in terms of interfaces/media types , those media types being carried out over dedicated communication protocols (TD *CommunicationProtocol* class introduced in Figure A.23). Finally, a Thing provides interactionPattern which describes the functionalities a Thing is providing and the way to trigger them,

- InteractionPattern describes the functionalities a Thing is providing. Those functionalities are the following: get or set a properties (TD *Property* class introduced in Figure A.23), execute an function, i.e. an action (TD *Action* class introduced in Figure A.23), as well as functions related to eventing management on that thing and some of its properties (TD *Event* class introduced in Figure A.23).

- DataSchema.

- DataType.

A.1.5 Ontology validation methods

A.1.5.0 Introduction

In clause A.1.5 of the present document, an overview of existing ontology validation methods is presented. Those models will be investigated here with the objective of selecting the most appropriate one for qualifying the Smart BAN retained and proposed unified ontology.

For qualifying and validating an ontology, three approaches have already been defined:

- Evolution-based validation.
- Symbolic-based validation.
- Attribute-based validation.

The first aforementioned validation method consists in observing the evolution of the ontology usage, over the time. The original ontology schema is a posteriori compared to all the instances of that ontology that was introduced and used during a given period of time. The retained evaluation criteria are, for example:

- Ontology domain changes. It is introduced for tracking any new knowledge that could have been added to the domain formalized by the ontology.
- Changes in the ontology usage perspectives in a given domain. It is introduced for tracking any changes in ontology usage impacting the ontology conceptualization.
- Ontology specification changes. It is introduced for tracking ontology stability by detection of the number of new objects or attributes that have been introduced in the original ontology graph.

In the context of Smart BANs, it is clear that this validation method cannot be used alone since it is mandatory to validate the specified ontology before its usage in eHealth/medical applications. This validation method has therefore to be combined with ether a symbolic based and/or an attribute based approach.

The second validation approach is the symbolic one. It is a rule-based evaluation where two rule sets are defined: one containing rules directly coming from the ontology itself and the other containing rules coming from the user domain. The first set encompasses rules like dissimilarities, matching, inclusion or inheritance between objects and or attributes. The second set incorporates rules coming from the user domain for identifying conflicting properties in the ontology specifications.

Finally, the third approach consists in scanning the ontology graph for measuring the knowledge level through the use of quantitative metrics like, e.g. location and distribution of the ontology objects in the graph.

All the emerging ontology frameworks for each of the aforementioned approaches will be identified, introduced and evaluated in the scope of TC SmartBAN.

A.1.5.1 Common used symbolic-based method

While ontologies are used to infer implicit knowledge, the evaluation approach based on rules is essential. Basically, this approach checks for consistency which means that the ontology does not contain any contradictory facts. From literature [i.39] three consistency's types can be distinguished:

- Structural consistency: to check that the ontology respects the rules underlying the ontology language. In OWL language, it means checking for TBox rules. It means to verify the classes' structures like subclasses, disjoint classes and equivalent classes. For example, a subclass of a class should be a class.
- Logical consistency: to validate that the ontology conforms the logical theory of the ontology. These rules are describes in Abox rules. They allow the verification of the data properties and object properties constraints. The domain and ranges are checked in addition to the values, the equality between properties and the inequality between properties. Moreover, the logical consistency assures that a functional property should be unique in the ontology model. The inverse properties, reflexive, irreflexive, symmetric and asymmetric properties are tested in this level.

- User-defined: These rules are added by the users. Some inferences can be gathered from the ontology model itself, but others may not be expressible in the ontology language (usually OWL) and require a more functional representation. Here comes the role of the Semantic Web Rule Language SWRL [i.11].
- As part of the owl rules, the user can determine the maximum cardinality and minimum cardinality for the object properties.

After consistency checking, the satisfiability (i.e. if it is possible for a class to have instances) is tested. Then, the classification process is executed in order to create the complete class hierarchy. Finally, the realization is executed where most specific classes that individuals belongs to are shown.

A.1.5.2 Common used attribute-based methods

A wide variety of metrics has been proposed in order to evaluate the ontologies in terms of complexity and completeness and other metrics were proposed in order to compare between different ontologies and rank it.

OntoClean [i.49] is a framework that helps developers to evaluate their ontologies in order to check for inconsistency and conflicts. The developer should add meta-property (Essence, Rigidity, Identity & Unit) for the predefined properties. A property is tagged as "Essence" if it is essential to it. A special form of essentiality is rigidity. In this case, the property is essential for all entities. Thus, a property can vary from being rigid, semi-rigid to anti-rigid (i.e. which means not essential to all properties). The other type of meta-property is used to determine the equality between the entities. This is done by indicating the identity of the property. In addition, the developer can designate the unity of a property (if it is a unit property or depends of other properties). The unity will help in checking the consistency of the ontology. Ontoclean can be used with Protégé. However, OntoClean is considered complex for a developer especially if he is dealing with large ontologies where he has to add meta-property for many properties.

The OntoQA tool developed by LSDIS Lab at the University of Georgia is a simpler analysis method that includes a number of metrics that includes individuals, thus it allows for the automatic measurement of ontologies. In OntoQA [i.39], metrics (features) are divided into two groups: schema metrics, which address the design of the ontology schema, and instance metrics, that address the way instances are organized within the ontology. Although the ontology design accuracy of modelling the domain knowledge cannot be definitely known, metrics in this category indicate the richness, width, depth, and inheritance of an ontology schema design. This is done by calculating:

- The relationship richness. It measures the number of the properties in the selected class that are actually being used in the instance ontology relative to the number of relationships defined for the selected class in the ontology definition.
- Inheritance richness. It shows the number of inheritance relationship from the total number of relations.
- And attribute richness. It indicates the distribution of instance over classes.

The way data is placed within ontology is also a very important measure of ontology quality because it can indicate the effectiveness of the ontology design and the amount of real-world knowledge represented by the ontology. Instance metrics include metrics that describe the KB (Knowledge Base) as a whole, and metrics that describe the way each schema class is being utilized in the KB. These metrics are the class richness, class connectivity that indicates the number of instances of other classes connected by relationships to instances of the selected class, class importance and the cohesion.

Contrary to the previous methods that are used only by developer, OntoCAT ontology consumer analysis [i.50] is proposed to be used by ontology developer and user for evaluation purposes. It is composed of two metrics categories: intentional metrics and extensional metrics. Intentional metrics are calculated based on the ontology definition itself, i.e. its classes and subclasses and their properties. Extensional metrics deal with the instances and how effectively the ontology is including the domain knowledge. In other terms, OntoQA schema metrics match OntoCAT intentional metrics, and OntoQA instance metrics match OntoCAT extensional metrics calculated on the ontology occurrences [i.50]. OntoCAT define the following groups of metrics:

- Intentional size metrics: show the numbers of classes, relationships and properties in a tree or sub-tree.
- Extensional size metrics: show the number of occurrences of classes, relationships and properties.
- Intentional structural metrics: show structural metrics are similar to size metrics, since they are over the entire intentional ontology, i.e. over all the root trees defined in the ontology (if no concept or class is specified).

- Extensional structural metrics: same as the previous metrics but on occurrences.
- Summarization metrics: select the object occurrences (for extensional) and classes (for intentional) having the maximum number of links in handout. For intentional, the count of links is the number of subclasses and superclasses defined. For extensional, the links are based on the relationships specified for creating its structure.

OntoCAT can be used during the life cycle of the ontology due to the use of extensional metrics that shows the evaluation of the ontology after individuals' creation.

OntoMetric [i.51] is designed to help the users in choosing the appropriate ontology. It checks the content, the language, the developed methodology, the building tools and the usage cost to decide which ontology is suitable for a certain domain of application. The user should choose a reference ontology, and based on the analytic hierarchy process, it decides weather or not to reuse the ontology.

In conclusion, where some evaluation methods were designed to help the users in selecting the adequate ontology, others focus on helping the developers to identify the complexity of their ontologies, the core of their ontologies and the consistency. In fact, while the users look for accuracy, adaptability and extensibility, clarity, and organizational fitness in certain domain, the developers should provide conciseness (no irrelevant elements or redundant representation of semantics), lightness (minimum allocation resources in terms of memory and processing), consistency (no contradictions) and flexibility to be applied in different applications. Thus, the ontology should be tested during the development life cycle, evaluated by users during its usage and ranked comparing to other ontologies covering the same domain. A combination of the previous mentioned methodologies and maybe new methodologies can be used to evaluate an ontology.

A.1.6 Preliminary conclusion concerning existing sensor ontologies

Lots of ontologies were proposed to manage WSN heterogeneity focusing on different application domains or operation management. OGC proposed different standards to describe the sensor nodes, the data and the observation process (SensorML and O&M Observation). However, the description is not based on semantic ontology and is therefore not allowing similarity detection or conflict resolution that have in particular to be handled within SmartBANs. W3C incubator has proposed the SSN ontology which describes the sensor nodes, data and observation process. WSSN has next extended SSN by adding the communication process, the data streaming and the state to the ontology specifications. Faculty of Technical Sciences, University of Novi Sad focused on solving the hardware heterogeneity in wireless sensor network. It proposed an ontology that describes the component, the sensor nodes, the data, the communication process and Energy component of the sensor. However, to effectively manage a wireless sensor network, more components should be added for aggregation function, data fusion, routing, error detection and others.

On the eHealth side, lots of standards for healthcare message exchange and terminology have been developed such as CEN/TC 251 [i.28], ISO/IEEE 11073 [i.30], ASTM E31 [i.33] or Bluetooth profiles for medical devices. CEN/TC 251 (EN 13606 [i.29] standard) defines the high-level logical models for any kind of EHR and hereby enables syntactic interoperability. However, this only guaranties common data structure exchange procedures but does not ensure that the data meaning will be interpreted identically by all parties. Regarding ISO/IEEE 11073 [i.30], it defines the data format, in addition to the data storage and data transmission, for eHealth environments. ASTM E31 short analysis shows that this standard is the closest to an EHR object model. However, it is not so extensible and still lacks a lot of essential information. Finally, Bluetooth LE alleviates the interoperability between medical devices by presenting specific data formats and units, in addition to the notification systems and communication between different medical parties. Unfortunately it is only dedicated for Bluetooth enabled devices.

In conclusion, none of the existing ontologies, as well as no combination of the existing ontologies, is able to provide a complete level of reasoning over medical data stored in multiple locations. That's confirms the need of a specification and qualification of a unified extended data representation model and transfer format in order to enhance the management of EHR systems and make Smart BANs a reality. This SmartBAN open data model is specified in clause 6 of the present document.

Annex B (normative): OWL-DL in Turtle formalization of the SmartBAN ontology

B.1 quantity.ttl

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1#> .
@prefix dcterms: <http://purl.org/dc/terms#> .
@prefix vann: <http://purl.org/vocab/vann#> .
@prefix quantity: https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/quantity#> .
@prefix qudt: <http://qudt.org/schema/qudt#> .
@base https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/quantity#> .

https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/quantity#> rdf:type owl:Ontology ;
                                          dcterms:description "This ontology defines th quantities of
the SmartBAN ontology and its qauntity kind refering to QUDT ontology."@en ;
                                          dcterms:creator <https://www.etsi.org/technologies/smart-
body-area-networks> ;
                                          dcterms:contributor [a foaf:Person ; foaf:name "Lina
Nachabe" ] ;
                                           dcterms:contributor [a foaf:Person ; foaf:name "Aya Saïd"
] ;
                                             dcterms:license <https://opensource.org/license/mit/> ;
                                             dcterms:created "2024-12-16"^^xsd:date ;
                                             dcterms:modified "2024-12-16"^^xsd:date;
                                             dcterms:issued "2025-01-20"^^xsd:date;
                                             dcterms:rights "Copyright © ETSI smart body area
netwoks"@en ;
                                             dcterms:title "SmartBAN Quantity Ontology"@en ;
                                             vann:preferredNamespacePrefix "quantity" ;
                                             vann:preferredNamespaceUri
https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/quantity#> ;
                                             rdfs:label "Quantity"@en ;
                                             owl:versionInfo "2.0".

#################################################################
#    Annotation Properties
#################################################################

###  http://purl.org/dc/terms/contributor
dcterms:contributor rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/creator
dcterms:creator rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/description
dcterms:description rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/issued
dcterms:issued rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/license
dcterms:license rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/modified
dcterms:modified rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/title
dcterms:title rdf:type owl:AnnotationProperty .
```

```
###  http://purl.org/vocab/vann/preferredNamespacePrefix
vann:preferredNamespacePrefix rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespaceUri
vann:preferredNamespaceUri rdf:type owl:AnnotationProperty .

#################################################################
#    Object Properties
#################################################################

###  hasQuantityKind
quantity:hasQuantityKind rdf:type owl:ObjectProperty ;
           rdfs:domain quantity:Quantity ;
           rdfs:comment "Links a quantity to its quantity kind." ;
           rdfs:range quantity:QuantityKind;
           rdfs:label "Has Quantity Kind"@en .

###  hasUnit
quantity:hasUnit rdf:type owl:ObjectProperty ;
           rdfs:domain quantity:Quantity ;
           rdfs:domain quantity:Unit ;
           rdfs:comment "Associates a quantity to its unit of measurement." ;
           rdfs:label "Has Unit"@en .

#################################################################
#    Data properties
#################################################################

###  hasValue
quantity:hasValue rdf:type owl:DatatypeProperty ;
                 rdfs:domain quantity:Quantity;
                 rdfs:comment "Specifies the current value of a quantity" ;
                 rdfs:label "Has Value"@en .
#################################################################
#    Classes
#################################################################

###  Quantity
quantity:Quantity rdf:type owl:Class ;
          rdfs:subClassOf qudt:Quantity ;
          rdfs:comment "A quantity is the measurement of an observable property of a particular 
object, event, or physical systems." ;
          rdfs:label "Quantity"@en .

###  QuantityKind
quantity:QuantityKind rdf:type owl:Class ;
          rdfs:subClassOf qudt:QuantityKind;
          rdfs:comment "A Quantity Kind is any observable property that can be measured and 
quantified numerically. Familiar examples include length, temperature..." ;
          rdfs:label "Quantity Kind"@en .

