



Robust requirements gathering for ontologies in smart water systems

Shaun Howell¹ · Thomas Beach¹ · Yacine Rezgui¹

Received: 2 September 2016 / Accepted: 29 May 2020 / Published online: 22 July 2020
© The Author(s) 2020

Abstract

Urban environments are urgently required to become smarter to overcome sustainability and resilience challenges whilst remaining economically viable. This involves a vast increase in the penetration of ICT resources, both physical and virtual, with the requirement to factor in built environment, socio-economic and human artefacts. This paper, therefore, proposes a methodology for eliciting, testing, and deploying, requirements in the field of urban cybernetics. This extends best practice requirements engineering principles to meet the demands of this growing niche. The paper follows a case study approach of applying the methodology in the smart water domain, where it achieves positive results. The approach not only heavily utilises iteration alongside domain experts, but also mandates the integration of technical domain experts to ensure software requirements are met. A key novelty of the approach is prioritising a balance between (a) knowledge engineers' tenacity for logical accuracy, (b) software engineers' need for speed, simplicity, and integration with other components, and (c) the domain experts' needs to invoke ownership and hence nurture adoption of the resulting ontology.

Keywords Ontology engineering · Smart city · Smart water · Internet of things · Semantic web · Water management

1 Introduction

To tackle global sustainability and economic challenges through information and communication technology (ICT), urban environments, such as water networks, are undergoing a transformation towards 'smart domains, such as smart water, through the use of web-enabled sensors, analytics software, and decision support tools. Smart water networks have been noted to promote efficacy, efficiency, and resilience of water infrastructure [1, 2]. However, as with fields such as smart grids and smart cities, the application of ICT in the water value chain is restricted due to an inability to share data and knowledge, and hence interoperate, across the people and software components involved [3]. In smart grids, this has been stated by IEEE, one of the most authoritative bodies, to occur due to three main issues: lack of

machine communication protocols, lack of common data formats and lack of common meaning of exchanged content [4]. In the 'smart water' domain, the same core issues have restricted the utility and prevalence of ICT penetration. In the smart grid domain, this is being addressed in research, in part through the development of shared data and semantic models to facilitate data exchange, integration of legacy systems, and to enable system security and performance [4]. In the smart water domain, the same key functions of shared models are required, and so recognition of the value of a similar approach in this field is growing. Notably, a recent report from the ICT4Water cluster of European Commission 7th Framework (EC FP7) projects highlighted the need for standardised models to address the issue of interoperability in the smart water domain [5] and specifically indicated the importance of ontologies as a means to maintain semantic clarity and integrate knowledge. All of this leads to a clear emerging challenge in the smart water domain of developing common communication protocols, data models and semantic vocabularies.

Semantic models address the issue of interoperability by creating a shared understanding of the domain and a shared method of representing data and their meaning. Within this remit, various manifestations of what constitutes a 'semantic model' exist, which exhibit a tradeoff between

✉ Shaun Howell
shaunkhowell@gmail.com

Thomas Beach
BeachTH@cardiff.ac.uk

Yacine Rezgui
RezguiY@Cardiff.ac.uk

¹ BRE Institute of Sustainable Engineering, 52 The Parade, Cardiff CF24 3AA, UK

expressiveness and comprehension. Specifically, simple models tend not to capture the nuances of a domain, but are more easily developed, understood and utilised. However, the potential value of exploiting these ‘nuances’ should not be understated, with ontologies representing the highest level of expressive potential but also the greatest potential for human and computational complexity. These benefits have been acknowledged in the field of semantic web technologies through the World Wide Web Consortium (W3C) ‘semantic web stack’, which represents a generic framework for web-based interoperability and shows ontologies playing a critical role. Further, ontology-based models allow the use of inference to produce new knowledge about a system beyond what has been explicitly stated, allows the application of local rules for compliance checking or event triggering, and allows more simple integration of smart water domain knowledge with potential future synergies such as other smart city systems and beyond.

Specifically, whilst communication, protocol, and syntactic interoperability are already being addressed, such as through the internet of things, semantic interoperability remains a largely under-researched issue, especially in the field of water management. Towards addressing this challenge, the development and eventual standardisation of ontological representations of the domain would be highly beneficial.

Within ontology engineering, the role of clear, sufficient, and testable requirements is absolutely critical, to avoid boundless scope and verbosity. To this end, several methodologies exist for ontology engineering, which each have merits, but given the growing importance of the ontological modelling of urban systems, a specialist methodology would be valuable to the growing number of practitioners in this field. Particularly, the design of an ontology for the role of interoperability within an internet of things, or web of things, system, for use in a smart city environment, implies several nuances which support the need for a specialist methodology for this scenario, based on best practices and lessons learnt from the state of the art methodologies. Specifically, a formal recognition of the need for balance between computer science requirements regarding the ontology as a software entity and ontology engineering requirements regarding the ontology as a knowledge modelling tool, at the requirements stage, was not observed in the literature.

The above reasons serve as a clear precedent to develop ontological representations of the water domain, coupled with the need for a robust and repeatable methodology for requirement engineering for ontologies in urban cybernetic applications. The union of these two objectives has manifested in the WISDOM EC FP7 research project, which aims to develop a web-of-things platform, which provides application-layer interoperability through semantic modelling and a Hadoop-based time series store, as well as applications

which deliver domain value and serve as proof of concept for the platform. This served as an ideal case study-based approach for refining existing requirements engineering approaches for ontologies, specific to the field of urban cybernetics. This paper therefore aims to mature existing ontology requirements engineering approaches to (i) address recent changes to the ICT landscape, (ii) be more relevant to the growing case of smart city ontology-driven IoT applications, and (iii) balance the various stakeholders’ perspectives of the ontological process and outputs.

The overall approach adopted was an adaption of the NeOn methodology, within an upper requirement engineering methodology at the platform level. This upper methodology adopted a scenario-driven approach, and impacted heavily on the software requirements and scoping of the ontology requirements. The ontology requirements engineering approach heavily featured domain expert involvement in an iterative manner, combined with semi-automated web crawling and feature extraction. Specifically, the first phase of the semantic modelling activities was to thoroughly understand the challenge faced, followed by significant domain research and knowledge acquisition, and finally the production of formal requirement specifications for both the ontology and the overall ontology service. After gaining a conceptual understanding of the domain, the pilot sites and the role of the ontology service in the WISDOM platform, these were formalised into IDEF0 [6] process models, use case models, explicit scenarios and deployments, and finally, requirement specifications. These were all then iterated through a collaborative process with domain experts to promote their accuracy and completeness. Also, significantly, a collection of relevant ontological and non-ontological models was collected and evaluated for reuse, as is the best practice in ontological modelling. Once the specifications and assorted data objects were validated successfully, the ontology was conceptualised and the reusable resources were merged and extended into a domain-independent meta-model. The main contribution of the paper is the formalisation of the balance between knowledge modelling good practice, software development good practice, domain expert ‘buy-in’, and long-term usefulness of the domain ontology outside of its original scope. This contribution can be reflected on and benefitted from by others undertaking similar tasks, which is an increasingly frequent occurrence as the fields of IoT, Smart Cities, and cybernetics continue to converge.

The rest of the paper proceeds by describing the background and existing work in relevant fields in Sect. 2, followed by the presentation of the high-level platform-ontology requirements engineering approach in Sect. 3. Section 4 then provides detail on the scenario-driven software requirements approach, and Sect. 5 decomposes these into software requirements. Section 6 then describes the elicitation of ontology requirements from a knowledge modelling and

ongoing usefulness perspective. Section 7 then discusses the validation of the approach, before further discussion of the overall approach in Sect. 8 and concluding remarks.

2 Background

2.1 Introduction to ontologies

An ontology, in the broadest sense, is a shared and formal conceptualisation of a domain, and stems from the field of philosophy regarding the nature of knowledge and meaning. This has since been adopted within the computer science field, where it holds a more specific meaning as a software entity used to represent concepts, relationships, descriptions and restrictions in a domain. This has been specialised within the semantic web community as a specific means of integrating data, knowledge and meaning between people and software components. The specific definition is now discussed briefly.