###  Unit
quantity:Unit rdf:type owl:Class ;
          rdfs:comment "A unit of measure, or unit, is a particular quantity value that has been 
chosen as a scale for measuring other quantities the same kind (more generally of equivalent 
dimension).. Familiar examples include length, temperature..." ;
          rdfs:subClassOf qudt:Unit;
          rdfs:label "Unit"@en .
```

B.2 location.ttl

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1#> .
@prefix dcterms: <http://purl.org/dc/terms#> .
@prefix vann: <http://purl.org/vocab/vann#> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>.
@prefix location: <https://w3id.org/italia/onto/CLV/Address#>.
```

```
@prefix loc: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/location#> .
@base <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/location#> .

<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/location#> rdf:type owl:Ontology ;
                                         dcterms:description "This ontology defines the location of
the nodes in the SmartBAN."@en ;
                                         dcterms:creator <https://www.etsi.org/technologies/smart-
body-area-networks> ;
                                         dcterms:contributor [a foaf:Person ; foaf:name "Lina
Nachabe" ] ;
                                          dcterms:contributor [a foaf:Person ; foaf:name "Aya Saïd"
] ;
                                             dcterms:license <https://opensource.org/license/mit/> ;
                                             dcterms:created "2024-12-16"^^xsd:date ;
                                             dcterms:modified "2024-12-16"^^xsd:date;
                                             dcterms:issued "2025-01-20"^^xsd:date;
                                             dcterms:rights "Copyright © ETSI smart body area
netwoks"@en ;
                                             dcterms:title "SmartBAN Location Ontology"@en ;
                                             vann:preferredNamespacePrefix "loc" ;
                                             vann:preferredNamespaceUri
<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/location#> ;
                                             rdfs:label "Location"@en ;
                                             owl:versionInfo "2.0".

#################################################################
#    Annotation Properties
#################################################################
###  http://purl.org/dc/terms/contributor
dcterms:contributor rdf:type owl:AnnotationProperty .
###  http://purl.org/dc/terms/creator
dcterms:creator rdf:type owl:AnnotationProperty .
###  http://purl.org/dc/terms/description
dcterms:description rdf:type owl:AnnotationProperty .
###  http://purl.org/dc/terms/issued
dcterms:issued rdf:type owl:AnnotationProperty
###  http://purl.org/dc/terms/license
dcterms:license rdf:type owl:AnnotationProperty
###  http://purl.org/dc/terms/modified
dcterms:modified rdf:type owl:AnnotationProperty .
###  http://purl.org/dc/terms/title
dcterms:title rdf:type owl:AnnotationProperty .
###  http://purl.org/vocab/vann/preferredNamespacePrefix
vann:preferredNamespacePrefix rdf:type owl:AnnotationProperty .
###  http://purl.org/vocab/vann/preferredNamespaceUri
vann:preferredNamespaceUri rdf:type owl:AnnotationProperty .
#################################################################
#    Classes
#################################################################
###  Address
loc:Address rdf:type owl:Class ;
             rdfs:subClassOf loc:Location, location:Address ;
             rdfs:comment "Represents the surface area or external anatomical regions of the human
body within the context of the SmartBAN. It encompasses areas where wearable or implantable sensors
may be attached or deployed for physiological monitoring or healthcare applications." ;
             rdfs:label "Body Surface"@en .
###  BodySurface
loc:BodySurface rdf:type owl:Class ;
             rdfs:subClassOf loc:Location ;
             rdfs:comment "Represents the surface area or external anatomical regions of the human
body within the context of the SmartBAN. It encompasses areas where wearable or implantable sensors
may be attached or deployed for physiological monitoring or healthcare applications." ;
             rdfs:label "Body Surface"@en .
###  External
loc:External rdf:type owl:Class ;
          rdfs:subClassOf loc:Location;
          rdfs:comment "Denotes locations or spatial regions external to the human body within the
BAN environment. It includes physical spaces, ambient surroundings, or environmental contexts where
devices operate or interact with users or other entities." ;
          rdfs:label "External"@en .
###  Implant
loc:Implant rdf:type owl:Class ;
         rdfs:subClassOf loc:Location ;
```

```
        rdfs:comment "Represents devices or sensors that are surgically implanted or embedded
within the human body for monitoring physiological parameters or delivering therapeutic
interventions. Implants facilitate continuous health monitoring or treatment within the BAN." ;
        rdfs:label "Implant"@en .
###  Location
loc:Location rdf:type owl:Class ;
         rdfs:comment "Represents a spatial location or geographical coordinates within the BAN.
Locations enable tracking, mapping, and contextualization of devices and events in physical space,
supporting location-based services and situational awareness." ;
         rdfs:label "Location"@en .
###  LocationCoordinates
loc:LocationCoordinates rdf:type owl:Class ;
                    rdfs:subClassOf loc:External, geo:SpatialThing;
                    rdfs:comment "Refers to geographical coordinates or spatial positions within
the BAN environment. It provides precise location information using latitude, longitude, and
altitude coordinates, enabling accurate tracking, navigation, or spatial analysis." ;
                    rdfs:label "Location coordinates"@en .
```

B.3 wban.ttl

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1#> .
@prefix dcterms: <http://purl.org/dc/terms#> .
@prefix vann: <http://purl.org/vocab/vann#> .
@prefix wban: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/wban#> .
@prefix loc: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/location#> .
@prefix quantity: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/quantity#> .
@base <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/wban#> .

<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/wban#>  rdf:type owl:Ontology ;
 owl:versionIRI <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/wban#> ;
                                             dcterms:description "A WBAN has a communication
process, contains nodes and has a contact. The contact is a foaf Agent wich can be a person or
organisation. The contact plays a role of online resource, responsible party or health actor. The
health actor uses the WBAN."@en ;
                                             dcterms:creator
<https://www.etsi.org/technologies/smart-body-area-networks> ;
                                             dcterms:contributor [a foaf:Person ; foaf:name "Lina
Nachabe" ] ;
                                             dcterms:contributor [a foaf:Person ; foaf:name "Aya
Saïd" ] ;
                                             dcterms:license <https://opensource.org/license/mit/> ;
                                             dcterms:created "2024-12-16"^^xsd:date ;
                                             dcterms:modified "2024-12-16"^^xsd:date;
                                             dcterms:issued "2025-01-20"^^xsd:date;
                                             dcterms:rights "Copyright © ETSI smart body area
netwoks"@en ;
                                             dcterms:title "SmartBAN Wireless Body Area Network
Ontology"@en ;
                                             vann:preferredNamespacePrefix "wban" ;
                                             vann:preferredNamespaceUri
<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/wban#> ;
                                             rdfs:label "Wireless Body Area Network"@en ;
                                             owl:versionInfo "2.0".
dcterms:issued a owl:AnnotationProperty .
dcterms:modified a owl:AnnotationProperty .
dcterms:source a owl:AnnotationProperty .
dcterms:creator a owl:AnnotationProperty .
dcterms:title a owl:AnnotationProperty .
dcterms:description a owl:AnnotationProperty .
dcterms:license a owl:AnnotationProperty .
dcterms:publisher a owl:AnnotationProperty .
vann:preferredNamespacePrefix a owl:AnnotationProperty .
vann:preferredNamespaceUri a owl:AnnotationProperty .
```

```
#################################################################
#    Object Properties
#################################################################

### hasCommunicationProcess
wban:hasCommunicationProcess rdf:type owl:ObjectProperty ;
                         rdfs:domain wban:WBAN ;
                         rdfs:range wban:CommunicationProcess ;
                         rdfs:comment "Links a WBAN (Wireless Body Area Network) to its
communication process, indicating how the network handles data communication (e.g. periodic, event-
driven)." ;
                         rdfs:label "has Communication Process"@en .

### isAssociatedTo
wban:isAssociatedTo rdf:type owl:ObjectProperty ;
                         rdfs:domain wban:Role ;
                         rdfs:range wban:WBAN ;
                         rdfs:comment "Links a Role to the corresponding WBAN." ;
                         rdfs:label "is Associated To"@en .


###  hasContact
wban:hasContact rdf:type owl:ObjectProperty ;
            rdfs:domain wban:WBAN ;
            rdfs:range wban:Contact ;
            rdfs:comment "Associates a WBAN with a contact entity, which can be a person,
responsible party, or online resource involved in the network's management or usage." ;
            rdfs:label "has Contact"@en .


### hasWbanLocation
wban:hasWbanLocation rdf:type owl:ObjectProperty ;
              rdfs:comment "Links the WBAN to its location. The location is described in Location
ontology" ;
              rdfs:domain wban:WBAN ;
              rdfs:range loc:Location ;
              rdfs:label "Has Wban Location"@en .


###  playsRole
wban:playsRole rdf:type owl:ObjectProperty ;
            rdfs:domain wban:Contact ;
            rdfs:range wban:Role ;
            rdfs:comment "Specifies the role of a contact." ;
            rdfs:label "plays Role"@en .


### uses
wban:uses rdf:type owl:ObjectProperty ;
            rdfs:domain wban:Patient ;
            rdfs:range wban:WBAN ;
            rdfs:comment "Specifies the WBAN used by a Patient." ;
            rdfs:label "uses"@en .

### isUsedBy
wban:isUsedBy rdf:type owl:ObjectProperty ;
             rdfs:inverseOf wban:uses;
            rdfs:domain wban:WBAN ;
            rdfs:range wban:Patient ;
            rdfs:comment "Specifies the  Patient using the WBAN" ;
            rdfs:label "Is Used By"@en .

### Patients properties-------------------------------------------------------------------
### Height
wban:hasHeight rdf:type owl:ObjectProperty ;
            rdfs:domain wban:Patient ;
            rdfs:range quantity:Quantity ;
            rdfs:comment "Specifies the  Patient's height" ;
            rdfs:label "has Height"@en .
### hasWeight
wban:hasWheight rdf:type owl:ObjectProperty ;
            rdfs:domain wban:Patient ;
            rdfs:range quantity:Quantity ;
            rdfs:comment "Specifies the  Patient's Weight" ;
            rdfs:label "has Weight"@en .
### MaxHR
wban:hasMaxHeartRate rdf:type owl:ObjectProperty ;
            rdfs:domain wban:Patient ;
```

```
            rdfs:range quantity:Quantity ;
            rdfs:comment "Specifies the  maximum heart rate of a Patient" ;
            rdfs:label "Has Maximum Heart Rate"@en .

### RestingHR
wban:hasRestingHeartRate rdf:type owl:ObjectProperty ;
            rdfs:domain wban:Patient ;
            rdfs:range quantity:Quantity ;
            rdfs:comment "Specifies the  resting heart rate of a Patient" ;
            rdfs:label "Has Resting Heart Rate"@en .

### FatBurnHR
wban:hasFatBurnHeartRate rdf:type owl:ObjectProperty ;
            rdfs:domain wban:Patient ;
            rdfs:range quantity:Quantity ;
            rdfs:comment "Specifies the  fat burn heart rate of a Patient" ;
            rdfs:label "Has Fat Burn Heart Rate"@en .

### HipCircumference
wban:hasHipCircumference rdf:type owl:ObjectProperty ;
            rdfs:domain wban:Patient ;
            rdfs:range quantity:Quantity ;
            rdfs:comment "Specifies the hip circimference of a Patient" ;
            rdfs:label "Has Hip Circumference"@en .

### WaistCircumference
wban:hasWaistCircumference rdf:type owl:ObjectProperty ;
            rdfs:domain wban:Patient ;
            rdfs:range quantity:Quantity ;
            rdfs:comment "Specifies the waist circimference of a Patient" ;
            rdfs:label "Has Waist Circumference"@en .

#################################################################
#    Data properties
#################################################################

###  ApplicationDomain
wban:ApplicationDomain rdf:type owl:DatatypeProperty ;
                   rdfs:domain wban:WBAN ;
                   rdfs:range xsd:string ;
                   rdfs:comment "Defines the application domain of a WBAN, specifying the field or
area where the WBAN is applied, such as healthcare, sports, or military. The value is a string." ;
                   rdfs:label "Application Domain"@en .

### BloodType
wban:BloodType rdf:type owl:DatatypeProperty ;
    rdfs:domain wban:Patient ;
    rdfs:range [
        rdf:type rdfs:Datatype ;
        owl:oneOf [
            rdf:type rdf:List ;
            rdf:first "A+" ;
            rdf:rest [
                rdf:type rdf:List ;
                rdf:first "A-" ;
                rdf:rest [
                    rdf:type rdf:List ;
                    rdf:first "B+" ;
                    rdf:rest [
                        rdf:type rdf:List ;
                        rdf:first "B-" ;
                        rdf:rest [
                            rdf:type rdf:List ;
                            rdf:first "AB+" ;
                            rdf:rest [
                                rdf:type rdf:List ;
                                rdf:first "AB-" ;
                                rdf:rest [
                                    rdf:type rdf:List ;
                                    rdf:first "O+" ;
                                    rdf:rest [
                                        rdf:type rdf:List ;
                                        rdf:first "O-" ;
                                        rdf:rest rdf:nil
                                    ]
                                ]
                            ]
                        ]
```

```
                    ]
                ]
            ]
        ]
    ] ;
    rdfs:comment "A blood type is a classification system that allows healthcare providers to
determine whether your blood is compatible or incompatible with someone else's blood." ;
    rdfs:label "Blood Type"@en .


###  Density
wban:Density rdf:type owl:DatatypeProperty ;
         rdfs:domain wban:WBAN ;
         rdfs:range xsd:int ;
         rdfs:comment "Indicates the density of a WBAN, measured as an integer. This can refer to
the number of nodes or devices per unit area within the WBAN." ;
         rdfs:label "Density"@en .


###  DeploymentTime
wban:DeploymentTime rdf:type owl:DatatypeProperty ;
                rdfs:domain wban:WBAN ;
                rdfs:range xsd:dateTime ;
                rdfs:comment "Specifies the deployment time of a WBAN, represented as a date-time
value. This property records when the WBAN was set up or became operational." ;
                rdfs:label "Deployment Time"@en .


### FaultTolerence
wban:FaultTolerence rdf:type owl:DatatypeProperty ;
                rdfs:domain wban:WBAN ;
                rdfs:range xsd:real;
                rdfs:comment "Defines the fault tolerance of a WBAN, measured as a real number. This
property indicates the system's ability to continue operating in the event of a failure." ;
                rdfs:label "Fault Tolerence"@en .
###  Language
wban:Language rdf:type owl:DatatypeProperty;
        rdfs:domain wban:Patient ;
        rdfs:range xsd:string ;
        rdfs:comment "Defines the native language of the patient." ;
        rdfs:label "Laguage"@en .


###  LifeTime
wban:LifeTime rdf:type owl:DatatypeProperty;
        rdfs:domain wban:WBAN ;
        rdfs:range xsd:duration ;
        rdfs:comment "Specifies the lifetime of a WBAN." ;
        rdfs:label "Life Time"@en .


### Reference
wban:Reference rdf:type owl:DatatypeProperty ;
      rdfs:domain wban:OnlineRessource ;
      rdfs:range xsd:anyURI ;
      rdfs:comment "Links an online resource to its URI, providing a web address or link to the
resource. This is used to reference online information or services." ;
      rdfs:label "Reference"@en .


###  Phenomena
wban:Phenomena rdf:type owl:DatatypeProperty ;
           rdfs:domain wban:WBAN ;
           rdfs:range xsd:string ;
           rdfs:comment "Describes phenomena related to a WBAN, represented as a string. This
property can include descriptions of environmental or operational phenomena relevant to the WBAN's
function." ;
           rdfs:label "Phenomena"@en .


### RadioTech
wban:RadioTechnology rdf:type owl:DatatypeProperty ;
           rdfs:domain wban:WBAN ;
           rdfs:range xsd:string ;
           rdfs:comment "Defines the radio technology used by a WBAN, specifying the type of
wireless communication technology, such as Bluetooth or Zigbee." ;
           rdfs:label "Radio Technology"@en .
```

```
### Topology
wban:Topology rdf:type owl:DatatypeProperty ;
          rdfs:domain wban:WBAN ;
          rdfs:range [ rdf:type rdfs:Datatype ;
                       owl:oneOf [ rdf:type rdf:List ;
                                   rdf:first "Adhoc" ;
                                   rdf:rest [ rdf:type rdf:List ;
                                              rdf:first "Mesh" ;
                                              rdf:rest [ rdf:type rdf:List ;
                                                         rdf:first "Others" ;
                                                         rdf:rest [ rdf:type rdf:List ;
                                                                    rdf:first "P2P" ;
                                                                    rdf:rest [ rdf:type rdf:List ;
                                                                               rdf:first "Star" ;
                                                                               rdf:rest rdf:nil
                                                                             ]
                                                                  ]
                                                       ]
                                            ]
                                 ]
                     ] ;
          rdfs:comment "Indicates the network topology of a WBAN, represented by a string from a
predefined set of values (e.g. Adhoc, Mesh, P2P, Star). This property describes how the nodes within
the WBAN are interconnected." ;
          rdfs:label "Topology"@en .
###  UserID
wban:UserID rdf:type owl:DatatypeProperty , owl:FunctionalProperty ;
        rdfs:domain wban:Patient ;
        rdfs:comment "Connects a patient to their user ID, providing a unique identifier for the
patient within the system." ;
        rdfs:label "User ID"@en .