A semantic web ontology is a collection of statements about a domain, structured in a machine-interpretable manner. These collectively form a rich description of the entities and logic perceived in the domain. By expressing data relative to that domain perception, the data are given context, and so is more easily and powerfully consumed by applications. Modern ontologies typically use the W3C semantic web stack, and so are written in a subset of the web ontology language (OWL). This mandates that the statements are formed as resource description framework (RDF) triples, which is to say that they must follow the format of subject–predicate–object, such as ‘Dog’ ‘isATypeOf’ ‘Animal’. Typically though, each atom of a triple is a uniform resource identifier, which often resembles a web address, and must be unique. By formalising all parts of these statements in a detailed manner, they become machine interpretable and enable easier integration across semantic web resources, as well as inference and rule applicability.

An ‘ontology’ often contains two distinct components: a domain ontology, which includes statements generic across all instances of the domain, and the instantiation of this, which contains statements specific to an instance of the domain. The union of these two components forms a knowledge base, and alongside an inference engine, a query engine and a storage capability, this composes a knowledge management system. Within these, the inference engine utilises the statements made to infer new knowledge, the query engine is the method of extracting data and knowledge from the knowledge base, and the storage capability physically and virtually stores the OWL and RDF data on disks and in computer memory. This system may be accessed directly by an interface exposed to users, but more commonly, it will form the backend of an application or several applications.

2.2 Semantic modelling in urban cybernetics and smart water

As a relatively recent challenge, ontological models in the smart city field, and especially the smart water field, are sparse, although some relevant examples were identified. In the broader smart city field, the ISO/IEC Joint Technical Committee’s report on smart cities [7] has highlighted the need for ontologies. Interestingly, IBM developed the SCRIBE smart city ontology 5 years ago [8], and commented on the lack of available ontologies and stable OWL tools at the time. More recently, the CityPulse FP7 project applied semantic tags to Smart City IoT data streams such as twitter feeds through a ‘Stream Annotation Ontology’: a simple ontology about data quality and events, as well as a ‘Complex Event Service’ ontology, which also modelled patterns, preferences, and services [9]. An ontology for city logistics has been developed [10], including the social, technical, and process-oriented aspects across 263 classes. The SEMANCO project developed a large smart city ontology in OWL DL-Lite_A for the purpose of data integration [11], resulting in 592 classes. This appeared to be intended for the exchange of static data in the planning phase of urban areas, given the lack of sensor concepts and dynamic data provision. A very interesting example is the Wi-City ontology [12], because it adopts a cyber–physical–social perspective, reusing existing ontologies to provide a more holistic and powerful integration of services in a smart city system of systems. Also of interest is the foundational ontology for smart city indicators [13], partly because it formally addresses the provenance of data. Another data integration ontology, the open 311 ontology [14], adopts a socio-technical perspective to merge existing public urban datasets with models including CityGML, Towntology, and DBpedia, although the work seems limited in terms of semantic depth. The CityGML standard itself [15] formalises concepts and relationships relevant to geospatial knowledge in cities, and some semantics of the nature of objects and spaces in cities, but in an insufficient manner for interoperability of operational smart city data across verticals. Finally, BSI:PAS 182 [16] proposes a high-level smart city ontology, which serves as an important step, albeit a ‘lowest common denominator’ approach, which hence captures little semantic depth.

Specifically in the smart water field, semantic interoperability is an even more embryonic challenge, caused by the growth of IoT and smart water networks. Attention is increasingly being paid to this challenge though, with an ongoing cluster of EC research projects, ICT4Water, highlighting the importance of semantic modelling [5]. One relevant existing water ontology is that developed in the WatERP project [17], which models water balance concepts from the clean water network at a high level. The WatERP ontology is split conceptually into a ‘supply and demand

ontology’, ‘observation and measurement ontology’ and an ‘alerts and actions’ ontology. Whilst the WatERP ontology is arguably not comprehensive enough, its ‘alerts and actions’ concepts could be reused as a succinct description of water alerts.

Another highly relevant artefact; the semantic water interoperability model (SWIM) [18] formalizes a description of water sector devices such as sensors, pumps, reservoirs and valves. In addition, Waternomics, another ICT4Water project, has developed a ‘linked data model’. In neighbouring environmental science fields, the works of CUAHSI [19], SWEET [20] and HydrOntology [21] are highly notable. The INSPIRE utility network data model [22] is highly relevant to the domain of smart water, but only formalizes more simple smart water concepts and relationships, hence only including 68 named entities. Regardless, this work is still highly relevant, and aligning a comprehensive smart water ontology with this would be highly valuable, as well as the specifications from the INSPIRE hydrography, observation and measurement, and environmental monitoring facility themes. These ontologies, and several others, are compared in Table 1. Finally, WaterML2 [23] is an important standard

but does not express domain semantics, rather it is an encoding format for hydrologic time series data.

From a detailed analysis of previous efforts in semantic modelling in the water management domain, it is evident that research is currently immature towards the challenge of smart water ontological representations; most efforts have been targeted at the earth science domain rather than the manmade water value chain. Therefore, there is a significant gap in the field of capturing in-depth knowledge of the technological, network, social, sensory and ICT artefacts involved in water management decisions in a water value chain.

2.3 Existing requirement capture methods in related fields

Ontology engineering methodologies can be broadly categorised into manual, automatic, or semi-automatic. In general, manual approaches take significantly more human development time, but produce a model more closely coupled with a target system [24]. Given the need in software development for highly specialised artefacts to be

Table 1 Comparison of relevant existing semantic models

Acronym/name	Description	Author	# Entities	Date
SWIM	Device level IoT semantic model for the water industry	Aquamatrix	41	2016
WaterML2	Common format for exchange of hydrological time series data	OGC	131	2014
INSPIRE Water Utility Network Data Model	The INSPIRE directive is establishing an infrastructure for spatial information exchange in Europe, resulting in data models for many application domains, including utility networks, of which water and sewer networks are a subset	EC	68	2013
WatERP	Lightweight ontology of generic concepts for water sensing and management	EURECAT	25 classes	2013
Water data transfer format	Format for transferring flood warning and forecasting data to the governing body	Australian Bureau of Meterology	337	2013
Water Innovation Thesaurus	Aims to facilitate collaboration for water innovation by establishing and highlighting recognised terminology and providing clear definitions for these as well as demonstrating the relationships between terms	EIP Water	548	2013
CityGML UtilityADE	CityGML ADE for the modelling of utility networks in 3d city models, based on topology and component descriptions	OGC	317	2012
SWEET	Middle-level ontology for environmental terminology	NASA	6000	2011
Hydrologic Ontology for Discovery	Supports the discovery of time series hydrologic data collected at a fixed point	CUAHSI	4098	2010
HydrOntology	Aims to integrate data sources regarding hydrographical information from a civil engineering or town planning perspective and a top-down methodology	Vilches-Blázquez et al	250	2009

developed for each system, it was deemed that manual curation of the ontology would be preferable. However, given the requirement of the project to seek benefit from the ontology beyond the lifecycle of the project, such as by contributing to standards, it was deemed that aspects of an automated approach would be beneficial. Manual and then automated approaches are now discussed.

Arguably, the predominant modern methodology for ontology engineering is the NeOn methodology [25], which resulted from an EC 6th framework program research project by the same name. This provides a comprehensive framework of ontology engineering activities, and a number of pathways through the process depending on the specifics of the situation. The approach emphasises the early stages of knowledge gathering, feasibility studying, and requirement specification. Further, the development of ontologies for semantic web application presents a significantly different situation to that which was present before the relatively recent growth of internet connected devices, cloud computing, and service-oriented architectures. This renders many of the well-established, more traditional methodologies unsuitable when faced with this modern challenge. Hence, whilst many of the recommendations and best practices of approaches such as Uschold [26–28] still stand, specific activities are best guided by recent works. Methontology [29] has been well regarded for some time [30], but predated NeOn significantly, which itself is now 8 years old, although is still active. A more recent work, [31], specifically addresses the development of ontologies for the semantic website, although still describes the field as immature. This work utilises many of the same key themes of competency questions, iteration, and many supporting activities around the core ontology implementation phase. Further, [31] identifies the similarities between ontology engineering and traditional object-oriented programming model development, as well as the differences, and the need for modern approaches to balance the two.

Regarding automated ontology generation, the main stages once a corpus has been established are concept extraction, taxonomy extraction, and non-taxonomical extraction [32]. Several methods of automated ontology extraction have been proposed in the literature, such as statistical triple-based identification of noun–verb–noun triples [33], WordNet-based sense disambiguation [34], or a simpler lexico-syntactic pattern identification [35]. Of these, the WordNet-based approach is especially interesting, but the approach observed arguably does not leverage WordNet maximally to sort the extracted concepts in order of relevance to the domain. The approach utilised hence extends this WordNet-based method of concept and relationship extraction based on the definitions offered in WordNet, as described later.

Given the lack of semantic web relevance of earlier methods, and the lack of adoption of very recent methods, the NeOn method was chosen primarily to guide the development of the ontology. The core principles were hence adopted from the NeOn methodology, as well as many of its specific recommendations, but it was adapted to account for developments in the field since its initial publication, including the growth of the internet of things, service-oriented architectures, and automated ontology extraction. This is described in the following section.

3 Overview of approach

The approach taken is now described, first in terms of the methodology used to produce a valid research finding, and secondly in terms of the technological work conducted.

3.1 Research methodology

The choice was made to adopt a pragmatic research philosophy, incorporating aspects of positivism alongside the interpretivism often used outside of natural science. This meant that the social aspects of requirements engineering were embraced as part of the research methodology.

The methodology could be most accurately described as participatory action research (PAR) [36] or case study research [37]. Action research has been described as an “inquiry that is done by or with insiders to an organization or community” [38], and PAR then specifies that researcher learning and participation is an explicit part of the research methodology. Case study research, by no means exclusive to PAR, is a form of inductive social science research [37, 39].

Yin states that a case study approach is suitable when asking “how” or “why” questions, provided the researcher has little control over the subjects’ behaviour, and the focus is contemporary rather than historical phenomena. The current work is indeed asking ‘how should requirements be elicited’ for this new field, it did not aim to control the behaviour of the practitioners (subjects), and the phenomenon being studied is contemporary.

In line with this methodological stance, Sect. 7 includes a defence of the validity of the research.

3.2 Technical methodology

Our requirements engineering approach, illustrated in Fig. 1, was inspired by the well-established NeOn methodology for ontology development through the reuse of existing semantic resources, within a broader platform-level requirements engineering process. The adaptations were for two main reasons: the growth of the internet of things since NeOn was created, and the need for the ontology

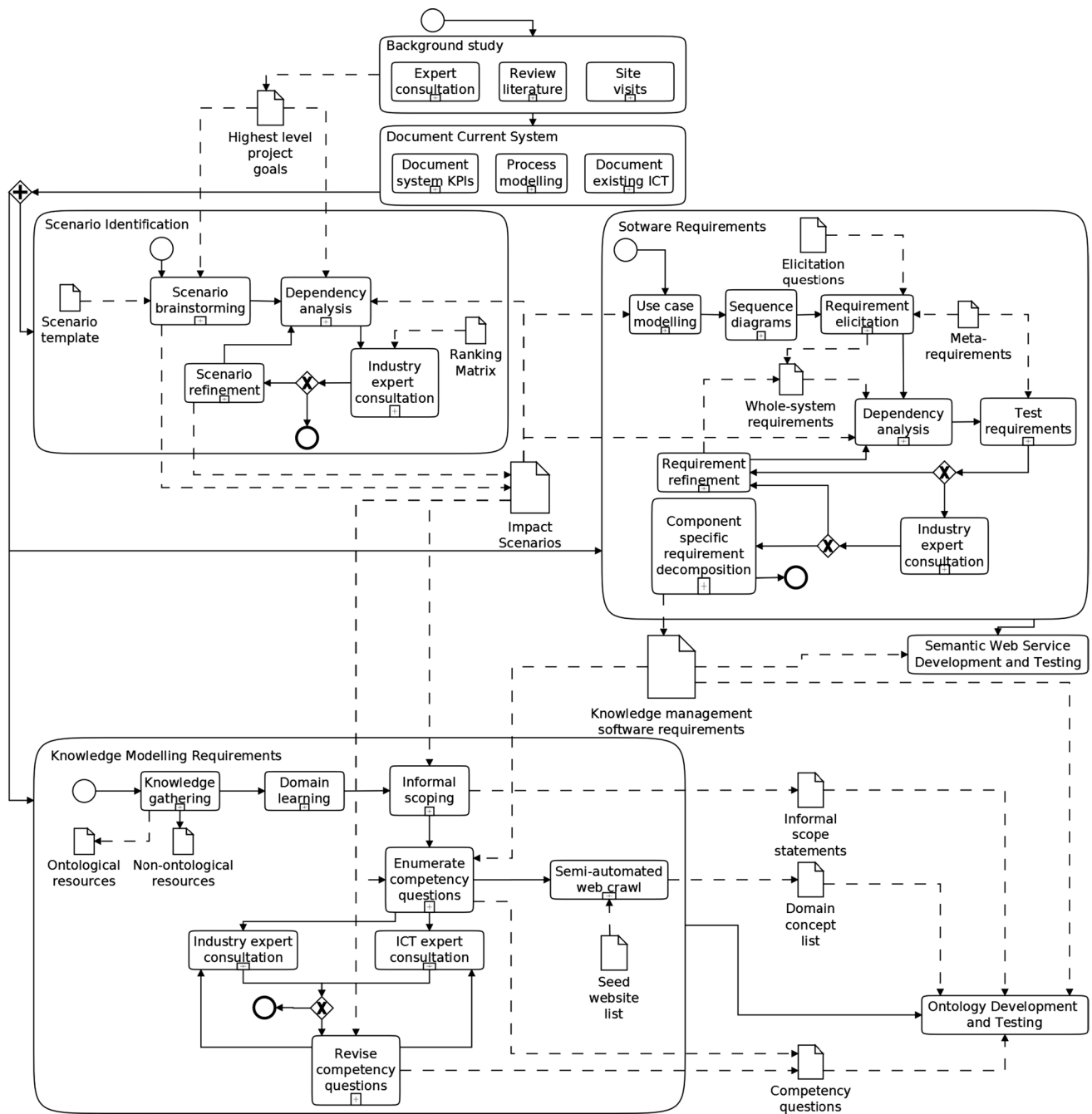


Fig. 1 Main knowledge management requirement engineering processes and artefacts

to have value outside of the target system as a benchmark in the field. This involved balancing the knowledge engineering objectives prioritised by NeOn with the software engineering objectives of the overall IoT project, and the softer requirements from the domain experts of fostering ownership and human intelligibility. This involved a significant knowledge gathering and requirements capture process, which itself consisted of a thorough methodology.

Regarding software requirements, the first stage was to gather knowledge about the domain, target systems, and intended value proposition of the overall software solution. Following this, formal modelling was conducted of the business processes involved in the target system, and scenarios for the use of ICT within these were developed. Next, an analysis and design process was followed to produce software requirements for the overall software solution,

through use case specifications and sequence diagrams. These requirements were then iterated alongside domain experts, and the previously developed scenarios, to ensure a comprehensive set of requirements was produced. Next, a system architecture was curated, and the requirements were decomposed into separate requirements for each component, producing a draft set of requirements for the ontology web service.

Requirements from a knowledge engineering perspective were iterated from the higher level software requirements, through further knowledge acquisition and scoping. First, a literature review was conducted of the semantic resources in the field, to give context to the model. Second, the software requirements were decomposed further into competency questions, conceptually orchestrated through the project's scenarios. These were then formed as a set of formal SPARQL queries which the ontology was required to answer, and which were adapted as the project matured and the role of the ontology service became clearer.

From the domain value perspective, the requirements development process focussed on the existing knowledge management systems in the domain, and existing semantic resources. From this knowledge, the value proposition of the ontology became clearer, by defining which potential benefits of ontological representations could be of value to the domain. From these informal definitions, the domain value requirements were developed semi-formally. These requirements were developed and resulted in softer concepts, which defined in part the development process itself of the ontology, as well as which semantic resources would be valuable to reuse, the systems which the ontology should interoperate, and how the ontology should be framed to domain experts.

Once the requirements had been developed, and subsequently iterated to produce a coherent set across the three perspectives, the development of the ontology and accompanying software was undertaken. The requirements were utilised throughout the process to test and guide the development at each iteration and at each contact point with experts from the ICT or target domains. Finally, the formal requirements were utilised as a litmus test to assist with the ontology's validation and the testing of the ontology web service.

4 Platform impact scenario identification and user requirements

The first milestone of the requirements engineering process was to produce platform-level impact scenarios. These described the various impact pathways for the software within the existing business processes and software frameworks present in industry. From these scenarios, project (or whole-system) level requirements were elicited, from which the knowledge modelling requirements were implied. These

were formalised through system analysis and design, in an iterative process with computer science experts. The development of impact scenarios began with informal knowledge gathering through expert consultation, literature review, site visits, and analysis of the existing products and processes at the client organisations.