###  WBANID
wban:WBANID rdf:type owl:DatatypeProperty ,
                 owl:FunctionalProperty ;
        rdfs:domain wban:WBAN ;
        rdfs:range xsd:anyURI ;
        rdfs:comment "Defines a unique identifier for a WBAN, represented as a URI. This property is
also a functional property, meaning each WBAN has a unique ID." ;
        rdfs:label "Wban ID"@en .


#################################################################
#    Classes
#################################################################

###  ByRequest
wban:ByRequest rdf:type owl:Class ;
           rdfs:subClassOf wban:CommunicationProcess ;
           rdfs:comment "A subclass of CommunicationProcess representing communication processes
initiated by a specific request. This process is not periodic or event-driven but occurs upon
demand." ;
           rdfs:label "By Request"@en .


### CommunicationProcess
wban:CommunicationProcess rdf:type owl:Class ;
                      rdfs:comment "Represents the processes involved in data communication within a
WBAN. This class can encompass various communication strategies and protocols." ;
                      rdfs:label "Communication Process"@en .


###  Contact
wban:Contact rdf:type owl:Class ;
         rdfs:subClassOf foaf:Agent ;
         rdfs:comment "Represents contact entities, which can be a person or an organization." ;
         rdfs:label "Contact"@en .

###  Disease
wban:Disease rdf:type owl:Class ;
         rdfs:comment "A disorder of structure or function in a person, especially one that has a
known cause and a distinctive group of symptoms, signs, or anatomical changes.." ;
         rdfs:label "Disease"@en .


### EventDriven
```

```
wban:EventDriven rdf:type owl:Class ;
              rdfs:subClassOf wban:CommunicationProcess ;
              rdfs:comment "A subclass of CommunicationProcess where communication is triggered by
specific events rather than occurring periodically or by request." ;
              rdfs:label "Event Driven"@en .


###  OnlineRessource
wban:OnlineRessource rdf:type owl:Class ;
                 rdfs:subClassOf wban:Contact ;
                 rdfs:comment "A subclass of Contact representing online resources such as web pages
or online services that can be contacted or accessed." ;
                 rdfs:label "Online Ressource"@en .


### Organisation
wban:Organisation rdf:type owl:Class ;
              rdfs:subClassOf wban:contact, foaf:Organization ;
              rdfs:comment "A subclass of Agent representing organizations that have responsibility
or involvement in the WBAN. Organizations can include companies, institutions, or other groups." ;
              rdfs:label "Organization"@en .

###  Patient
wban:Patient rdf:type owl:Class ;
         rdfs:subClassOf wban:HealthActor ;
         rdfs:comment "A subclass of Health Actor representing individuals receiving medical care or
being monitored within the WBAN. Patients have specific health-related properties and measurements."
;
         rdfs:label "Patient"@en .

###  HealthActor
wban:HealthActor rdf:type owl:Class ;
         rdfs:subClassOf wban:Role ;
         rdfs:comment "The eHealth actors like e.g. caregivers, patients, users, helpers...." ;
         rdfs:label "Health Actor"@en .

###  Periodic
wban:Periodic rdf:type owl:Class ;
          rdfs:subClassOf wban:CommunicationProcess ;
          rdfs:comment "A subclass of CommunicationProcess where communication occurs at regular
intervals, defined by a set period." ;
          rdfs:label "Periodic"@en .


###  Person
wban:Person rdf:type owl:Class ;
        rdfs:subClassOf wban:Contact, foaf:Person ;
        rdfs:comment "A subclass of Agent representing individuals." ;
        rdfs:label "Person"@en .


###  ResponsibleParty
wban:ResponsibleParty rdf:type owl:Class ;
                  rdfs:subClassOf wban:Role ;
                  rdfs:comment "A subclass of Role representing entities responsible for the WBAN."
;
                  rdfs:label "Responsible Party"@en .


###  Role
wban:Role rdf:type owl:Class ;
      owl:equivalentClass [ rdf:type owl:Class ;
                            owl:unionOf ( wban:ResponsibleParty
                                          wban:HealthActor
                                        )
                          ] ;
      rdfs:comment "Represents the role of contact entities, which can include  health actors, or
responsible parties. This class is equivalent to a union of these entities." ;
      rdfs:label "Role"@en .