One of the key challenges was to ensure the solution developed was general enough. A methodology was, therefore, adopted which captured the stakeholder-orientated socio-technical and business requirements. This methodology consisted of four high-level stages, with each stage being broken down into a series of tasks. These stages were conducted in close collaboration with the industrial stakeholders, to foster early engagement with the developed artefacts, 'buy-in' of domain experts, and genuine business and industry value of the project outputs. Figure 2 illustrates the four stages of the requirements capture process, which are now described in more detail in the following sections.

4.1 Business process modelling

The first stage of the requirements capture process involved achieving a high-level understanding of the structure and the processes involved in the water value chain from industrial experts. To achieve this, the first stage was broken down into two tasks: (a) documenting water processes using the IDEF0 [6] functional modelling methodology and (b) the analysis of network topology specifications.

To produce IDEF0 models for each pilot, the system within each pilot location was analysed and the following tasks were performed:

1. Document the high-level processes that the water goes through within the system.
2. For each process, identify the inputs and outputs.

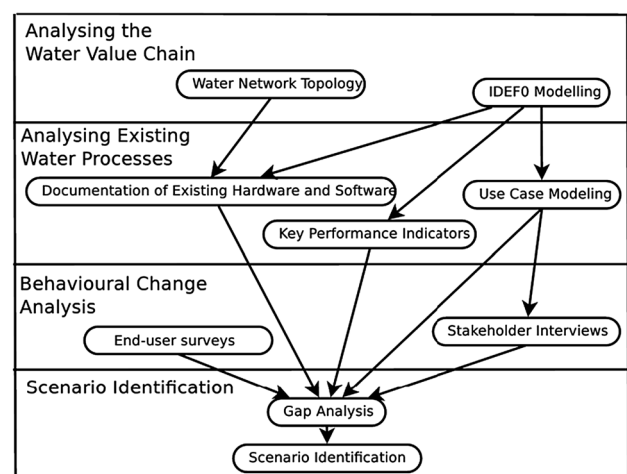


Fig. 2 The scenario identification process

3. For each process, identify the constraints and mechanisms.
4. Each process in this model should be broken down and the IDEF0 modelling process repeated for each sub-process.

An example IDEF0 model is presented in Fig. 3, for the processes of water treatment.

4.2 Analysis of existing water systems and processes

The second stage of the requirements capture methodology builds on the understanding of the client's water processes and topology. This stage consists of three tasks: (a) documentation of existing hardware and software used within the pilot (b) documentation of key performance indicators and (c) UML (Universal Modelling Language) Use Case Modelling [2].

The first task involved identifying further information for each of the mechanisms identified as part of the IDEF0 modelling. To achieve this, a template was completed by industry experts, describing information such as name, type, data storage technology, and file format.

The second task involved specifying in more detail the key performance indicators for the various processes within the system, the majority of which had been identified as constraints during the previous stage of the process. The final task in the second stage was to understand the interactions of actors with the water system. To achieve this, a series of use case modelling exercises were conducted. The IDEF0 models were analysed and all actors that featured as

mechanisms were used as a starting point for generating use cases. These described in a standard notation the interactions of individuals with the target system.

4.3 Scenario identification

The process of scenario identification was undertaken by engaging each client in a facilitated workshop, then developing these ideas and refining the produced set of scenarios through collaboration with industry experts. To facilitate ideas in the initial workshop, four categories were chosen to focus the process, based on the described overarching aims of the system. These were behavioural change, energy reduction, business process improvement, and supply chain water loss reduction.

For each scenario the following fields were populated: name, description, objectives, artefacts to be developed, input data, existing technologies to utilise, output data, actors (during demonstration and at other times), times applicable, and anticipated impact.

Once generated, scenarios were matured, collaboratively revised, and iterated until a final set of scenarios, sufficiently covering all targeted aspects of the water value chain, was identified. This iteration involved a detailed ranking matrix which weighted stakeholder responses to various aspects of the scenarios based on their perceived reliability in evaluating each factor. In addition, a dependency analysis was conducted whereby the high-level goals of the system were stated, and these were mapped to each of the detailed scenarios. This enabled accountability and logical consistency, and empowered a comparison between scenarios based on their interactions towards delivering the high-level goals. An

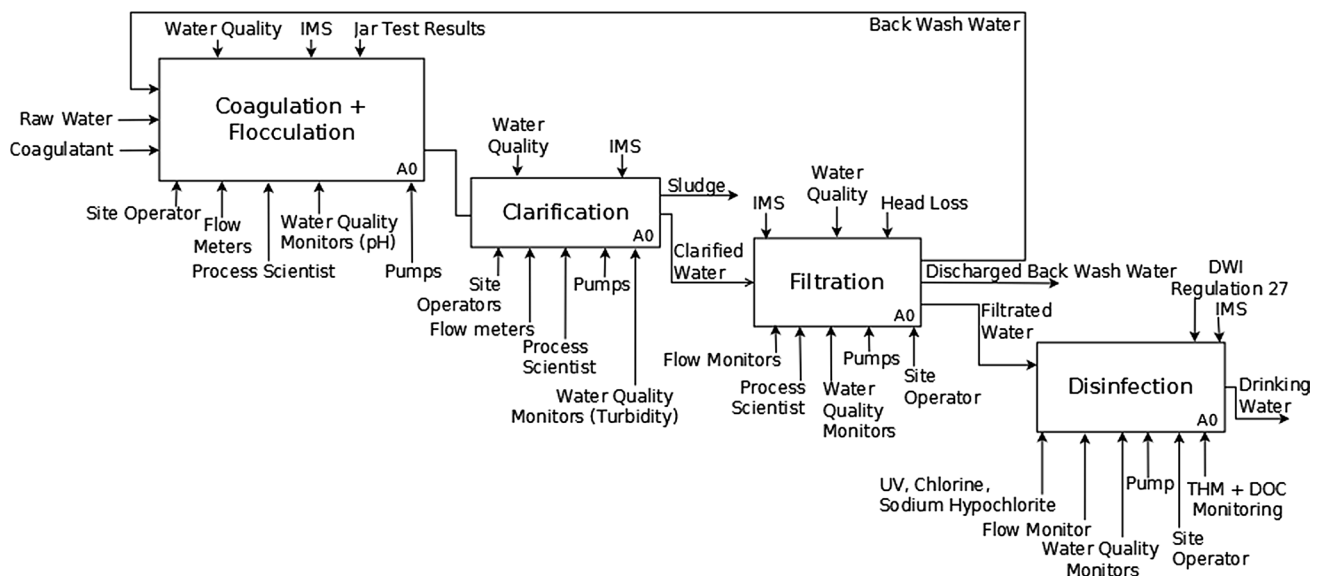


Fig. 3 Example business process model for eliciting user requirements

excerpt of these goals and the mapping are shown in Table 2 and Fig. 4, respectively, where the black cells indicate that the scenario delivers on the system goal, and grey cells indicate some contribution towards the goal.

The scenarios provided a critical component in successfully targeting and defining the requirements of the ICT system. The scenarios represented user requirements, as they primarily originated from user elicitation, and used language which was not technological from a software developer’s perspective.

5 System requirements from impact scenario decomposition

The described impact scenarios served as a guiding set of initial intentions, which represented a project decomposition and description task. These effectively elaborated on the broad goals of the ICT system in a detailed and formal manner. Following this, the next milestone was the

development of software requirements at the whole-platform level. A common system analysis and design methodology was adopted whereby the software solution was considered as a ‘black box’ with which users interact. By utilising this black box approach, the requirements specification process consisted of the following steps:

1. Meta-requirements: This step was undertaken by describing a series of meta-requirements. These meta-requirements define the guiding principles by which the quality of the solution’s requirements can be verified.
2. Requirement elicitation: This step involved the elicitation and description of the functional and non-functional requirements of the target system.
3. Iterative refinement: This step consisted of the iterative improvement and refinement of the requirements. Specifically this step compared the generated requirements to the meta-requirements defined within Step 1.
4. Revision and end-user validation: This final step involved the external review and Quality Assurance

Table 2 Example stakeholder-oriented systemic ‘goals’, indicating high-level requirements

Goal ID	Short name	Description
G1	Water reduction	Quantifiable and significant reduction of water consumption
G2	Demand stabilisation	Peak-period reduction of water and energy distribution loads
G14	Web app	Online decision support and visualisation environment
G24	System feedback	Report the current state of the system to users, including their direct impact