### WBAN
wban:WBAN rdf:type owl:Class ;
      rdfs:comment "Represents a Wireless Body Area Network, a network of wearable computing
devices. WBANs are used for monitoring and managing various health and activity-related parameters."
;
      rdfs:label "Wireless Body Area Network"@en .
```

B.4 node.ttl

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1#> .
@prefix dcterms: <http://purl.org/dc/terms#> .
@prefix vann: <http://purl.org/vocab/vann#> .
@prefix node: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/node#> .
@prefix loc: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/location#> .
@prefix quantity: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/quantity#> .
@base <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/node#> .

<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/node#> rdf:type owl:Ontology ;
                                         dcterms:description "This ontology defines the structure
and properties for various node types within Smart BANs. It offers a unified framework for
characterizing health devices, including sensors and actuators, along with their interfaces and
operational attributes, facilitating interoperability and management of networked health
entities."@en ;
                                         dcterms:creator <https://www.etsi.org/technologies/smart-
body-area-networks> ;
                                         dcterms:contributor [a foaf:Person ; foaf:name "Lina
Nachabe" ] ;
                                          dcterms:contributor [a foaf:Person ; foaf:name "Aya Saïd"
] ;
                                             dcterms:license <https://opensource.org/license/mit/> ;
                                             dcterms:created "2024-12-16"^^xsd:date ;
                                             dcterms:modified "2024-12-16"^^xsd:date;
                                             dcterms:issued "2025-01-20"^^xsd:date;
                                             dcterms:rights "Copyright © ETSI smart body area
netwoks"@en ;
                                             dcterms:title "SmartBAN Node Ontology"@en ;
                                             vann:preferredNamespacePrefix "node" ;
                                             vann:preferredNamespaceUri
<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/node#> ;
                                             rdfs:label "Node"@en ;
                                             owl:versionInfo "2.0".

#################################################################
#    Annotation Properties
#################################################################

###  http://purl.org/dc/terms/contributor
dcterms:contributor rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/creator
dcterms:creator rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/description
dcterms:description rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/issued
dcterms:issued rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/license
dcterms:license rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/modified
dcterms:modified rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/title
dcterms:title rdf:type owl:AnnotationProperty .
```

```
###  http://purl.org/vocab/vann/preferredNamespacePrefix
vann:preferredNamespacePrefix rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespaceUri
vann:preferredNamespaceUri rdf:type owl:AnnotationProperty .

#################################################################
#    Object Properties
#################################################################

###  dutyCycle
node:dutyCycle rdf:type owl:ObjectProperty ;
           rdfs:domain node:Processor ;
           rdfs:range quantity:Quantity ;
           rdfs:comment "Links a processor to its duty cycle, representing the proportion of time a
processor is actively performing tasks compared to its total operational time. A higher duty cycle
indicates more active processing." ;
           rdfs:label "Duty Cycle"@en .

###  modePower
node:modePower rdf:type owl:ObjectProperty ;
           rdfs:domain node:Mode ;
           rdfs:range quantity:Quantity ;
           rdfs:comment "Associates a mode with its power consumption, detailing the amount of power
consumed when a device operates in a particular mode. This helps in managing energy efficiency and
operational planning." ;
           rdfs:label "Mode Power"@en .


###  position
node:position rdf:type owl:ObjectProperty;
          rdfs:domain node:Node ;
          rdfs:range loc:Location;
          rdfs:comment "Links a node to its geographical or spatial coordinates, defining its
physical location within the network or a defined space. This is important for tasks involving
location-based services and tracking." ;
          rdfs:label "Position"@en .


###  wakeupLatency
node:wakeupLatency rdf:type owl:ObjectProperty ;
               rdfs:domain node:NodeType ;
               rdfs:range quantity:Quantity;
               rdfs:comment "Connects a node type to the delay experienced when transitioning from a
low-power state to an active state. Lower wakeup latency is essential for systems requiring rapid
responsiveness." ;
               rdfs:label "Wake Up Latency"@en .


### cost
node:cost rdf:type owl:ObjectProperty ;
      rdfs:domain node:NodeType ;
      rdfs:range quantity:Quantity;
      rdfs:comment "Associates a node type with its cost, encompassing the economic or resource
expenditure required to implement or maintain a node of that type. This is used for budgeting and
resource allocation." ;
      rdfs:label "cost"@en .


###  dimension
node:dimension rdf:type owl:ObjectProperty ;
           rdfs:domain node:NodeType ;
           rdfs:range quantity:Quantity ;
           rdfs:comment "Relates a node type to its physical size or spatial extent, including
parameters such as height, width, and depth. This information is important for physical installation
and spatial planning." ;
           rdfs:label "Dimension"@en .


###  frequency
node:frequency rdf:type owl:ObjectProperty ;
           rdfs:domain node:Processor ;
           rdfs:range quantity:Quantity;
           rdfs:comment "Links a processor to its operating frequency, indicating the speed at which
the processor's clock operates, typically measured in hertz (Hz). Higher frequencies generally
correspond to faster processing speeds." ;
```

```
           rdfs:label "Frequency"@en .


###  AvailableFlash
node:hasAvailableFlash rdf:type owl:ObjectProperty ;
                   rdfs:domain node:MemoEnergy ;
                   rdfs:range quantity:Quantity;
                   rdfs:comment "Connects a memory-related entity to the amount of flash memory
currently available for use. Flash memory is a type of non-volatile storage used in various
devices." ;
                   rdfs:label "has Available Flash"@en .


###  AvailableRAM
node:hasAvailableRAM rdf:type owl:ObjectProperty ;
                 rdfs:domain node:MemoryEnergy ;
                 rdfs:range quantity:Quantity;
                 rdfs:comment "Relates a memory-related entity to the available random-access memory
(RAM), which is crucial for the temporary storage of data for active processes." ;
                 rdfs:label "has Available RAM"@en .

###  baud
node:baudRate rdf:type owl:ObjectProperty ;
      rdfs:domain node:InterfaceType ;
      rdfs:range quantity:Quantity ;
      rdfs:comment "Provides the baud rate of an interface type, indicating the speed of data
transmission in symbols per second." ;
      rdfs:label "Baud Rate"@en .

###  hasEnergySrc
node:hasEnergySrc rdf:type owl:ObjectProperty ;
              rdfs:domain node:NodeType ;
              rdfs:range node:EnergySource ;
              rdfs:comment "Associates a node type with its energy source, specifying the type of
power supply used, such as batteries or external power sources. This is critical for managing energy
resources and sustainability." ;
              rdfs:label "Has Energy Source"@en .


###  hasInterface
node:hasInterface rdf:type owl:ObjectProperty ;
              rdfs:domain node:NodeType ;
              rdfs:range node:Interface ;
              rdfs:comment "Links a node to its interfaces, indicating the points of interaction
through which the node can communicate with other devices or networks. Interfaces can include ports,
network adapters, and other connection points." ;
              rdfs:label "Has Interface"@en .


###  hasInterfaceType
node:hasInterfaceType rdf:type owl:ObjectProperty ;
                  rdfs:domain [ rdf:type owl:Class ;
                                owl:unionOf ( node:Interface
                                              node:NodeType
                                            )
                              ] ;
                  rdfs:range node:InterfaceType ;
                  rdfs:comment "Relates an interface or node type to its specific type of interface,
detailing the kind of connection or communication protocol used, such as Ethernet, Wi-Fi, or
Bluetooth." ;
                  rdfs:label "has interface type"@en .


###  hasLinkState
node:hasLinkState rdf:type owl:ObjectProperty ;
              rdfs:domain node:Node ;
              rdfs:range node:LinkState ;
              rdfs:comment "Connects a node to the current status of its network link, providing
information about the connection quality, operational status, and any issues affecting the link." ;
              rdfs:label "Has Link State"@en .


###  hasLinkToNode
node:hasLinkToNode rdf:type owl:ObjectProperty ;
               rdfs:domain node:LinkState ;
               rdfs:range node:Node ;
               rdfs:comment "Relates a link state to a specific node it is connected to, detailing
the endpoints of the network link. This helps in understanding network topology and connections." ;
```

```
               rdfs:label "Has Link to Node"@en .


###  MAXFlash
node:MAXFlash rdf:type owl:ObjectProperty ;
             rdfs:domain node:Processor ;
             rdfs:range quantity:Quantity;
             rdfs:comment "Associates a processor with its maximum supported flash memory capacity, 
defining the upper limit of non-volatile storage the processor can handle." ;
             rdfs:label "Max Flash Memory"@en .


###  MAXRAM
node:MAXRAM rdf:type owl:ObjectProperty ;
            rdfs:domain node:Processor ;
            rdfs:range quantity:Quantity;
           rdfs:comment "Connects a processor to its maximum supported RAM capacity, indicating the 
highest amount of temporary memory the processor can manage effectively." ;
           rdfs:label "Max Random Access Memory"@en .


###  hasMemoEnergy
node:hasMemoEnergy rdf:type owl:ObjectProperty ;
               rdfs:domain node:Node ;
               rdfs:range node:MemoEnergy ;
               rdfs:comment "Links a node to its memory energy, representing the node's energy 
storage or status related to its memory operations." ;
               rdfs:label "Has Memory Energy"@en .


### hasMode
node:hasMode rdf:type owl:ObjectProperty ;
          rdfs:domain node:NodeType ;
          rdfs:range node:Mode ;
          rdfs:comment "Relates a node type to the operational modes it supports, such as standby, 
active, or sleep modes. This is important for managing energy efficiency and functionality." ;
          rdfs:label "Has Mode"@en .


###  hasNodeType
node:hasNodeType rdf:type owl:ObjectProperty ;
             rdfs:domain node:Node ;
             rdfs:range node:NodeType ;
             rdfs:comment "Connects a node to its specific type, categorizing the node within the 
network based on its function, capabilities, or other defining characteristics." ;
             rdfs:label "Has Node Type"@en .


### hasProcessor
node:hasProcessor rdf:type owl:ObjectProperty ;
              rdfs:domain node:NodeType ;
              rdfs:range node:Processor ;
              rdfs:comment "Associates a node type with its processor, specifying the computational 
unit responsible for executing instructions and processing data within nodes of that type." ;
              rdfs:label "Has Processor"@en .


###  RemainBatteryLevel
node:hasRemainBattery rdf:type owl:ObjectProperty ;
                  rdfs:domain node:MemoEnergy ;
                  rdfs:range quantity:Quantity;
                  rdfs:comment "Relates a memory energy entity to the remaining battery capacity, 
providing information on how much power is left for operation." ;
                  rdfs:label "Remaining Battery Level"@en .


### inCoverage
node:inCoverage rdf:type owl:ObjectProperty ;
            rdfs:domain node:NodeType ;
            rdfs:range quantity:Quantity;
            rdfs:comment "Connects a node type to its network coverage distance." ;
            rdfs:label "In Coverage"@en .


###  inCurrentMode
node:inCurrentMode rdf:type owl:ObjectProperty ;
               rdfs:domain node:NodeType ;
               rdfs:range node:Mode ;
```

```
               rdfs:comment "Links a node to its current operational mode, detailing the specific
mode in which the node is functioning at a given time." ;
               rdfs:label "In Current Mode"@en .



###  outCoverage
node:outCoverage rdf:type owl:ObjectProperty ;
             rdfs:domain node:NodeType ;
             rdfs:range quantity:Quantity;
             rdfs:comment "Relates a node type to its status outside the network coverage area,
indicating nodes that are not within the network's operational range." ;
             rdfs:label "Out Coverage"@en .


###  receivingSensitivity
node:receivingSensitivity rdf:type owl:ObjectProperty ;
                      rdfs:domain node:NodeType ;
                      rdfs:range quantity:Quantity;
                      rdfs:comment "Associates a node type with its sensitivity to incoming signals,
detailing the minimum signal strength required for effective reception." ;
                      rdfs:label "Receiving Sensitivity"@en .


###  velocity
node:velocity rdf:type owl:ObjectProperty ;
          rdfs:domain node:NodeType ;
          rdfs:range quantity:Quantity;
          rdfs:comment "Connects a node type to its speed or rate of motion, important for
applications involving mobile or moving nodes within the network." ;
          rdfs:label "Velocity"@en .


#################################################################
#    Data properties
#################################################################

###  addressType
node:addressType rdf:type owl:DatatypeProperty ;
          rdfs:subPropertyOf node:interfaceAddress ;
          rdfs:range xsd:string ;
          rdfs:comment "Specifies the type or category of an address associated with an interface,
such as IPv4, IPv6, MAC address, etc." ;
          rdfs:label "Adress Type"@en .


### addressValue
node:addressValue rdf:type owl:DatatypeProperty ;
           rdfs:subPropertyOf node:interfaceAddress ;
           rdfs:comment "Provides the actual value of an address for an interface, detailing the
unique identifier used for network communication." ;
           rdfs:label "Address Value"@en .


###  errorRate
node:errorRate rdf:type owl:DatatypeProperty ;
           rdfs:domain node:LinkState ;
           rdfs:range xsd:float ;
           rdfs:comment "Associates a link state with its error rate, quantifying the proportion of
erroneous data transmissions over the link. This metric is important for assessing link reliability
and quality." ;
           rdfs:label "Error Rate"@en .


###  firmwareVersion
node:firmwareVersion rdf:type owl:DatatypeProperty ;
             rdfs:domain node:NodeType ;
             rdfs:range xsd:string ;
             rdfs:comment "Specifies the firmware version running on a node type, providing
information about the software version controlling the hardware." ;
             rdfs:label "Firmware Version"@en .


###  firmware
node:firmware rdf:type owl:DatatypeProperty ;
          rdfs:domain node:NodeType ;
          rdfs:range xsd:string ;
```

```
          rdfs:comment "Details the firmware used by a node type, indicating the specific software
that provides low-level control over the device's hardware." ;
          rdfs:label "Firmware"@en .


### interfaceDescription
node:interfaceDescription rdf:type owl:DatatypeProperty ;
                rdfs:domain node:InterfaceType ;
                rdfs:range xsd:string ;
                rdfs:comment "Provides a description of an interface type, explaining its
characteristics, functionalities, and use cases." ;
                rdfs:label "Interface Description"@en .


###  interfaceLayer
node:interfaceLayer rdf:type owl:DatatypeProperty ;
          rdfs:domain node:InterfaceType ;
          rdfs:range xsd:string ;
          rdfs:comment "Specifies the network layer (such as physical, data link, network) at which
an interface type operates, defining its role within the network stack." ;
          rdfs:label "Interface Layer"@en .


###  interfaceName
node:interfaceName rdf:type owl:DatatypeProperty ;
         rdfs:domain node:InterfaceType ;
         rdfs:range xsd:string ;
         rdfs:comment "Specifies the name of an interface type, which may be used for identification
and configuration purposes." ;
         rdfs:label "Interface Name"@en .


###  interfaceProtocol
node:interfaceProtocol rdf:type owl:DatatypeProperty ;
             rdfs:domain node:InterfaceType ;
             rdfs:range xsd:string ;
             rdfs:comment "Specifies the communication protocol used by an interface type, such as
TCP/IP, UDP, or others, defining the rules for data exchange." ;
             rdfs:label "Interface Protocol"@en .


###  interfaceType
node:interfaceType rdf:type owl:DatatypeProperty ;
         rdfs:domain node:InterfaceType ;
         rdfs:range xsd:string ;
         rdfs:comment "Provides the type of an interface, such as wired, wireless, serial, etc.,
indicating the method of connection and communication." ;
         rdfs:label "Interface Type"@en .


###  interfaceAddress
node:interfaceAddress rdf:type owl:DatatypeProperty ;
                  rdfs:domain node:Interface ;
                  rdfs:range xsd:string;
                  rdfs:comment "Specifies the address of an interface, used for identifying and
accessing the interface within the network." ;
                  rdfs:label "Interface Address"@en .


###  interfaceID
node:interfaceID rdf:type owl:DatatypeProperty ,
                      owl:FunctionalProperty ;
             rdfs:domain node:Interface ;
             rdfs:range xsd:anyURI ;
             rdfs:comment "Provides a unique identifier for an interface, often in the form of a
URI." ;
             rdfs:label "Interface Identifier"@en .


###  modeID
node:modeID rdf:type owl:DatatypeProperty ,
                 owl:FunctionalProperty ;
        rdfs:domain node:Mode ;
        rdfs:range xsd:anyURI ;
        rdfs:comment "Specifies a unique identifier for a mode, often in the form of a URI." ;
        rdfs:label "Mode Identifier"@en .
```

```
###  modeName
node:modeName rdf:type owl:DatatypeProperty ;
          rdfs:domain node:Mode ;
          rdfs:range xsd:string ;
          rdfs:comment "Provides the name of a mode, used for identification and operational
purposes." ;
          rdfs:label "Mode Name"@en .


###  NodeID
node:nodeID rdf:type owl:DatatypeProperty ,
                 owl:FunctionalProperty ;
        rdfs:domain node:Node ;
        rdfs:range xsd:anyURI ;
        rdfs:comment "Specifies a unique identifier for a node, often in the form of a URI." ;
        rdfs:label "Node Identfier"@en .


### NodeTypeDescription
node:nodeTypeDescription rdf:type owl:DatatypeProperty ;
                     rdfs:domain node:NodeType ;
                     rdfs:range xsd:string ;
                     rdfs:comment "Gives a detailed description of a node type, explaining its
characteristics, functionalities, and use cases." ;
                     rdfs:label "Node Type Description"@en .


### NodeTypeName
node:nodeTypeName rdf:type owl:DatatypeProperty ;
              rdfs:domain node:NodeType ;
              rdfs:range xsd:string ;
              rdfs:comment "Provides the name of a node type, used for identification and
classification purposes." ;
              rdfs:label "Node Type Name"@en .


###  operatingConstraints
node:operatingConstraints rdf:type owl:DatatypeProperty ;
                      rdfs:domain node:NodeType ;
                      rdfs:range xsd:string ;
                      rdfs:comment "Specifies any operational constraints associated with a node
type, such as environmental conditions, power requirements, or other limitations." ;
                      rdfs:label "Operating Constraints"@en .


### ProcessorID
node:processorID rdf:type owl:DatatypeProperty ;
             rdfs:domain node:Processor ;
             rdfs:range xsd:anyURI ;
             rdfs:comment "Provides a unique identifier for a processor, often in the form of a
URI." ;
             rdfs:label "Processor Identifier"@en .


###  rechargeable
node:rechargeable rdf:type owl:DatatypeProperty ;
              rdfs:domain node:EnergySource ;
              rdfs:range xsd:boolean ;
              rdfs:comment "Indicates whether an energy source is rechargeable, providing a boolean
value (true/false)." ;
              rdfs:label "Rechargeable"@en .


###  serialNb
node:serialNb rdf:type owl:DatatypeProperty ;
          rdfs:domain node:Node ;
          rdfs:range xsd:string;
          rdfs:comment "Specifies the serial number of a node, used for unique identification of
hardware components." ;
          rdfs:label "Serial Number"@en .


###  sourceType
node:sourceType rdf:type owl:DatatypeProperty ;
            rdfs:domain node:EnergySource ;
            rdfs:range xsd:string ;
            rdfs:comment "Provides the type of an energy source, such as battery, solar, or AC
power, indicating the method of energy supply." ;
```

```
            rdfs:label "Source Type"@en .


###  trustLevel
node:trustLevel rdf:type owl:DatatypeProperty ;
            rdfs:domain node:Node ;
            rdfs:range xsd:integer;
            rdfs:comment "Specifies the trust level of a node, often used in security contexts to
indicate the reliability or integrity of a node." ;
            rdfs:label "Trust Level"@en .





### delay
node:delay rdf:type owl:DatatypeProperty ;
       rdfs:domain node:LinkState ;
       rdfs:range xsd:duration ;
       rdfs:comment "Specifies the delay associated with a link state, measured in units such as
milliseconds, indicating the latency of the link." ;
       rdfs:label "Delay"@en .


###  isGateway
node:isGateway rdf:type owl:DatatypeProperty ;
           rdfs:domain node:Interface ;
           rdfs:range xsd:boolean ;
           rdfs:comment "Indicates whether an interface functions as a gateway, providing a boolean
value (true/false)." ;
           rdfs:label "Is Gateway"@en .

###  MIPS
node:MIPS rdf:type owl:DatatypeProperty ;
      rdfs:domain node:Processor ;
      rdfs:range xsd:integer;
      rdfs:comment "Connects a processor to its performance capability, measured in Millions of
Instructions Per Second (MIPS). This indicates the number of instructions a processor can execute in
one second, providing a measure of its computational power." ;
      rdfs:label "MIPS"@en .

###  nodeDiscoveryTime
node:nodeDiscoveryTime rdf:type owl:DatatypeProperty ;
                   rdfs:domain node:NodeType ;
                   rdfs:range xsd:duration;
                   rdfs:comment "Relates a node type to the time required for the network to detect
and recognize a node of that type. It is crucial for network initialization and expansion
processes." ;
                   rdfs:label "Node Discovery Time"@en .

#################################################################
#    Classes
#################################################################

###  Actuator
node:Actuator rdf:type owl:Class ;
          rdfs:subClassOf node:Node ;
          owl:disjointWith node:Sensor ;
          rdfs:comment "A subclass of nodes that represent devices capable of performing actions or
controlling mechanisms in response to signals, often used in automation and control systems." ;
          rdfs:label "Actuator"@en .

###  Hub
node:Hub rdf:type owl:Class ;
     rdfs:subClassOf node:Node ;
     rdfs:comment "A subclass of nodes that act as central points in a network, connecting multiple
devices and facilitating communication among them, commonly used in network topologies." ;
     rdfs:label "Hub"@en .

###  Node
node:Node rdf:type owl:Class ;
      rdfs:comment "Represents an individual device or entity within the network, capable of
communication, processing, or other functions, forming the fundamental units of the network." ;
      rdfs:label "Node"@en .


###  NodeType
node:NodeType rdf:type owl:Class ;
```

```
          rdfs:disjointWith node:Inerface;
          rdfs:comment "Represents categories or types of nodes, grouping devices based on shared
characteristics, functionalities, or roles within the network." ;
          rdfs:label "Node Type"@en .

###  Sensor
node:Sensor rdf:type owl:Class ;
        rdfs:subClassOf node:Node ;
        rdfs:comment "A subclass of nodes that represent devices capable of detecting and measuring
environmental conditions or changes, such as temperature sensors, motion detectors, or light
sensors." ;
        rdfs:label "Sensor"@en .

###  Wearable
node:Wearable rdf:type owl:Class ;
        rdfs:subClassOf node:Node ;
        rdfs:comment "A subclass of nodes that represent devices that the patient wears." ;
        rdfs:label "Wearable"@en .

###  Mode
node:Mode a rdfs:Class ;
    rdfs:comment "The operational mode of a node." ;
      rdfs:label "Mode"@en .

###  Interface
node:Interface a rdfs:Class ;
    rdfs:comment "An interface for communication between nodes." ;
      rdfs:label "Interface"@en .

###  InterfaceType
node:InterfaceType a rdfs:Class ;
    rdfs:comment "The type of interface used in the WBAN." ;
      rdfs:label "Interface Type"@en .

### Processor
node:Processor a rdfs:Class ;
    rdfs:comment "A processor associated with a node." ;
      rdfs:label "Processor"@en .

### EnergySource
node:EnergySource a rdfs:Class ;
    rdfs:comment "An energy source powering a node." ;
      rdfs:label "Energy Source"@en .

### MemoEnergy
node:MemoryEnergy a rdfs:Class ;
    rdfs:comment "The remaining memory and energy of a node." ;
      rdfs:label "Memory Energy"@en .

#### Link State
node:LinkState a rdfs:Class ;
    rdfs:comment "The state of the communication link between nodes." ;
      rdfs:label "Link State"@en .
```

B.5 process.ttl

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1#> .
@prefix dcterms: <http://purl.org/dc/terms#> .
@prefix vann: <http://purl.org/vocab/vann#> .
@prefix quantity: https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/quantity#> .
@prefix process: https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/process#> .
@prefix dcat: <https://www.w3.org/TR/vocab-dcat-3#> .
@base https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/process#> .

https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/process#> rdf:type owl:Ontology ;
```

```
                                          dcterms:description "This ontology defines the process,
data, actions and measurements within Smart BANs."@en ;
                                          dcterms:creator <https://www.etsi.org/technologies/smart-
body-area-networks> ;
                                          dcterms:contributor [a foaf:Person ; foaf:name "Lina
Nachabe" ] ;
                                           dcterms:contributor [a foaf:Person ; foaf:name "Aya Saïd"
] ;
                                              dcterms:license <https://opensource.org/license/mit/> ;
                                              dcterms:created "2024-12-16"^^xsd:date ;
                                              dcterms:modified "2024-12-16"^^xsd:date;
                                              dcterms:issued "2025-01-20"^^xsd:date;
                                              dcterms:rights "Copyright © ETSI smart body area
netwoks"@en ;
                                              dcterms:title "SmartBAN Process Ontology"@en ;
                                              vann:preferredNamespacePrefix "process" ;
                                              vann:preferredNamespaceUri
https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/process#> ;
                                              rdfs:label "Process"@en ;
owl:versionInfo "2.0".

#################################################################
#    Object Properties
#################################################################

###  SampleRate
process:SampleRate rdf:type owl:ObjectProperty ;
            rdfs:domain process:Process ;
            rdfs:range quantity:Quantity;
            rdfs:comment "Associates a process with its sampling rate, defining the frequency at
which data is sampled or collected by the process." ;
            rdfs:label "Sample Rate"@en .
###  calibration
process:calibration rdf:type owl:ObjectProperty ;
             rdfs:domain process:Process ;
             rdfs:range quantity:Quantity;
             rdfs:comment "Connects a process to its calibration parameters or procedures, detailing
the adjustments made to ensure accurate operation." ;
             rdfs:label "Calibration"@en .



###  drift
process:drift rdf:type owl:ObjectProperty ;
       rdfs:domain process:Process ;
       rdfs:range quantity:Quantity;
       rdfs:comment "Relates a process to its drift characteristics, describing any gradual changes
or deviations from the expected behavior over time." ;
       rdfs:label "Drift"@en .


###  hasConstraint
process:hasConstraint rdf:type owl:ObjectProperty ;
               rdfs:domain process:Data ;
               rdfs:range process:Constraint ;
               rdfs:comment "Associates data with constraints, specifying limitations or conditions
that must be satisfied for the data to be valid or usable." ;
               rdfs:label "Has Constraint"@en .


### hasData
process:hasData rdf:type owl:ObjectProperty ;
         rdfs:domain process:Process ;
         rdfs:range process:Data ;
         rdfs:comment "Links a process to the data it operates on, indicating the input data
required for the process or the output data generated by the process." ;
         rdfs:label "Has Data"@en .


###  hasDerivedData
process:hasDerivedData rdf:type owl:ObjectProperty ;
                rdfs:domain process:Process ;
                rdfs:range process:Data ;
                rdfs:comment "Relates a process to data that is derived or generated as a result of
the process's operation." ;
                rdfs:label "Has Derived Data"@en .
```

```
###  hasInput
process:hasInput rdf:type owl:ObjectProperty ;
          rdfs:domain process:Process ;
          rdfs:range process:Process ;
          rdfs:comment "Links a process to its input process." ;
          rdfs:label "Has Input"@en .

###  hasOuput
process:hasOuput rdf:type owl:ObjectProperty ;
          rdfs:domain process:Process ;
          rdfs:range process:Process ;
          rdfs:comment "Relates a process to its output process." ;
          rdfs:label "Has Output"@en .


###  hasMesurement
process:hasMeasurement rdf:type owl:ObjectProperty ;
               rdfs:domain process:Data ;
               rdfs:range process:Measurement ;
               rdfs:comment "Connects data to its measurement parameters, detailing the method or
process used to obtain the data and any associated measurement characteristics." ;
               rdfs:label "Has Measurement"@en .




###  precision
process:precision rdf:type owl:ObjectProperty ;
           rdfs:domain process:Process ;
           rdfs:range quantity:Quantity;
           rdfs:comment "Links a process to its precision characteristics, specifying the level of
accuracy or detail achieved by the process." ;
           rdfs:label "Precision"@en .


###  resolution
process:resolution rdf:type owl:ObjectProperty ;
            rdfs:domain process:Process ;
            rdfs:range quantity:Quantity;
            rdfs:comment "Connects a process to its resolution, indicating the smallest detectable
change or increment in the process's output." ;
            rdfs:label "Resolution"@en .


###  responseOrder
process:responseOrder rdf:type owl:ObjectProperty ;
               rdfs:domain process:Process ;
               rdfs:range quantity:Quantity;
               rdfs:comment "Relates a process to its response order, detailing the sequence or
order in which the process responds to inputs or events." ;
               rdfs:label "Response Order"@en .


### uom
process:uom rdf:type owl:ObjectProperty ;
     rdfs:domain process:Process ;
     rdfs:range quantity:Unit;
     rdfs:comment "Associates a process with its unit of measurement, specifying the measurement
units used for the process's inputs, outputs, or parameters." ;
     rdfs:label "Unit of Measurement"@en .

###  latency
process:latency rdf:type owl:ObjectProperty ;
         rdfs:domain process:Process ;
         rdfs:range quantity:Quantity;
         rdfs:comment "Associates a process with its latency, defining the delay or response time
associated with the process's operation." ;
         rdfs:label "latency"@en .

#################################################################
#    Data properties
#################################################################

###  AlgorithmVersion
process:AlgorithmVersion rdf:type owl:DatatypeProperty ;
                  rdfs:domain [ rdf:type owl:Class ;
```

```
                                owl:unionOf ( process:Compression
                                              process:Security
                                            )
                              ] ;
                  rdfs:range xsd:string ;
                  rdfs:comment "Specifies the version of an algorithm used by a process, providing
information about the algorithm's implementation or characteristics." ;
                  rdfs:label "Algorithm Version"@en .


###  CompAlgorithm
process:CompAlgorithm rdf:type owl:DatatypeProperty ;
               rdfs:domain process:Compression ;
               rdfs:range xsd:string ;
               rdfs:comment "Associates compression-related processes with the algorithms used for
compression, specifying the compression methods or techniques applied." ;
               rdfs:label "Compression Algorith"@en .


###  Key
process:Key rdf:type owl:DatatypeProperty ;
     rdfs:domain process:Security ;
     rdfs:range xsd:string;
     rdfs:comment "Relates security-related processes to encryption keys or cryptographic keys used
for securing data or communications." ;
     rdfs:label "Encryption Key"@en .


###  KeyLength
process:KeyLength rdf:type owl:DatatypeProperty ;
           rdfs:domain process:Security ;
           rdfs:range xsd:integer ;
           rdfs:comment "Connects security-related processes to the length of encryption keys or
cryptographic keys used, indicating the key size or strength." ;
           rdfs:label "Encryption Key Length"@en .


###  MaxValue
process:MaxValue rdf:type owl:DatatypeProperty ;
          rdfs:domain process:Validity ;
          rdfs:comment "Specifies the maximum value associated with validity-related processes,
defining the upper limit or threshold for valid data or measurements." ;
          rdfs:label "Maximum Value"@en .


###  inValue
process:MinValue rdf:type owl:DatatypeProperty ;
          rdfs:domain process:Validity ;
          rdfs:comment "Associates validity-related processes with the minimum value, defining the
lower limit or threshold for valid data or measurements." ;
          rdfs:label "Minimum Value"@en .


###  ProcessID
process:ProcessID rdf:type owl:DatatypeProperty ,
                    owl:FunctionalProperty ;
           rdfs:domain process:Process ;
           rdfs:range xsd:anyURI ;
           rdfs:comment "Provides a unique identifier for a process." ;
           rdfs:label "Process Identifier"@en .


###  QOS
process:QOS rdf:type owl:DatatypeProperty ;
     rdfs:domain process:Data ;
     rdfs:range xsd:int ;
     rdfs:comment "Associates data with its quality of service (QoS) class value." ;
     rdfs:label "Quality of Service"@en .


###  SecAlgorithm
process:SecAlgorithm rdf:type owl:DatatypeProperty ;
              rdfs:domain process:Security ;
              rdfs:range xsd:string ;
              rdfs:comment "Relates security-related processes to the algorithms used for encryption
or security measures, specifying cryptographic algorithms or methods." ;
              rdfs:label "Security Algorithm"@en .
```

```
###  TTL
process:TTL rdf:type owl:DatatypeProperty ;
     rdfs:domain process:Measurement ;
     rdfs:range xsd:int ;
     rdfs:comment "Specifies the Time-to-Live (TTL) value associated with measurements, indicating
the duration or lifespan of the measurement data." ;
     rdfs:label "Time to Live"@en .


###  TimeStamp
process:TimeStamp rdf:type owl:DatatypeProperty ;
           rdfs:domain process:Measurement ;
           rdfs:range xsd:dateTimeStamp ;
           rdfs:comment "Associates measurements with timestamps, indicating the time at which the
measurements were taken or recorded." ;
           rdfs:label "Time Stamp"@en .


###  valid
process:valid rdf:type owl:DatatypeProperty ;
       rdfs:domain process:Measurement ;
       rdfs:range xsd:boolean ;
       rdfs:comment "Specifies the validity status of measurements, indicating whether the
measurements are considered valid or invalid." ;
       rdfs:label "Valid"@en .


###  value
process:value rdf:type owl:DatatypeProperty ;
       rdfs:domain quantity:Quantity ;
       rdfs:comment "Provides the value of measurements, indicating the measured quantity,
parameter, or characteristic." ;
       rdfs:label "Value"@en .


#################################################################
#    Classes
#################################################################

###  Compression
process:Compression rdf:type owl:Class ;
             rdfs:subClassOf process:Constraint ;
             rdfs:comment "Represents processes related to data compression, which reduce the size
of data for efficient storage or transmission." ;
             rdfs:label "Compression"@en .


### Constraint
process:Constraint rdf:type owl:Class ;
 owl:unionOf ( process:Compression process:Operating process:Security process:Validity );
            rdfs:comment "Represents constraints or limitations associated with data, processes, or
measurements, defining conditions that must be satisfied." ;
            rdfs:label "Constraint"@en .


###  Data
process:Data rdf:type owl:Class ;
      rdfs:subClassOf dcat:Dataset ;
      rdfs:comment "Represents data involved in processes, including input data, output data, or
intermediate data generated during processing." ;
      rdfs:label "Data"@en .



###  Operating
process:Operating rdf:type owl:Class ;
           rdfs:subClassOf process:Constraint ;
           rdfs:comment "Represents operational constraints or requirements associated with
processes, defining conditions or limitations for process execution." ;
           rdfs:label "Operating"@en .


###  Process
process:Process rdf:type owl:Class ;
         rdfs:comment "Represents processes or operations performed on data, including data
processing tasks, computational algorithms, or signal processing operations." ;
```

```
        rdfs:label "Process"@en .


### Security
process:Security rdf:type owl:Class ;
          rdfs:subClassOf process:Constraint ;
          rdfs:comment "Represents security-related processes or measures, including encryption, 
authentication, or access control mechanisms." ;
          rdfs:label "Security"@en .


### Validity
process:Validity rdf:type owl:Class ;
          rdfs:subClassOf process:Constraint ;
          rdfs:comment "Represents validity constraints or requirements associated with data or 
measurements, defining conditions for data correctness or accuracy." ;
          rdfs:label "Validity"@en .

### Measurement
process:Measurement rdf:type owl:Class ;
          rdfs:comment "Represents the specific quantitative or qualitative data derived from the 
physical property being sensed." ;
          rdfs:label "Measurement"@en .

### Action
process:Action rdf:type owl:Class ;
          rdfs:comment "Represents the extent or nature of the physical manipulation or control that 
is being executed by the actuator, often described in terms of degrees, speeds, positions, or on/off 
states. These values are the result of input commands (often electrical or digital signals) that 
trigger the actuator to perform the desired physical action." ;
          rdfs:label "Action"@en .
```

B.6 ontofull.ttl

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1#> .
@prefix dcterms: <http://purl.org/dc/terms#> .
@prefix vann: <http://purl.org/vocab/vann#> .
@prefix node: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/node#>.
@prefix quantity: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/quantity#>.
@prefix location: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/location#>.
@prefix wban: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/wban#>.
@prefix process: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/process#>.
@prefix full: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ontofull#>.

@base <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ontofull#> .

<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ontofull#> rdf:type owl:Ontology ;
                                         dcterms:description "This ontology defines the 
SmartBAN."@en ;
                                         dcterms:creator <https://www.etsi.org/technologies/smart-
body-area-networks> ;
                                         dcterms:contributor [a foaf:Person ; foaf:name "Lina 
Nachabe" ] ;
                                          dcterms:contributor [a foaf:Person ; foaf:name "Aya Saïd" 
] ;
                                             dcterms:license <https://opensource.org/license/mit/> ;
                                             dcterms:created "2024-12-16"^^xsd:date ;
                                             dcterms:modified "2024-12-16"^^xsd:date;
                                             dcterms:issued "2025-01-20"^^xsd:date;
                                             dcterms:rights "Copyright © ETSI smart body area 
netwoks"@en ;
                                             dcterms:title "SmartBAN Ontology"@en ;
```

```
                                             vann:preferredNamespacePrefix "loc" ;
                                             vann:preferredNamespaceUri
<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ontofull#>;
                                             rdfs:label "Ontofull"@en ;
                                             owl:imports node:; quantity:; location:; process:;
wban:;
                                             owl:versionInfo "2.0".

#################################################################
#    Annotation Properties
#################################################################

###  http://purl.org/dc/terms/contributor
dcterms:contributor rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/creator
dcterms:creator rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/description
dcterms:description rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/issued
dcterms:issued rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/license
dcterms:license rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/modified
dcterms:modified rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/title
dcterms:title rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespacePrefix
vann:preferredNamespacePrefix rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespaceUri
vann:preferredNamespaceUri rdf:type owl:AnnotationProperty .

#################################################################
#    Object Properties
#################################################################

full:BelongTo a owl:ObjectProperty ;
    rdfs:domain node:Node ;
    rdfs:range wban:WBAN ;
    owl:inverseOf full:Contains ;
    owl:maxCardinality 1 .

full:Contains a owl:ObjectProperty ;
    rdfs:domain wban:WBAN ;
    rdfs:range node:Node ;
    owl:inverseOf full:BelongTo .

full:DoAction a owl:ObjectProperty ;
    rdfs:domain node:Actuator ;
    rdfs:range process:Action .

full:ElectHub a owl:ObjectProperty ;
    rdfs:domain wban:WBAN ;
    rdfs:range node:Hub ;
    owl:maxCardinality 1 .


full:isMeasuredBy a owl:ObjectProperty ;
    rdfs:domain process:Measurement ;
    rdfs:range wban:Sensor ;
    owl:maxCardinality 1 ;
    owl:inverseOf full:Measures .
```

```
full:Measures a owl:ObjectProperty ;
    rdfs:domain node:Sensor ;
    rdfs:range process:Measurement ;
    owl:minCardinality 1 ;
    owl:inverseOf full:isMeasuredBy .

full:usedFor a owl:ObjectProperty ;
    rdfs:domain node:Node ;
    rdfs:range process:Process ;
    owl:minCardinality 1 .
```

B.7 ServiceProfile.ttl (SmartBAN ServiceProfile module is called MyOntoServiceProfile ontology)

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix serviceProfile: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceProfile#> .
@prefix vann: <http://purl.org/vocab/vann#> .
@prefix dcterms: <http://purl.org/dc/terms#> .
@prefix foaf: <http://xmlns.com/foaf/0.1#> .
@base <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceProfile#> .

<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceProfile#> rdf:type
owl:Ontology ;
                                                 dcterms:creator
<https://www.etsi.org/committee/smartban#> ;
                                                 dcterms:contributor [a foaf:Person ; foaf:name
"Lina Nachabe" ] ;
                                                 dcterms:contributor [a foaf:Person ; foaf:name
"Aya Saïd" ] ;
                                                 dcterms:created "2015-12-01"^^xsd:date ;
                                                dcterms:rights "Copyright © ETSI Smart Body Area
Netwoks"@en ;
                                                 dcterms:description "This ontology defines a
comprehensive framework for categorizing and detailing services within Smart BANs. It encompasses
various service types and parameters, including quality and constraints, facilitating effective
service management and interoperability across diverse applications."@en ;
                                                dcterms:license
<https://opensource.org/license/mit/> ;
                                                  dcterms:modified "2024-09"@en ;
                                                  dcterms:issued "2024-09"@en ;
                                                 dcterms:title "SmartBan Service Profile
Ontology"@en ;
                                                  vann:preferredNamespacePrefix "serviceProfile" ;
                                                   vann:preferredNamespaceUri
<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceProfile#> ;
                                                    rdfs:label "SmartBan Service Profile
Ontology"@en ;
                                                      owl:versionInfo "v2.0" .