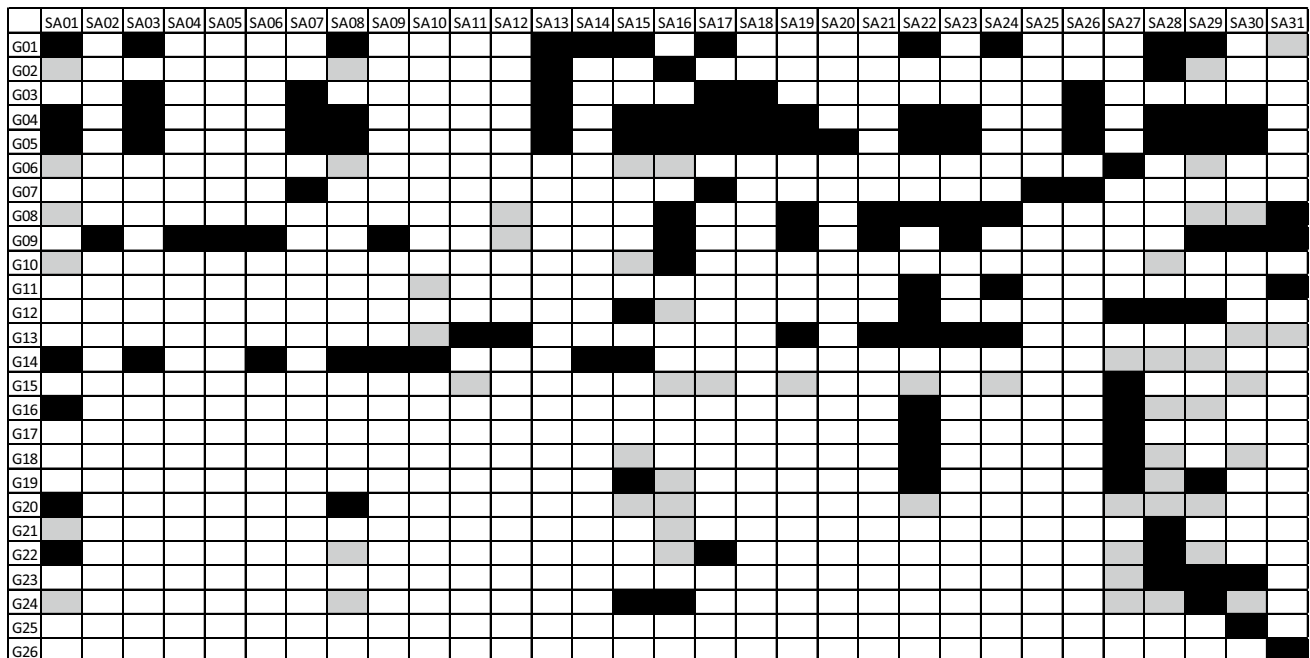


Fig. 4 Mapping indicating relevance of candidate scenarios for each overall goal

(QA) of requirements by the stakeholders within the water chain to mitigate bias.

As mentioned previously, this approach was inherently multi-disciplinary due to the multi-disciplinary origins of the scenarios, and this was reinforced by the varied team which reviewed the process. The remainder of this section will now describe each of these steps in further detail.

5.1 Meta-requirements

The main goal of the first stage of the requirement specification process was to document a set of meta-requirements that enabled the knowledge of when the requirement specification was acceptable. This question relates to the quality of the generated requirements themselves in terms of their testability and clarity. Taking note of existing requirement engineering meta-models [41, 42], these meta-requirements were presented through a list of statements, such as the following:

- A requirement must describe clearly what is required of the software solution such that it can facilitate the achievement of each scenario's goals.
- A requirement must contain sufficient details of the functionality and performance of the stated functionality so that it can effectively inform the detailed system design.
- A requirement must be explicitly described including any constraints which must be met for the solution to be deemed acceptable, as well as any which are 'preferred' or 'optional'.
- A requirement must utilise appropriate language such that requirements are testable.

5.2 Requirement elicitation

The software requirement elicitation process involved inferring the requirements of the final solution based on previous specifications, scenario descriptions and end-user perspectives. This process was conducted using a hybrid top-down and bottom-up approach. These top-down elements of the approach were necessary because the majority of the boundaries of the solution's scope were previously established as part of scenario descriptions, which constitutes a top-down approach. However, these scenarios also need to be considered alongside the end-user and data-oriented perspectives, mandating significant elements of a bottom-up approach. Figure 5 illustrates this process, showing that the overall requirements specification was derived from a number of sources including: the end-users of the software solution (the water network operators and water consumers), the scenario descriptions, external parties and consideration of data-orientated perspectives.

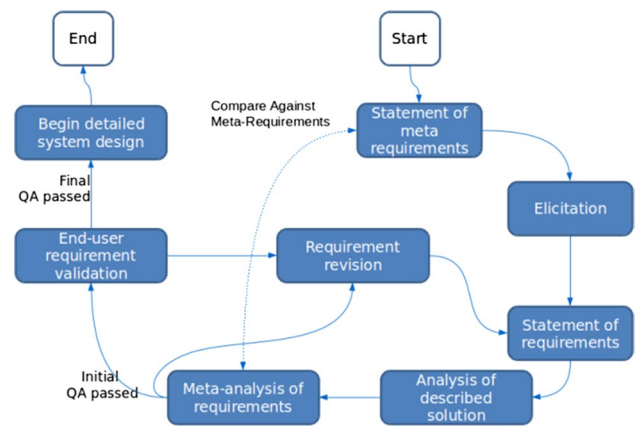


Fig. 5 Requirement elicitation processes

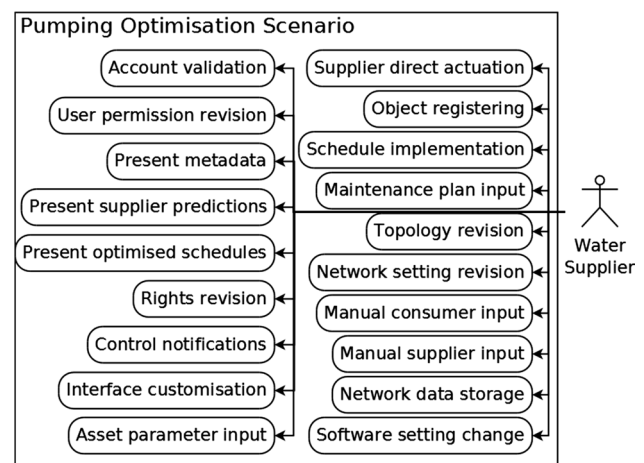


Fig. 6 Example use case diagram for eliciting user requirements

As can be seen from Fig. 5, the majority of the high-level system requirements were defined using a top-down approach, driven by analysing the views of the solution's users, previous documentation, and external parties views to gain a complete picture of the system. However, a consideration of the technologies and data already available within the water-value chain added extra information and provided a pathway to the practical implementation of the system.

The starting point for this process was the previously described knowledge gathering, process modelling, and impact scenarios. Extracting software requirements from these involved an iterative brainstorming process. Use case diagrams were developed, such as shown in Fig. 6. These diagrams were then compared and considered alongside each scenario's goals and subsequently revised through the analysis and refinement process described in the next section. The involvement of the client and end users in this iterative process was especially important in order to consider their views on what functionality is expected, how they would

expect to conduct each scenario and the likely interactions between the user and ICT solution.

Another aspect that was considered during the elicitation of requirements was the contextualisation of the solution within the wider ‘future cities’, ‘sustainable development’ and ‘smart resource management’ fields. This helped to ensure that the solution represented an extension to the state-of-the-art. This consisted of a thorough literature review and further consultation of the existing body of work, as well as a consideration of the legislation the solution must comply with to be implemented.

Finally, a set of non-functional requirement elicitation questions established by Michigan State University [43] was used to assist with gathering additional non-functional requirements. This approach considered aspects related to the quality of the functions delivered, across each of the perspectives already identified.

The end result of this process was a set of definitive statements of requirements. These requirements were varied in terms of terminology, depth of specification and compliance with the meta-requirements, as they were the result of an organic and multi-perspective elicitation process. These initial requirements were then unified and homogenised by abstracting them from their scenario specific contexts and considering the entire set as a description of the overall system. This enabled the initial validation and improvement of the requirements, as omissions, duplications, ambiguities and variations in terminologies were exposed. Again, whilst this process mainly focussed on the functional requirements, the results of the non-functional requirements elicitation questions were subject to a similar abstraction and distillation process to ensure they met the meta-requirements stated.