#################################################################
#    Annotation Properties
#################################################################

###  http://purl.org/dc/terms/contributor
dcterms:contributor rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/creator
dcterms:creator rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/description
dcterms:description rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/issued
```

```
dcterms:issued rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/license
dcterms:license rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/modified
dcterms:modified rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/title
dcterms:title rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespacePrefix
vann:preferredNamespacePrefix rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespaceUri
vann:preferredNamespaceUri rdf:type owl:AnnotationProperty .

#################################################################
#    Object Properties
#################################################################

### hasParameter
serviceProfile:hasParameter rdf:type owl:ObjectProperty ;
              rdfs:range serviceProfile:Parameter ;
              rdfs:comment "Links a service profile to its associated parameter(s). This property
defines what parameters are involved in a service profile, helping to detail the specific aspects or
configurations related to the service." ;
              rdfs:label "has parameter"@en .


###  hasCalculationMethod
serviceProfile:hasCalculationMethod rdf:type owl:ObjectProperty ;
                      rdfs:domain serviceProfile:QoS ;
                      rdfs:range serviceProfile:QoI ;
                      rdfs:comment "Associates a Quality of Service (QoS) with a method for
calculating Quality of Information (QoI). This helps define how the quality metrics for the
information provided by the service are derived." ;
                      rdfs:label "has calculation method"@en .


### hasConstarint
serviceProfile:hasConstarint rdf:type owl:ObjectProperty ;
               rdfs:subPropertyOf serviceProfile:hasServiceParameter ;
               rdfs:domain serviceProfile:ServiceProfile ;
               rdfs:range serviceProfile:ServiceConstraints ;
               rdfs:comment "Links a service profile to its constraints, specifying the limitations
or conditions under which the service operates. This property is a subproperty of
'hasServiceParameter' and indicates the rules or boundaries defined for the service." ;
               rdfs:label "has constraint"@en .


###  hasQoI
serviceProfile:hasQoI rdf:type owl:ObjectProperty ;
        rdfs:subPropertyOf serviceProfile:hasServiceParameter ;
        rdfs:domain serviceProfile:ServiceProfile ;
        rdfs:range serviceProfile:QoI ;
        rdfs:comment "Connects a service profile to its Quality of Information parameters, detailing
the aspects that define the accuracy, completeness, and other information quality metrics." ;
        rdfs:label "has quality of information"@en .


###  hasQoS
serviceProfile:hasQoS   rdfs:subPropertyOf serviceProfile:hasServiceParameter ;
        rdfs:domain serviceProfile:ServiceProfile ;
        rdfs:range serviceProfile:QoS ;
        rdfs:comment "Links a service profile to its Quality of Service parameters, defining the
performance characteristics such as delay, jitter, and reliability." ;
        rdfs:label "has quality of service"@en .


###  hasServiceCategory
serviceProfile:hasServiceCategory rdf:type owl:ObjectProperty ;
                    rdfs:domain serviceProfile:ServiceProfile ;
```

```
                    rdfs:range serviceProfile:ServiceCategory ;
                    rdfs:comment "Associates a service profile with its category, grouping the
service into a broader classification that helps understand its general function or application
area." ;
                    rdfs:label "has service category"@en .


###  hasServiceClassification
serviceProfile:hasServiceClassification rdf:type owl:ObjectProperty ;
                          rdfs:domain serviceProfile:ServiceProfile ;
                          rdfs:comment "Links a service profile to its classification, defining how
the service is categorized or classified within the ontology." ;
                          rdfs:label "has service classification"@en .


###  hasServiceParameter
serviceProfile:hasServiceParameter rdf:type owl:ObjectProperty ;
                     rdfs:comment "General property for linking a service profile to its parameters,
including both constraints and quality metrics, providing a detailed specification of the service
attributes." ;
                     rdfs:label "has service parameter"@en .


###  hasServiceProcess
serviceProfile:hasServiceProcess rdf:type owl:ObjectProperty ;
                   rdfs:domain serviceProfile:ServiceProfile ;
                   rdfs:range serviceProfile:ServiceProcess ;
                   rdfs:comment "Connects a service profile to the service process it involves,
detailing the sequence of operations or steps that the service performs." ;
                   rdfs:label "has service process"@en .


###  hasServiceType
serviceProfile:hasServiceType rdf:type owl:ObjectProperty ;
                rdfs:domain serviceProfile:ServiceProfile ;
                rdfs:range serviceProfile:ServiceType ;
                rdfs:comment "Associates a service profile with its type, specifying the kind of
service it is, such as Alert, Command, Monitoring, etc." ;
                rdfs:label "has service type"@en .


#################################################################
#    Data properties
#################################################################

###  Cause
serviceProfile:Cause rdf:type owl:DatatypeProperty ;
       rdfs:domain serviceProfile:Alert ;
       rdfs:range xsd:string ;
       rdfs:comment "Describes the cause associated with an alert, providing information about what
triggered the alert. This helps in understanding the context and reason behind an alert." ;
       rdfs:label "cause"@en .


###  CommandDescription
serviceProfile:CommandDescription rdf:type owl:DatatypeProperty ;
              rdfs:domain serviceProfile:Command ;
              rdfs:range xsd:string ;
              rdfs:comment "Gives a description of a command, detailing what the command does and
its purpose. This is essential for understanding the functionality of different commands within the
service profile." ;
              rdfs:label "command description"@en .


### RequestPeriod
serviceProfile:RequestPeriod rdf:type owl:DatatypeProperty ;
               rdfs:domain serviceProfile:Periodic ;
               rdfs:comment "Specifies the period or interval at which requests are made in a
periodic service, detailing the frequency of requests." ;
               rdfs:label "request period"@en .


###  Text
serviceProfile:Text rdf:type owl:DatatypeProperty ;
      rdfs:domain serviceProfile:Notification ;
      rdfs:range xsd:string ;
      rdfs:comment "Contains the text of a notification, providing the actual message or content
delivered in a notification service." ;
```

```
      rdfs:label "text"@en .


###  accuracy
serviceProfile:accuracy rdf:type owl:DatatypeProperty ;
          rdfs:domain serviceProfile:QoI ;
          rdfs:comment "Defines the accuracy of the Quality of Information (QoI), indicating how
precise the information is compared to the actual value or standard." ;
          rdfs:label "accuracy"@en .


###  code
serviceProfile:code rdf:type owl:DatatypeProperty ,
               owl:FunctionalProperty ;
      rdfs:domain serviceProfile:ServiceType ;
      rdfs:comment " Provides a unique code for a service type, acting as an identifier for
different types of services." ;
      rdfs:label "code"@en .


### completness
serviceProfile:completness rdf:type owl:DatatypeProperty ;
             rdfs:domain serviceProfile:QoI ;
             rdfs:comment "Describes the completeness of the Quality of Information (QoI),
indicating the extent to which the information is complete and covers all necessary aspects." ;
             rdfs:label "completness"@en .


### correctness
serviceProfile:correctness rdf:type owl:DatatypeProperty ;
             rdfs:domain serviceProfile:QoI ;
             rdfs:comment "Indicates the correctness of the Quality of Information (QoI), defining
how accurate and error-free the information is." ;
             rdfs:label "correctness"@en .


###  delay
serviceProfile:delay rdf:type owl:DatatypeProperty ;
       rdfs:domain serviceProfile:QoS ;
       rdfs:comment "Specifies the delay in the Quality of Service (QoS), measuring the time taken
for data to travel from source to destination." ;
       rdfs:label "delay"@en .


###  freshness
serviceProfile:freshness rdf:type owl:DatatypeProperty ;
           rdfs:domain serviceProfile:QoI ;
           rdfs:comment "Defines the freshness of the Quality of Information (QoI), indicating how
up-to-date and timely the information is." ;
           rdfs:label "freshness"@en .


###  jitter
serviceProfile:jitter rdf:type owl:DatatypeProperty ;
        rdfs:domain serviceProfile:QoS ;
        rdfs:comment "Measures the variation in delay in the Quality of Service (QoS), important for
understanding the consistency of the data transmission." ;
        rdfs:label "jitter"@en .


###  latency
serviceProfile:latency rdf:type owl:DatatypeProperty ;
         rdfs:domain serviceProfile:QoI ;
         rdfs:comment "Specifies the latency in the Quality of Information (QoI), measuring the
delay before the transfer of data begins following an instruction for its transfer." ;
         rdfs:label "latency"@en .


### packetLoss
serviceProfile:packetLoss rdf:type owl:DatatypeProperty ;
            rdfs:domain serviceProfile:QoS ;
            rdfs:comment "Describes the packet loss in the Quality of Service (QoS), indicating the
percentage of packets lost during transmission, affecting the reliability of the service." ;
            rdfs:label "packet loss"@en .


###  precision
serviceProfile:precision rdf:type owl:DatatypeProperty ;
```

```
           rdfs:domain serviceProfile:QoI ;
           rdfs:comment "Indicates the precision of the Quality of Information (QoI), defining the
detail level of the information." ;
           rdfs:label "precision"@en .


###  reliability
serviceProfile:reliability rdf:type owl:DatatypeProperty ;
             rdfs:domain serviceProfile:QoS ;
             rdfs:comment "Defines the reliability of the Quality of Service (QoS), indicating how
dependable and consistent the service is over time." ;
             rdfs:label "reliability"@en .


###  robustness
serviceProfile:robustness rdf:type owl:DatatypeProperty ;
            rdfs:domain serviceProfile:QoS ;
            rdfs:comment "Measures the robustness of the Quality of Service (QoS), describing how
well the service can handle variations and continue to function under different conditions." ;
            rdfs:label "robustness"@en .


### security
serviceProfile:security rdf:type owl:DatatypeProperty ;
          rdfs:domain serviceProfile:QoS ;
          rdfs:comment "Indicates the security level of the Quality of Service (QoS), detailing the
measures taken to protect the service from unauthorized access and threats." ;
          rdfs:label "security"@en .


### serviceDescription
serviceProfile:serviceDescription rdf:type owl:DatatypeProperty ;
                    rdfs:domain serviceProfile:ServiceProfile ;
                    rdfs:range xsd:string ;
                    rdfs:comment "Provides a description of a service profile, detailing the
purpose, functionality, and key characteristics of the service." ;
                    rdfs:label "service description"@en .


###  serviceName
serviceProfile:serviceName rdf:type owl:DatatypeProperty ,
                                    owl:FunctionalProperty ;
                           rdfs:domain serviceProfile:ServiceProfile ;
                           rdfs:range xsd:anyURI ;
                           rdfs:comment "Specifies the name of a service profile, providing a unique
identifier for the service, often in the form of a URI." ;
                           rdfs:label "service name"@en .


### throughput
serviceProfile:throughput rdf:type owl:DatatypeProperty ;
                          rdfs:domain serviceProfile:QoS ;
                          rdfs:comment "Measures the throughput in the Quality of Service (QoS),
indicating the amount of data successfully transmitted over a network in a given period." ;
                          rdfs:label "throughput"@en .


#################################################################
#    Classes
#################################################################

###  Alert
serviceProfile:Alert rdf:type owl:Class ;
                     rdfs:subClassOf serviceProfile:ServiceType ,
                                     [ rdf:type owl:Restriction ;
                                       owl:onProperty serviceProfile:hasServiceCategory ;
                                       owl:someValuesFrom serviceProfile:ServiceCategory
                                     ] ,
                                     [ rdf:type owl:Restriction ;
                                       owl:onProperty serviceProfile:hasServiceType ;
                                       owl:someValuesFrom serviceProfile:ServiceType
                                     ] ,
                                     [ rdf:type owl:Restriction ;
                                       owl:onProperty serviceProfile:hasServiceCategory ;
                                       owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                                       owl:onClass serviceProfile:ServiceCategory
                                     ] ,
                                     [ rdf:type owl:Restriction ;
```

```
                                        owl:onProperty serviceProfile:hasServiceType ;
                                        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                                        owl:onClass serviceProfile:ServiceType
                                      ] ;
                     rdfs:comment "Represents a type of service that issues alerts or notifications.
This class includes restrictions ensuring that each alert has a specific service category and type."
;
                     rdfs:label "Alert"@en .


### Ambient
serviceProfile:Ambient rdf:type owl:Class ;
                       rdfs:subClassOf serviceProfile:ServiceCategory ;
                       rdfs:comment "A subclass of ServiceCategory that includes services related to
the ambient environment, such as monitoring weather conditions or environmental data." ;
                       rdfs:label "Ambient"@en .


###  Automation
serviceProfile:Automation rdf:type owl:Class ;
                          rdfs:subClassOf serviceProfile:Ambient ;
                          rdfs:comment "A subclass of Ambient, representing services involved in
automating processes, often in smart home or industrial settings." ;
                          rdfs:label "Automation"@en .


### Body
serviceProfile:Body rdf:type owl:Class ;
                    rdfs:subClassOf serviceProfile:ServiceCategory ;
                    rdfs:comment "A subclass of ServiceCategory that includes services related to
the human body, such as health monitoring or fitness tracking." ;
                    rdfs:label "Body"@en .


###  Command
serviceProfile:Command rdf:type owl:Class ;
                       rdfs:subClassOf serviceProfile:ServiceType ;
                       rdfs:comment "Represents a type of service that involves issuing commands,
often used in control systems and automation." ;
                       rdfs:label "Command"@en .


###  General
serviceProfile:General rdf:type owl:Class ;
                       rdfs:subClassOf serviceProfile:Body ;
                       rdfs:comment "A subclass of Body, representing general services related to
the body that do not fall into more specific categories like medical." ;
                       rdfs:label "General"@en .


###  LegalConstraint
serviceProfile:LegalConstraint rdf:type owl:Class ;
                               rdfs:subClassOf serviceProfile:ServiceConstraints ;
                               rdfs:comment "Represents legal constraints associated with services,
detailing legal requirements or restrictions that must be adhered to." ;
                               rdfs:label "Legal constraint"@en .


###  Medical
serviceProfile:Medical rdf:type owl:Class ;
                       rdfs:subClassOf serviceProfile:Body ;
                       rdfs:comment "A subclass of Body, representing services specifically related
to medical and healthcare applications." ;
                       rdfs:label "Medical"@en .


### Monitoring
serviceProfile:Monitoring rdf:type owl:Class ;
                          rdfs:subClassOf serviceProfile:ServiceType ;
                          rdfs:comment "Represents a type of service that involves monitoring
various conditions or parameters. This class can include subclasses for different monitoring
methods, such as periodic or on-request monitoring." ;
                          rdfs:label "Monitoring"@en .


###  Notification
serviceProfile:Notification rdf:type owl:Class ;
                            rdfs:subClassOf serviceProfile:ServiceType ,
```

```
                                             [ rdf:type owl:Restriction ;
                                               owl:onProperty serviceProfile:hasServiceType ;
                                               owl:maxQualifiedCardinality
"1"^^xsd:nonNegativeInteger ;
                                               owl:onClass serviceProfile:ServiceType
                                             ] ;
                            rdfs:comment "Represents a type of service that sends notifications.
This class includes restrictions to ensure proper classification of notification services." ;
                            rdfs:label "Notification"@en .


### Object
serviceProfile:Object rdf:type owl:Class ;
                      rdfs:subClassOf serviceProfile:ServiceCategory ;
                      rdfs:comment "A subclass of ServiceCategory that includes services related to
physical objects, such as asset tracking or inventory management." ;
                      rdfs:label "Object"@en .


### OnRequest
serviceProfile:OnRequest rdf:type owl:Class ;
                         rdfs:subClassOf serviceProfile:Monitoring ;
                         rdfs:comment "A subclass of Monitoring, representing services that provide
monitoring on-demand rather than continuously or periodically." ;
                         rdfs:label "On request"@en .