5.3 Analysis and iterative refinement process

Following the elicitation and gathering of initial requirement statements, it was necessary to thoroughly analyse these and subsequently revise them until the meta-requirements were adequately met. This process involved considering the requirements within the contexts of the individual scenarios as well as the overall project and the field.

Immediately following the initial specification of the required functionality through the use case diagrams, it was deemed necessary to describe the sequential nature of some of the user interactions that were implied in the scenarios. This enabled a better understanding of the human–machine interactions within each scenario. Whilst the scenarios featured varied user interactions, some were constrained into typical usage patterns. The developed sequence diagrams hence indicated example sequences of processes which typify the scenario in question using generic internal component names as they were likely to manifest in the final solution. These highlighted the required functions and likely

inputs and outputs of each, facilitating the refinement of the initial list.

The refinement of the requirements was especially important as, directly following the elicitation, they used varying terminologies and exhibited significant redundancy. The process of iterative refinement was completed to robustly analyse whether the solution described was sufficient to enable the scenarios to accomplish their respective goals. To this end, the final list of functional requirements was analysed by comparison against the scenarios. This process allowed some confidence that no requirements had been missed during the process of requirement elicitation. Gaps in the requirement specification were highlighted and the requirements were hence revised.

The main tool that used to facilitate this analysis process was an explicit mapping between the 13 discreet scenarios and the functional requirements. This maintained the train of logic from high-level goals to scenarios to requirements, which was discussed regarding Fig. 4, and provided an overview of the interactions and dependencies between scenarios.

Once the solution described by the requirements was deemed to meet the intended outcome of the solution, the requirements themselves were checked for quality, completeness and testability against the meta-requirements. These meta-requirements were used to guide the process of developing and writing the requirements and were used as a checklist to bound the solution space. Where the requirements failed to meet one or more of the meta-requirements they were revised and both levels of quality assurance checking were conducted again. The final stage of validating the requirement specification was then to pass the proposed requirements to the end users and the software developers to ensure their acceptance of the proposed requirements.

5.4 Decomposition to ontology service software requirements

Once the system-level requirements were deemed sufficient, these were decomposed further into component-level requirements, including the knowledge management software service, which was based on an ontology. Hence, beyond scoping the ontology as a knowledge modelling task, it was critical to consider the end use of the ontology within the solution as an ICT component throughout the development stages of the task. Therefore, beyond the scope definition and competency questions, the software requirements were taken into account in designing the ontology as well as developing the software which deployed it. These requirements arose from the system analysis and design process detailed previously; by analysing the use cases and scenarios to elicit the main functions required by the knowledge management service of the platform. Example excerpts of the

initial elicitation and final software requirements are presented in Tables 3 and 4.

6 Knowledge modelling requirement specification

Producing the requirements of the ontology as a knowledge modelling artefact was a critical priority. As ontologies aim to progress towards domain consensus, it is preferable that they meet not only the system specific objectives, but balance this need with the goal of achieving an agreeable, complete, and sufficient representation of the domain. This is beneficial in the emerging field of urban cybernetics, as the initially intended system is highly likely to evolve to be integrated with external systems, new system-level functionality, and the ontology itself may be reused elsewhere, so foresight of this within the requirements engineering is highly beneficial. Towards this, the close involvement of industrial partners and varied stakeholders in developing the competency questions and intended scope of the ontology

was prioritised. Again, this was conducted iteratively whilst refining the software-level requirements towards a cohesive and complete set of requirements. The stages and results of this process: domain learning, competency question setting and semi-automated web-based feature extraction, are now elaborated in turn before reflecting on the effect of the need for domain consensus on the requirements.

6.1 Domain learning and ontology reuse

The first stage of all ontology engineering methodologies encountered was knowledge gathering and less formal scoping. Within the current case study, this was largely accomplished through the aforementioned ‘whole-system’ knowledge gathering and requirement development. However, further consultation was conducted with domain experts by an ontological expert in order to frame the domain perspective in an ontology engineering manner and to begin the conceptualisation of the domain.

The first phase of the semantic modelling activities was to thoroughly understand the challenge faced, followed by

Table 3 Example output of the requirements elicitation process of informal scenario-based requirements

Requirement group	Requirement sub-group	Elicitation question	Behaviour and feedback scenario
Functional requirements	Functionality	What will the system do?	See use case diagram and scenario description
		Are there several modes of operation?	Idle mode, where the system checks for significant water usage and recommends changes if needed. Active mode, where the application is operated by users for evaluating behaviours and neighbourhood data
	Data	What should the format of the input and output data be?	The format of input data will be dependant on the data sources and the sensors used, while the output data should be intuitive and accurate to the end-users
	Users	Who will use the system?	Primarily end-users and possibly water utility staff
Quality requirements	Performance	Are there constraints on execution speed, response time, or throughput?	Viewing pertinent data should be performed immediately. The response time for evaluating a set of behaviours should be very quick while giving the recommended changes of behaviours could be within minutes
	Usability	How easy should it be for a user to understand and use the system?	Very simple. Menus should be intuitively structured with the functionalities introduced to the users

Table 4 Example system requirements produced

Title	Brief description	Priority
Virtual representation of water network	The water ontology should provide a detailed virtual representation of the current state of the water network. Including artefacts and properties of all physical components	Mandatory
Querying ontology	Should provide an effective query based approach for accessing data within the ontology	Mandatory
Managing ontology model	A system administrator should be able to update an existing ontology model or to replace it with a new version, without affecting the overall system integrity	Desirable

significant domain research and knowledge acquisition, and finally the production of formal requirement specifications for both the ontology and the overall ontology service. These requirement specifications are living documents and continue to naturally evolve as the ICT system is maintained and matured and requirements are added, removed or refined. As well as the previously described software analysis and design processes and artefacts, a collection of relevant ontological and non-ontological models were collected and evaluated for reuse. Once the specifications and assorted data objects were analysed and validated successfully, the ontology was conceptualised and the reusable resources were merged and extended into a domain-independent meta-model.

One output of this stage was a concise set of statements about what the ontology should be, which served as a nucleus for later formal requirement statements. These included the following statements of what the ontology should be:

- A model of the entire water network, including topology
- A model of the actuation and instrumentation layer deployed in the water network (pumps, valves, different types of sensors) and their relationship to water network topology
- Realised on top of a semantic data storage technology, e.g. RDF store
- Should provide efficient semantic query interfaces for the ontology, e.g. via query wrappers developed on top of SPARQL
- Should be based on existing ontologies such as the semantic sensor network (SSN) ontology and emerging efforts such as SWIM.

6.2 Scoping and competency questions

The scoping and specification of an ontology is a crucial stage, as the ontology must be detailed and accurate enough for the foreseen queries, utilize sufficient abstraction and breadth for potential future reuse, yet be concise enough to meet the performance requirements of its intended application. The scoping was hence conducted alongside the process modelling and scenario modelling, the software requirement specifications, and domain expert consultations, through ontology expert formalisation of these preliminary activities into competency questions. This section therefore summarises the requirements of the semantic models from a knowledge modelling perspective and hence, competency questions are produced to test the vocabulary's 'planned functionality'.

These questions represent a range of queries the ontology should be able to answer directly or through inference, and serve as a 'litmus test' of whether the ontology delivers the planned functionality. This then serves as an initial

validation of the ontology, with further validation required into whether this 'planned functionality' is sufficient within the intended application, and whether the modeling choices are agreed upon amongst the ICT solution's stakeholders, and ideally beyond.

The target system aims to implement intelligent sensing and analytics to integrate the management of the water network across conceptual scales, such as the domestic and whole-system scales, whilst improving management schemes and domestic consumption profiles. From this statement, as well as the statements gathered from the domain learning stage, the knowledge domain can begin to be conceptualised and bounded. The ontology was hence required to formalise a description of the sensing in the water network, the network itself, the social entities involved, the domestic entities and the relationships between them. To further elaborate on the scope of the ontology, statements regarding what is not focused on in this ontology helped to clarify the boundaries of the target knowledge domain:

- Natural artefacts will be included, but their management is not a focal point
- Electricity consumption will be included, but its management is not a focal point
- Non-domestic consumers are not a focal point
- Managing the internal operation of treatment plants and pumping stations is not a focal point.

By understanding and acknowledging these boundaries, reusing the ontology in future applications is facilitated, as its role alongside other ontologies becomes clearer. For example, it could be aligned with a model of treatment plant concepts to enrich and integrate data between high-level system management and asset-level performance objectives.