###  Periodic
serviceProfile:Periodic rdf:type owl:Class ;
                        rdfs:subClassOf serviceProfile:Monitoring ;
                        rdfs:comment "A subclass of Monitoring, representing services that perform
monitoring at regular intervals." ;
                        rdfs:label "Periodic"@en .


###  QoI
serviceProfile:QoI rdf:type owl:Class ;
                   rdfs:subClassOf serviceProfile:ServiceParameter ;
                   rdfs:comment "Represents the Quality of Information parameters, detailing various
metrics related to the quality of information provided by a service." ;
                   rdfs:label "Quality of information"@en .


###  QoS
serviceProfile:QoS rdf:type owl:Class ;
                   rdfs:subClassOf serviceProfile:ServiceParameter ;
                   rdfs:comment "Represents the Quality of Service parameters, detailing various
metrics related to the performance and reliability of the service." ;
                   rdfs:label "Quality of service"@en .


###  SecurityConstraint
serviceProfile:SecurityConstraint rdf:type owl:Class ;
                                  rdfs:subClassOf serviceProfile:ServiceConstraints ;
                                  rdfs:comment "Represents security constraints associated with
services, detailing security measures and requirements that must be met." ;
                                  rdfs:label "Security constraint"@en .


###  ServiceCategory
serviceProfile:ServiceCategory rdf:type owl:Class ;
                               owl:equivalentClass [ rdf:type owl:Class ;
                                                     owl:unionOf ( serviceProfile:Ambient
                                                                   serviceProfile:Body
                                                                   serviceProfile:Object
                                                                 )
                                                   ] ;
                               rdfs:comment "Represents categories of services, grouping services
into broader classes such as Ambient, Body, or Object. It is defined as a union of these
subclasses." ;
                               rdfs:label "Service category"@en .


###  ServiceConstraints
serviceProfile:ServiceConstraints rdf:type owl:Class ;
                    rdfs:subClassOf serviceProfile:ServiceParameter ;
                    rdfs:comment "Represents constraints associated with services, detailing
limitations or conditions for service operation." ;
```

```
                    rdfs:label "Service constraints"@en .


### ServiceParameter
serviceProfile:ServiceParameter rdf:type owl:Class ;
                                rdfs:comment "Represents various parameters associated with
services, including both QoI and QoS metrics as well as constraints." ;
                                rdfs:label "Service parameter"@en .


###  ServiceProfile
serviceProfile:ServiceProfile rdf:type owl:Class ;
                              rdfs:comment "Represents the profile of a service, encompassing all
relevant parameters, constraints, and classifications that define the service." ;
                              rdfs:label "Service profile"@en .


###  ServiceType
serviceProfile:ServiceType rdf:type owl:Class ;
                           owl:equivalentClass [ rdf:type owl:Class ;
                                                 owl:unionOf ( serviceProfile:Alert
                                                               serviceProfile:Command
                                                               serviceProfile:Monitoring
                                                               serviceProfile:Notification
                                                             )
                                               ] ;
                           rdfs:comment "Represents types of services, categorizing them into
specific functions such as Alert, Command, Monitoring, and Notification. It is defined as a union of
these subclasses." ;
                           rdfs:label "Service type"@en .


###  ValidityConstraint
serviceProfile:ServiceConstraints rdf:type owl:Class ;
                                  rdfs:subClassOf serviceProfile:ServiceConstraints ;
                                  rdfs:comment "Represents constraints related to the validity of
services, ensuring that services meet specific criteria or conditions." ;
                                  rdfs:label "Validity constraint"@en .


###  Weather
serviceProfile:Weather rdf:type owl:Class ;
                       rdfs:subClassOf serviceProfile:Ambient ;
                       rdfs:comment "A subclass of Ambient, representing services related to weather
monitoring and forecasting." ;
                       rdfs:label "Weather"@en .
```

B.8 ServiceProcess.ttl (SmartBAN ServiceProcess module is called MyOntoServiceProcess ontology)

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix serviceProcess: https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceProcess#> .
@prefix vann: <http://purl.org/vocab/vann#> .
@prefix dcterms: <http://purl.org/dc/terms#> .
@prefix foaf: <http://xmlns.com/foaf/0.1#> .
@base https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceProcess#> .

https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceProcess#>  rdf:type
owl:Ontology ;
                                                    owl:versionIRI
https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceProcess#> ;
                                                    dcterms:creator
<https://www.etsi.org/committee/smartban#> ;
                                                    dcterms:contributor [a foaf:Person ; foaf:name
"Lina Nachabe" ] ;
                                                    dcterms:contributor [a foaf:Person ; foaf:name
"Aya Saïd" ] ;
```

```
                                                    dcterms:created "2015-12-01"^^xsd:date ;
                                                    dcterms:rights "Copyright © ETSI Smart Body Area
Netwoks"@en ;
                                                    dcterms:description "This ontology serves as a
foundational element for modeling service processes within Smart BANs. It provides a comprehensive
framework that defines relationships among atomic, composite, and simple processes, along with their
input and output parameters. Additionally, it encompasses preconditions, postconditions, and
calculation methods, facilitating effective management and integration of service processes in
heterogeneous environments."@en ;
                                                    dcterms:license
<https://opensource.org/license/mit/> ;
                                                    dcterms:modified "2024-09"@en ;
                                                    dcterms:issued "2024-09"@en ;
                                                    dcterms:title "SmartBan Service Process
Ontology"@en ;
                                                    vann:preferredNamespacePrefix "serviceProcess" ;
                                                    vann:preferredNamespaceUri
https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceProcess#> ;
                                                    rdfs:label "SmartBan Service Process
Ontology"@en ;
                                                    owl:versionInfo "v2.0" .

#################################################################
#    Annotation Properties
#################################################################

###  http://purl.org/dc/terms/contributor
dcterms:contributor rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/creator
dcterms:creator rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/description
dcterms:description rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/issued
dcterms:issued rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/license
dcterms:license rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/modified
dcterms:modified rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/title
dcterms:title rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespacePrefix
vann:preferredNamespacePrefix rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespaceUri
vann:preferredNamespaceUri rdf:type owl:AnnotationProperty .

#################################################################
#    Object Properties
#################################################################

###  collapses
serviceProcess:collapses rdf:type owl:ObjectProperty ;
           rdfs:domain serviceProcess:Simple ;
           rdfs:range serviceProcess:Composite ;
           rdfs:comment "Indicates that a simple service process can be collapsed into a composite
service process. It represents the transformation or integration of simpler processes into more
complex, aggregated processes." ;
           rdfs:label "collapses"@en .


###  expandsTo
serviceProcess:expandsTo rdf:type owl:ObjectProperty ;
```

```
           rdfs:domain serviceProcess:Composite ;
           rdfs:range serviceProcess:Simple ;
           rdfs:comment "Signifies that a composite service process can be expanded into simpler
service processes. It shows how a complex process can be broken down into its simpler components." ;
           rdfs:label "expands to"@en .


###  hasCalculationMethod
serviceProcess:hasCalculationMethod rdf:type owl:ObjectProperty ;
                      rdfs:domain serviceProcess:ServiceProcess ;
                      rdfs:range serviceProcess:Method ;
                      rdfs:comment "Associates a service process with its method of calculation,
specifying the computational or procedural approach used to achieve the process's goals." ;
                      rdfs:label "has calculation method"@en .


### hasEffect
serviceProcess:hasEffect rdf:type owl:ObjectProperty ;
           rdfs:domain serviceProcess:ServiceProcess ;
           rdfs:range serviceProcess:Effect ;
           rdfs:comment "Links a service process to its resulting effect, indicating the outcomes or
impacts produced by executing the process." ;
           rdfs:label "has effect"@en .


###  hasInput
serviceProcess:hasInput rdf:type owl:ObjectProperty ;
          rdfs:subPropertyOf serviceProcess:hasParameter ;
          rdfs:domain serviceProcess:ServiceProcess ;
          rdfs:range serviceProcess:InputParam ;
          rdfs:comment "Specifies the input parameters required for a service process. It details
the necessary data or conditions needed to initiate or execute the process." ;
          rdfs:label "has input"@en .


### hasOutput
serviceProcess:hasOutput rdf:type owl:ObjectProperty ;
           rdfs:subPropertyOf serviceProcess:hasParameter ;
           rdfs:domain serviceProcess:ServiceProcess ;
           rdfs:range serviceProcess:OutputParam ;
           rdfs:comment "Defines the output parameters of a service process. It describes the data
or results produced upon the completion of the process." ;
           rdfs:label "has output"@en .


### hasParameter
serviceProcess:hasParameter rdf:type owl:ObjectProperty ;
              rdfs:domain serviceProcess:ServiceProcess ;
              rdfs:comment "A general property that relates a service process to its parameters,
encompassing both input and output parameters needed or produced by the process." ;
              rdfs:label "has parameter"@en .


###  hasPrecondition
serviceProcess:hasPrecondition rdf:type owl:ObjectProperty ;
                 rdfs:domain serviceProcess:ServiceProcess ;
                 rdfs:range serviceProcess:PreCondition ;
                 rdfs:comment "Indicates the preconditions required for a service process. These are
conditions that must be met before the process can begin." ;
                 rdfs:label "has precondition"@en .


### hasProcessRelation
serviceProcess:hasProcessRelation rdf:type owl:ObjectProperty ;
                    rdfs:domain serviceProcess:Simple ;
                    rdfs:range serviceProcess:ProcessRelation ;
                    rdfs:comment "Associates a simple service process with its process relations,
detailing how it connects or interacts with other processes." ;
                    rdfs:label "has process relation"@en .


###  hasResult
serviceProcess:hasResult rdf:type owl:ObjectProperty ;
           rdfs:domain serviceProcess:ServiceProcess ;
           rdfs:range serviceProcess:Result ;
           rdfs:comment "Links a service process to its result, specifying the outcomes achieved by
executing the process." ;
           rdfs:label "has result"@en .
```

```
###  isAssociatedWith
serviceProcess:isAssociatedWith rdf:type owl:ObjectProperty ;
                  rdfs:domain serviceProcess:ServiceProcess ;
                  rdfs:comment "Indicates an association between a service process and other related
entities or concepts, without specifying the exact nature of the relationship." ;
                  rdfs:label "is associated with"@en .


###  isRelatedTo
serviceProcess:isRelatedTo rdf:type owl:ObjectProperty ;
             rdfs:domain serviceProcess:ProcessRelation ;
             rdfs:range serviceProcess:Simple ;
             rdfs:comment "Specifies a relationship between a process relation and a simple service
process, indicating how they are connected or interact." ;
             rdfs:label "is related to"@en .


#################################################################
#    Data properties
#################################################################

### RelationType
serviceProcess:RelationType rdf:type owl:DatatypeProperty ;
              rdfs:domain serviceProcess:ProcessRelation ;
              rdfs:range xsd:string ;
              rdfs:comment "Specifies the type of relation in a process relation, typically
described as a string. This can include types like sequential, parallel, conditional, etc." ;
              rdfs:label "relation type"@en .


#################################################################
#    Classes
#################################################################

###  Aggregation
serviceProcess:Aggregation rdf:type owl:Class ;
             rdfs:subClassOf serviceProcess:Method ;
             rdfs:comment "Represents a method of calculation that aggregates or combines multiple
values or results, such as summing or averaging." ;
             rdfs:label "Aggregation"@en .


### Atomic
serviceProcess:Atomic rdf:type owl:Class ;
        rdfs:subClassOf serviceProcess:ServiceProcess ;
        owl:disjointWith serviceProcess:Simple ;
        rdfs:comment "Represents an indivisible service process that cannot be broken down further.
It is a fundamental process that operates as a single unit." ;
        rdfs:label "Atomic"@en .


###  Average
serviceProcess:Average rdf:type owl:Class ;
         rdfs:subClassOf serviceProcess:Aggregation ;
         rdfs:comment "A type of aggregation method that calculates the average value from a set of
numbers." ;
         rdfs:label "Average"@en .


###  Composite
serviceProcess:Composite rdf:type owl:Class ;
           rdfs:subClassOf serviceProcess:ServiceProcess ;
           rdfs:comment "Represents a complex service process composed of multiple simpler
processes. It shows how basic processes are integrated into a cohesive whole." ;
           rdfs:label "Composite"@en .


###  Condition
serviceProcess:Condition rdf:type owl:Class ;
           owl:equivalentClass [ rdf:type owl:Class ;
                                 owl:unionOf ( serviceProcess:PostCondition
                                               serviceProcess:PreCondition
                                             )
                               ] ;
           rdfs:comment "Represents conditions associated with service processes, including both
preconditions (required before a process) and postconditions (resulting after a process)." ;
```

```
           rdfs:label "Condition"@en .


###  Effect
serviceProcess:Effect rdf:type owl:Class ;
        rdfs:comment "Represents the outcome or impact resulting from a service process. It details 
the changes or results produced by the process." ;
        rdfs:label "Effect"@en .


### Formula
serviceProcess:Formula rdf:type owl:Class ;
         rdfs:subClassOf serviceProcess:Method ;
         rdfs:comment "Represents a mathematical or logical formula used in the calculation method 
of a service process." ;
         rdfs:label "Formula"@en .


###  InputParam
serviceProcess:InputParam rdf:type owl:Class ;
            rdfs:subClassOf serviceProcess:Parameter ;
            rdfs:comment "Represents the input parameters needed for a service process, detailing 
the data or conditions required to start or execute the process." ;
            rdfs:label "Input parameter"@en .


###  LeastSquare
serviceProcess:LeastSquare rdf:type owl:Class ;
             rdfs:subClassOf serviceProcess:Formula ;
             rdfs:comment "A specific type of formula used for regression analysis, minimizing the 
sum of the squares of the differences between observed and calculated values." ;
             rdfs:label "Least square"@en .


###  Max
serviceProcess:Max rdf:type owl:Class ;
     rdfs:subClassOf serviceProcess:Aggregation ;
     rdfs:comment "A type of aggregation method that identifies the maximum value from a set of 
numbers." ;
     rdfs:label "Max"@en .


###  Method
serviceProcess:Method rdf:type owl:Class ;
        rdfs:comment "Represents the methods or techniques used in service processes for 
calculations, operations, or other procedural activities." ;
        rdfs:label "Method"@en .


###  Min
serviceProcess:Min rdf:type owl:Class ;
     rdfs:subClassOf serviceProcess:Aggregation ;
     rdfs:comment "A type of aggregation method that identifies the minimum value from a set of 
numbers." ;
     rdfs:label "Min"@en .


###  OutputParam
serviceProcess:OutputParam rdf:type owl:Class ;
             rdfs:subClassOf serviceProcess:Parameter ;
             rdfs:comment "Represents the output parameters produced by a service process, detailing 
the data or results generated after the process's completion." ;
             rdfs:label "Output parameter"@en .


###  Parameter
serviceProcess:Parameter rdf:type owl:Class ;
           rdfs:comment "Represents the parameters associated with service processes, encompassing 
both inputs and outputs required or produced by the processes." ;
           rdfs:label "Parameter"@en .


###  PostCondition
serviceProcess:PostCondition rdf:type owl:Class ;
               rdfs:subClassOf serviceProcess:Condition ;
               rdfs:comment "Represents conditions that must be met after a service process has been 
completed. It details the expected state or results following the process." ;
               rdfs:label "Post-condition"@en .
```

```
### PreCondition
serviceProcess:PreCondition rdf:type owl:Class ;
              rdfs:subClassOf serviceProcess:Condition ;
              rdfs:comment "Represents conditions that must be met before a service process can
begin. It details the prerequisites required for the process's initiation." ;
              rdfs:label "Pre-condition"@en .


###  ProcessRelation
serviceProcess:ProcessRelation rdf:type owl:Class ;
                 rdfs:comment "Represents the relationships between different service processes,
detailing how they are connected or interact within a larger workflow." ;
                 rdfs:label "Process relation"@en .


###  esult
serviceProcess:Result rdf:type owl:Class ;
        rdfs:comment "Represents the outcomes or results achieved by executing a service process. It
details the end products or effects of the process." ;
        rdfs:label "Result"@en .


###  erviceProcess
serviceProcess:ServiceProcess rdf:type owl:Class ;
                owl:equivalentClass [ rdf:type owl:Class ;
                                      owl:unionOf ( serviceProcess:Atomic
                                                    serviceProcess:Composite
                                                    serviceProcess:Simple
                                                  )
                                    ] ;
                rdfs:comment "Represents the various types of service processes, whether they are
atomic (indivisible), composite (composed of simpler processes), or simple (basic processes)." ;
                rdfs:label "Service process"@en .


###  Simple
serviceProcess:Simple rdf:type owl:Class ;
        rdfs:subClassOf serviceProcess:ServiceProcess ;
        rdfs:comment "Represents basic service processes that are simple and not further divisible.
They operate as straightforward, fundamental processes." ;
        rdfs:label "Simple"@en .


### Sum
serviceProcess:Sum rdf:type owl:Class ;
     rdfs:subClassOf serviceProcess:Aggregation ;
     rdfs:comment "A type of aggregation method that calculates the sum of a set of numbers,
combining them into a single total value." ;
     rdfs:label "Sum"@en .
```

B.9 ServiceGrounding.ttl (SmartBAN ServiceGrounding module is called MyOntoServiceGrounding ontology)

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix serviceGrounding: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceGrounding#> .
@prefix vann: <http://purl.org/vocab/vann#> .
@prefix dcterms: <http://purl.org/dc/terms#> .
@prefix foaf: <http://xmlns.com/foaf/0.1#> .
@base <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceGrounding#> .

<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceGrounding> rdf:type
owl:Ontology ;
                                                   owl:versionIRI
<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceGrounding#> ;
```

```
                                                    dcterms:creator
<https://www.etsi.org/committee/smartban#> ;
                                                    dcterms:contributor [a foaf:Person ; foaf:name
"Lina Nachabe" ] ;
                                                    dcterms:contributor [a foaf:Person ; foaf:name
"Aya Saïd" ] ;
                                                    dcterms:created "2015-12-01"^^xsd:date ;
                                                    dcterms:rights "Copyright © ETSI Smart Body Area
Netwoks"@en ;
                                                    dcterms:description "This ontology establishes a
framework for defining and managing service grounding within communication protocols used in Smart
BANs. It captures the structure and semantics of message elements, options, and their associated
properties, enabling effective interaction and integration across diverse messaging protocols such
as HTTP, MQTT, and CoAP. By providing detailed specifications for message components, this ontology
facilitates interoperability and enhances the clarity of service interactions."@en ;
                                                    dcterms:license
<https://opensource.org/license/mit/> ;
                                                    dcterms:modified "2024-09"@en ;
                                                    dcterms:issued "2024-09"@en ;
                                                    dcterms:title "SmartBan Service Grounding
Ontology"@en ;
                                                    vann:preferredNamespacePrefix "serviceGrounding"
;
                                                    vann:preferredNamespaceUri
<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ServiceGrounding#> ;
                                                    rdfs:label "SmartBan Service Grounding
Ontology"@en ;
                                                    owl:versionInfo "v1.1.9" .

#################################################################
#    Annotation Properties
#################################################################

###  http://purl.org/dc/terms/contributor
dcterms:contributor rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/creator
dcterms:creator rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/description
dcterms:description rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/issued
dcterms:issued rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/license
dcterms:license rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/modified
dcterms:modified rdf:type owl:AnnotationProperty .


###  http://purl.org/dc/terms/title
dcterms:title rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespacePrefix
vann:preferredNamespacePrefix rdf:type owl:AnnotationProperty .


###  http://purl.org/vocab/vann/preferredNamespaceUri
vann:preferredNamespaceUri rdf:type owl:AnnotationProperty .

#################################################################
#    Object Properties
#################################################################

###hasMessageElements
serviceGrounding:hasMessageElements rdf:type owl:ObjectProperty ;
                    rdfs:domain serviceGrounding:ServiceGrounding ;
                    rdfs:range serviceGrounding:MessageElements ;
```

```
                    rdfs:comment "Links a service grounding instance to its constituent message
elements. These elements form the detailed structure or payload of the messages used in service
interactions." ;
                    rdfs:label "has message elements"@en .


###hasMessageOptions
serviceGrounding:hasMessageOptions rdf:type owl:ObjectProperty ;
                   rdfs:domain serviceGrounding:ServiceGrounding ;
                   rdfs:range serviceGrounding:MessageOption ;
                   rdfs:comment "Associates a service grounding instance with its message options,
which may include configurations or preferences that dictate how messages should be processed,
transmitted, or handled." ;
                   rdfs:label "has message options"@en .


#################################################################
#    Data properties
#################################################################

###FeildDescription
serviceGrounding:FeildDescription rdf:type owl:DatatypeProperty ;
                  rdfs:domain [ rdf:type owl:Class ;
                                owl:unionOf ( serviceGrounding:MessageElements
                                              serviceGrounding:MessageOption
                                            )
                              ] ;
                  rdfs:range xsd:string ;
                  rdfs:comment "Provides a textual description of a field within a message element
or message option, offering insights into its purpose, content, or usage." ;
                  rdfs:label "feild descriptions"@en .


###FeildLength
serviceGrounding:FeildLength rdf:type owl:DatatypeProperty ;
             rdfs:domain [ rdf:type owl:Class ;
                           owl:unionOf ( serviceGrounding:MessageElements
                                         serviceGrounding:MessageOption
                                       )
                         ] ;
             rdfs:range xsd:integer ;
             rdfs:comment "Specifies the length of a field within a message element or message
option, typically measured in bytes or characters, defining the size constraints of the field." ;
             rdfs:label "feild length"@en .


###FeildName
serviceGrounding:FeildName rdf:type owl:DatatypeProperty ;
           rdfs:domain [ rdf:type owl:Class ;
                         owl:unionOf ( serviceGrounding:MessageElements
                                       serviceGrounding:MessageOption
                                     )
                       ] ;
           rdfs:range xsd:string ;
           rdfs:comment "Provides the name of a field within a message element or message option,
used for identification and reference within the message structure." ;
           rdfs:label "feild name"@en .


###FeildOrder
serviceGrounding:FeildOrder rdf:type owl:DatatypeProperty ;
            rdfs:domain [ rdf:type owl:Class ;
                          owl:unionOf ( serviceGrounding:MessageElements
                                        serviceGrounding:MessageOption
                                      )
                        ] ;
            rdfs:range xsd:int ;
            rdfs:comment "Indicates the order or sequence of a field within a message element or
message option, which is crucial for correctly interpreting and assembling message data." ;
            rdfs:label "feild order"@en .


###FeildType
serviceGrounding:FeildType rdf:type owl:DatatypeProperty ;
           rdfs:domain [ rdf:type owl:Class ;
                         owl:unionOf ( serviceGrounding:MessageElements
                                       serviceGrounding:MessageOption
                                     )
```

```
                     ] ;
           rdfs:comment "Describes the data type of a field within a message element or message
option, such as integer, string, boolean, etc., defining how the field's data should be
interpreted." ;
           rdfs:label "feild type"@en .


###Port
serviceGrounding:Port rdf:type owl:DatatypeProperty ;
      rdfs:domain serviceGrounding:ServiceGrounding ;
      rdfs:range xsd:int ;
      rdfs:comment "Specifies the port number used by a service grounding instance for
communication, indicating the network port through which messages are sent or received." ;
      rdfs:label "port"@en .


###Reference
serviceGrounding:Reference rdf:type owl:DatatypeProperty ;
           rdfs:domain serviceGrounding:ServiceGrounding ;
           rdfs:range xsd:anyURI ;
           rdfs:comment "Provides a reference URI associated with a service grounding instance,
which may link to documentation, specifications, or related resources." ;
           rdfs:label "reference"@en .


###Required
serviceGrounding:Required rdf:type owl:DatatypeProperty ;
          rdfs:domain [ rdf:type owl:Class ;
                        owl:unionOf ( serviceGrounding:MessageElements
                                      serviceGrounding:MessageOption
                                    )
                      ] ;
          rdfs:range xsd:boolean ;
          rdfs:comment "Indicates whether a field within a message element or message option is
mandatory (true) or optional (false), affecting how strictly message data must conform to the
defined structure." ;
          rdfs:label "required"@en .


###Version
serviceGrounding:Version rdf:type owl:DatatypeProperty ;
         rdfs:domain serviceGrounding:ServiceGrounding ;
         rdfs:range xsd:string ;
         rdfs:comment "Specifies the version of a service grounding instance, providing version
control and compatibility information for the service's message structure and handling." ;
         rdfs:label "version"@en .


#################################################################
#    Classes
#################################################################

###BluetoothLEMsg
serviceGrounding:BluetoothLEMsg rdf:type owl:Class ;
                rdfs:subClassOf serviceGrounding:ServiceGrounding ;
                rdfs:comment "Represents messages used in Bluetooth Low Energy (BLE) communication,
a wireless protocol designed for low power consumption. It is a subclass of ServiceGrounding,
indicating it follows specific service grounding principles and structures." ;
                rdfs:label "Bluetooth LE message"@en .


###CoAP
serviceGrounding:CoAP rdf:type owl:Class ;
      rdfs:subClassOf serviceGrounding:Rest ;
      rdfs:comment "Represents messages used in the Constrained Application Protocol (CoAP), a
protocol designed for simple devices in the Internet of Things (IoT). It is a subclass of Rest,
indicating it adheres to RESTful principles." ;
      rdfs:label "CoAP"@en .


###Email
serviceGrounding:Email rdf:type owl:Class ;
       rdfs:subClassOf serviceGrounding:ServiceGrounding ;
       rdfs:comment "Represents email messages, including their structure and handling within
service interactions. It is a subclass of ServiceGrounding." ;
       rdfs:label "Email"@en .
```

```
###HL7Msg
serviceGrounding:HL7Msg rdf:type owl:Class ;
        rdfs:subClassOf serviceGrounding:ServiceGrounding ;
        rdfs:comment "Represents email messages, including their structure and handling within
service interactions. It is a subclass of ServiceGrounding." ;
        rdfs:label "Health Level 7 Message"@en .


###HTTP
serviceGrounding:HTTP rdf:type owl:Class ;
      rdfs:subClassOf serviceGrounding:Rest ;
      rdfs:comment "Represents messages used in the Hypertext Transfer Protocol (HTTP), a
foundational protocol for web communication. It is a subclass of Rest." ;
      rdfs:label "HTTP"@en .


###MQTT
serviceGrounding:MQTT rdf:type owl:Class ;
      rdfs:subClassOf serviceGrounding:ServiceGrounding ;
      rdfs:comment "Represents messages used in the Message Queuing Telemetry Transport (MQTT)
protocol, designed for lightweight, publish-subscribe network communication. It is a subclass of
ServiceGrounding." ;
      rdfs:label "MQTT"@en .


###MessageElements
serviceGrounding:MessageElements rdf:type owl:Class ;
                 rdfs:comment "Represents individual elements within a message, forming the building
blocks of the message structure." ;
                 rdfs:label "Message Elements"@en .


###MessageOption
serviceGrounding:MessageOption rdf:type owl:Class ;
               rdfs:comment "Represents optional elements or configurations within a message,
allowing customization and flexibility in message handling." ;
               rdfs:label "Message Option"@en .


###Rest
serviceGrounding:Rest rdf:type owl:Class ;
      rdfs:subClassOf serviceGrounding:ServiceGrounding ;
      rdfs:comment "Represents RESTful services and their messages, adhering to the principles of
Representational State Transfer (REST). It is a subclass of ServiceGrounding." ;
      rdfs:label "Rest"@en .


###SMS
serviceGrounding:SMS rdf:type owl:Class ;
     rdfs:subClassOf serviceGrounding:ServiceGrounding ;
     rdfs:comment "Represents Short Message Service (SMS) messages, used for text communication over
mobile networks. It is a subclass of ServiceGrounding." ;
     rdfs:label "SMS"@en .


###SOAP
serviceGrounding:SOAP rdf:type owl:Class ;
      rdfs:subClassOf serviceGrounding:ServiceGrounding ;
      rdfs:comment "Represents messages used in the Simple Object Access Protocol (SOAP), a protocol
for exchanging structured information in web services. It is a subclass of ServiceGrounding." ;
      rdfs:label "SOAP"@en .


###ServiceGrounding
serviceGrounding:ServiceGrounding rdf:type owl:Class ;
                 rdfs:comment "Represents the general concept of service grounding, encompassing
various protocols and message types used in service communication and interactions. It serves as the
foundational class for all specific message types and protocols defined within the ontology." ;
                 rdfs:label "Service Grounding"@en .
```

B.10 Service.ttl (SmartBAN Service module is called MyOntoService ontology)

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1#> .
@prefix dcterms: <http://purl.org/dc/terms#> .
@prefix vann: <http://purl.org/vocab/vann#> .
@prefix service: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ontoService#> .
@prefix serviceGrounding: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/serviceGrounding#> .
@prefix serviceProcess: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/serviceProcess#> .
@prefix serviceProfile: <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/serviceProfile#> .
@base <https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ontoService#> .

<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ontoService#> rdf:type owl:Ontology
;
                                         dcterms:description "This ontology defines the services  in
the SmartBAN."@en ;09:09 27/05/2025
                                         dcterms:creator <https://www.etsi.org/technologies/smart-
body-area-networks> ;
                                         dcterms:contributor [a foaf:Person ; foaf:name "Lina
Nachabe" ] ;
                                          dcterms:contributor [a foaf:Person ; foaf:name "Aya Saïd"
] ;
                                             dcterms:license <https://opensource.org/license/mit/> ;
                                             dcterms:created "2024-12-16"^^xsd:date ;
                                             dcterms:modified "2024-12-16"^^xsd:date;
                                             dcterms:issued "2025-01-20"^^xsd:date;
                                             dcterms:rights "Copyright © ETSI smart body area
netwoks"@en ;
                                             dcterms:title "SmartBAN Service Ontology"@en ;
                                             vann:preferredNamespacePrefix "loc" ;
                                             vann:preferredNamespaceUri
<https://forge.etsi.org/rep/nachabe/etsi-smartban-ontology/-
/blob/b283808a10b9850c2aadbbe85b6aad7ee7e87b65/SmartBANOntology/ontoService#> ;
                                             rdfs:label "Ontology Service"@en ;
                                             owl:versionInfo "2.0";
                                             owl:import serviceGrounding:, serviceProcess:,
serviceProfile:.
```

History

Version	Date	Status
V1.1.1	December 2015	Publication
V1.2.1	January 2026	Publication