The competency questions then utilized the scenario-driven approach and its subsequent outputs, by considering the main entities and their properties within each scenario. Questions were formed which elicited these properties. The opposite was also conducted in an organic manner; natural questions about the water value chain which were deemed likely were analysed for the property and entity they were referring to. Examples of various types of these are presented below:

Scenario 1 (Behaviour and Feedback):

How much water does person X consume per week, on average?

Property—average weekly water consumption.

Entity—domestic resident.

Which water meter is attached to house X?

Property—attached water meter.

Entity—domicile, domestic water meter.

Scenario 2 (Network monitoring):

What property does sensor X detect?

Property—observed property.

Entity—sensor.

Scenario 11 (Reservoir optimization):

What is the maximum storage volume of service reservoir X?

Property—max storage volume.

Entity—service reservoir.

Critically, the competency questions aimed to emulate the questions which other software components in the platform would be asking of the water value chain to meet their requirements and goals. This emulation was inherently an iterative process, such that the competency questions changed and adapted as the other software components became more mature, and so some interpretation was required in utilising the competency questions to guide the development of the ontology; that is to say, some foresight was required.

6.3 Semi-automated web crawl and feature extraction

The manual elicitation of domain knowledge and the scenario-based decomposition towards competency questions was coupled with a semi-automated web crawl and feature extraction process. The aim of this was to facilitate broader relevance of the ontology by aligning the terminology and semantics modelled with the wider water sector, by analysing web documents across the sector and ontological features from these, as a whole body of literature.

The full details of the semi-automated validation are beyond the scope of this paper, but it is briefly summarised here. The first stage consisted of automatically ‘crawling’ a manually selected list of relevant websites (and their linked websites) for public HTML (raw ‘screen text’), Microsoft Word, TXT and PDF documents based on loose rules for relevance, through a Python program written within the scope of the current work. This was based on the open source Scrapy library [44]. These were then processed to extract a list of all the words, and several metrics about them, per document, using the natural language toolkit library [45]. These data were then further processed through a Python script to identify the most relevant and likely candidate class and property names.

This first ranked the words by frequency and ‘term frequency, inverse document frequency’ (tf-idf, a common metric of the importance of a word in a document), filtered out ‘stop words’ such as ‘if’, ‘the’ and ‘is’, and names of companies, places, and people, then looked up the word in the WordNet lexical database and retrieved a definition of the word. The relevant words in this definition were then looked for in the list of words found in the crawled web documents, and their tf-idf values summed when they were found, to produce another metric of ‘importance’ for the

crawled document word in the target domain. This was then multiplied with each word’s tf-idf value to produce a hybrid measure of importance of the word, and the list was ordered by this metric of importance. Finally, the script utilised WordNet to separate the proper nouns (which would be instances of classes), nouns (which represent candidate classes) and adjectives and adverbs (candidate properties); other linguistic components were removed.

From the resulting data, the representative domain coverage of the ontology could be calculated when validating the ontology and possible missing classes and properties were identified. The outputs of this process at the requirements engineering stage were then iterated alongside the domain learning and competency question stating processes to achieve better domain coverage and relevance.

Within this process, the development and use of a WordNet relevancy analysis method in the manner chosen is a novel feature. This held the advantage beyond a simple tf-idf of ranking each word based not only on its occurrence within the corpus, but also based on its currently accepted definition(s). The computation of the overall relevancy metric for each word was a critical challenge of the approach, whereby the weighting of the tf-idf value and the WordNet relevancy value in determining the final ‘relevancy metric’ affected the resultant order of words. Further work could be conducted to determine the best approach to this challenge. In addition, as expected, the choice of seed websites affected the results, and so it was necessary to revise the list to include only those which described the domain from a more technical perspective, as the ontology aimed to capture the technical vocabulary of the domain.

Despite these challenges, the semi-automated process produced a set of object classes and relationships, ranked in order of relevancy to the domain. This was highly valuable in evaluating the relevancy of the domain ontology to the wider water sector, which helped to balance the requirement of producing a benchmark for the sector with producing a knowledge management artefact for this specific ICT system. Specifically, the outputs of this task assisted in evaluating against this requirement as the automated process was not affected by the domain perspectives of the project’s participants or agency, beyond the choice of seed website selection.

6.4 Requirements for progress towards domain consensus and standardisation

One ambitious goal was to contribute to the relatively new discourse in the water sector regarding semantic modelling and standardisation. Whilst a somewhat secondary goal, it was deemed worthwhile and feasible given the novelty of the concept in the water sector. Arguably this is also true in other smart city domains, such as smart government, smart food management, and smart mobility.

Towards this ambition, the best practices offered in the NeOn methodology regarding future reuse, abstractions, and intelligibility, were particularly taken into account. This involved prioritising the literature review of existing semantic resources in the field, and either reusing, or aligning in some way, the developed model. This, therefore, led to a small number of alignments which were deemed as requirements for the ontology developed. In addition, significant abstraction was stated as a requirement for the ontology, as this would allow its future alignment with upper ontologies or across to other domains with more ease. Intelligibility was also specified as a requirement, meaning that the ontology must make not only be logically consistent and valid, but somewhat intuitive for a trained person to understand. This soft requirement could be met by ensuring intuitive class hierarchies, avoiding very similar labelling of different entities, and excessive equivalence statements.

By achieving these goals, it was intended that the ontology would be more accessible, reusable, and modular, such that it could more easily contribute to the future development of a standardised semantic model for the domain.

7 Results and validation of approach

The multi-scale requirement engineering process adopted produced outputs at each stage, including business process diagrams, software use cases, sequence diagrams, scenario descriptions, meta-requirements, software requirements, competency questions, non-functional ontology and process requirements and an automated extraction of domain vocabulary. Examples of these have been presented throughout the paper. Following the development of the ontology and accompanying software, the requirements were tested against, and greatly benefitted, the evaluation of the ontology and the overall solution's acceptability against initial intentions. The requirements were validated through the process of developing and testing the artefacts within the case study. In addition to the ongoing experimental validation of the requirements throughout the process, they were validated through a number of specific workshops with domain and software experts, which aimed to evaluate the suitability of the ontology against the initial requirements, which served to evaluate the requirements' testability, and suitability for their intended purpose. In addition, the use of meta-requirements helped to validate the produced requirements considerably, by guiding the iteration process which they underwent.

The initial, automated check of the ontology's consistency through the built in Protégé reasoner has been consistently passed; the ontology does not contain contradictory statements. The competency questions can be answered by the ontology in its current form by 'asking' the questions as SPARQL queries. The domain expert validation

was conducted separately with the domain expert partners through 1-day workshops, and in both cases the ontology's modelling choices were broadly validated, the majority of the detailed modelling choices were validated and corroborated between workshops, and some revisions and extensions were suggested. An additional workshop with the WISDOM partners and special interest group experts was then also conducted, which served to validate that the changes made were sufficient and hence that the ontology was then sufficient. The domain ontology was tested for validity at a convening of industrial experts; the types of the 25 validating organisations are shown in Fig. 7.

The ontology was considered by a wide range of stakeholders in the water value chain at the validation workshop, most of whom had little bias towards the project. This offered a broad view on the ontology and hence tested its extent, as well as its detail in areas of the water value chain which the project partners are not experts in. Consensus was reached that the ontology represents a shared and sufficient conceptualization of the domain by this group, which represents a significant milestone in its validation, and of the requirements which supported its development and testing. Some of the comments from the expert validation session were:

1. The ontology addresses the problem of interacting between tools (such as GIS, SAP, and customer data).
2. Include alarms as well as sensors.
3. 'Governing body' is also called 'regulator'.
4. Include 'water testing company'.

These comments were all used to revise the ontology. The majority of comments were advisory or generic, such as regarding possible future work, rather than required changes in the scope currently addressed. Examples of these comments were

- The work could be considered as a type of enterprise service bus
- An ontology is also called a taxonomy

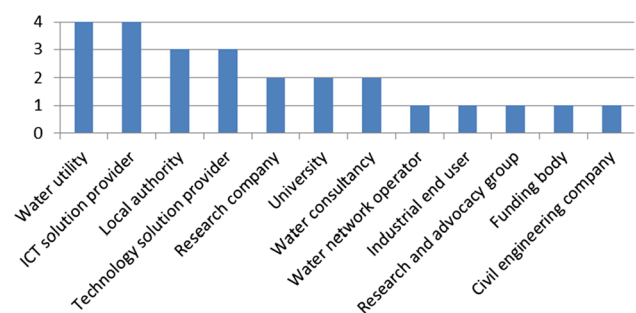


Fig. 7 Organisations involved in the validation process

- Sensors could also be ‘social sensors’, which report numbers of tweets, etc.
- Collaboration relationships exist between utilities which share a water resource

Table 5 presents example outcomes of the competency question testing, showing how the deployment sufficiently answers the questions when formalised as SPARQL queries, where the queries were answered in circa 15 ms.

From a research validity perspective, neither the field of participatory action research, nor the field of case study research, have a set of accepted guidelines for assessing research validity [38, 39], although some key characteristics of ‘good’ research in these fields include [39, 40]:

- Beginning with statements of the research philosophy.
- Maintaining good process descriptions and documentation of outputs.
- Use of complementary interpretation techniques such as triangulation.
- The production of a “parsimonious, testable, and logically coherent theory”.

A defence towards each of these criteria is summarised in Table 6.

Table 5 Example competency question testing evidence

Natural language question
What is Pipe X’s material?
SPARQL query
PREFIX wis: < https://www.WISDOM.org/WISDOMontology# >
PREFIX rdf: < https://www.w3.org/1999/02/22-rdf-syntax-ns# >
PREFIX dul: < https://www.loa-cnr.it/ontologies/DUL.owl# >
SELECT ?material
WHERE {
wis:Pipe_01 rdf:type wis:Main
wis:Pipe_01 wis:hasMaterial ?material}
Output (CSV format)
material
wis:PVC

Table 6 Defence of the research approach’s validity

Criteria	Defence
Statement of research philosophy	A pragmatist stance was adopted, utilising aspects of positivism in a broadly interpretivist approach
Good documentation	Many diagrams and documents were produced, totalling 394 delivered pages on the requirements capture methodology and outcome
Use of complementary techniques	As well as a pure action-research approach, several surveys were conducted to answer “what” questions, and empirical enquiries were conducted into the suitability of alternative technologies at several points. For brevity, these aspects are not reported here
Production of a parsimonious, testable, and logically coherent theory	The resultant theory can be simplified to the following, which meets these criteria: ‘Developing smart water ontologies following a requirements elicitation approach which is jointly owned by domain experts, ontology experts, and software engineers, leads to artefacts which provide more value over their lifetimes’

8 Discussion

The paper has presented a case study of requirements engineering for an ontology and accompanying web service for use in a smart water internet of things platform. The methodology adapted the popular NeOn methodology to be more suitable for the emerging case of urban IoT-based cybernetics. The key contribution is, therefore, an approach which balances the need in such projects for logical accuracy, software performance, and domain expert buy-in. The approach could be reused as a template in other urban ontological cybernetic cases, as the role of ontologies across smart cities has been widely noted.

One of the main benefits of the approach is balancing software requirements with domain specific requirements: computer experts must be able to develop applications from the model easily, but domain experts must clearly understand the terminology and cope with the abstraction and reuse of accepted ontological modelling practices. This benefit stems from the formal, regular, multi-stakeholder engagements in the process, and its iterative nature. By prioritising collaboration, and early domain expert engagement in the requirement setting, industrial experts gain more of a sense of ownership of the artefact.

As ontologies are an emerging field, especially within smart water, the soft aspects of the process are also due significant consideration. Requirement engineering for ontologies in fields where they are novel requires a careful balance of semantic accuracy and domain relevance. If the approach prioritises the ontological aspects of the task at the exclusion of the others, it may result in an artefact which is clear to semantic experts and which is logically optimal, but which appears far removed from the actual language of domain experts. This results in the model only being relevant within the originally intended setting, with less likelihood for reuse and adoption.

In the general case, ontologies created through a robust requirements gathering process can provide significant advantages. Two primary advantages are; (a) the presence

of a single vocabulary that can be used by systems to describe data in a given domain, and (b) the increased integration between IT systems that can be achieved using this defined semantic vocabulary. In fields where the use of technology is currently expanding this is especially important, as the rapid addition of different technologies leads to situations where an organisation possesses a variety of non-interoperable software systems. Furthermore, lack of a defined vocabulary for data within an organisation also hinders the adoption of new technologies due to incompatibilities between software systems and the schemas they use to store their data.

In the specific context of the smart water case study described in this paper, the presence of a semantic vocabulary (in the form of an ontology) allowed interoperability of systems used by the water network operator that previously required human intervention to provide such interoperability. Using this new ability to link their systems together, new functionality was deployed in the form of a proof of concept prototype that leveraged on data interoperability between sensor, Geographical Information System (GIS), customer and engineer management systems. Using this newfound interoperability, this provided new automation within the water network operator's systems. This included the following automated actions that were taken when water leakage was detected; (a) identification of geographical location of the leak, (b) identification of effected network segments, (c) calculation of the number of consumers effected and identification of any vulnerable consumers (i.e. health issues) that are involved and (d) identification of closest available engineer.

The approach was validated through the development and testing of an ontological model, accompanying web service, and wider software solution. By iteratively engaging with stakeholders, and validating the outputs within a wider group of industry experts, the closeness of the product to the initial intention and to the industry needs was sufficiently evidenced. By achieving these goals, it was deemed that the requirements were of a sufficiently high standard, as they were used to guide and test the artefacts towards that end state. The resulting ontology met all of the project requirements, was aligned with many well-regarded and emerging semantic resources in the sector, and is being pursued as a valuable outcome beyond the life of the project. It is essential, however, to acknowledge that whilst the ontology is deemed suitable, it is a living resource subject to continuous changes and adaptations. This, as a common feature in model lifecycles, reflects best practice in order to maintain the relevance of the artefact and to support progress to domain consensus whilst perspectives evolve over time.

9 Conclusion

The main contribution presented is a reusable ontology requirements engineering approach, tailored and tested for the increasingly relevant and important convergence of IoT and urban cybernetics. This was framed within a software research and development process for a smart water Internet of Things platform. The approach aimed to balance an emphasis at the requirement stage on the knowledge engineering, software engineering, and domain relevance aspects of the ontology. The requirements produced were used to develop a highly successful ontology and its accompanying software, which met the initial intention, industry needs, and academic gap well. The approach could be reused in similar tasks and neighbouring fields where smart Internet of Things approaches are growing in interest to promote robust, comprehensive and balanced requirements.

Acknowledgements The authors would like to acknowledge the financial support of the European Commission under the FP7 WISDOM (Grant number 619795) project, as well as the financial support of EPSRC (Engineering and Physical Sciences Research Council), and BRE (Building Research Establishment).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. CTRL+SWAN (2016) Cloud Technologies & Real time monitoring + Smart Water Network (AG126) | EIP Water. https://www.eip-water.eu/CTRL_SWAN. Accessed 18 Feb 2016
2. SWAN What are smart water networks. <https://www.swan-forum.com/swan-tools/a-layered-view/>. Accessed 19 July 2020
3. EIP Water AugMent (2016) Water Monitoring for Decision Support (AG124) | EIP Water. <https://www.eip-water.eu/AugMent>. Accessed 18 Feb 2016
4. IEEE Standards Committee, IEEE Standards Coordinating Committee 21 on Fuel Cells P Dispersed Generation and Energy Storage, Institute of Electrical and Electronics Engineers, IEEE-SA Standards Board (2011) IEEE guide for smart grid interoperability of energy technology and information technology operation with the electric power system (EPS), end-use applications and loads. Institute of Electrical and Electronics Engineers, New York
5. ICT4Water (2015) ICT4Water standards report
6. IDEF (2016) Integrated DEFINition Methods (IDEF). <https://www.idef.com/>. Accessed 1 June 2016
7. ISO/IEC JTC1 (2014) Smart Cities Preliminary Report

8. Rosario U-S, Biplav S, Robert JS (2011) Building a highly consumable semantic model for smarter cities
9. CityPulse citypulse_d3.1: Semantic Data Stream Annotation for Automated Framework
10. Anand N, Yang M, van Duin JHR, Tavasszy L (2012) GenCLOn: an ontology for city logistics. *Expert Syst Appl* 39:11944–11960. <https://doi.org/10.1016/j.eswa.2012.03.068>
11. Nemirovski G, Nolle A, Sicilia Á et al (2013) Data integration driven ontology design, case study smart city. In: International Conference on Web Intelligence, Mining and Semantics. ACM, p 43
12. Costanzo A, Faro A, Giordano D, Spampinato C (2016) An ontological ubiquitous city information platform provided with Cyber-Physical-Social-Systems. In: 2016 13th IEEE annual consumer communications and networking conference (CCNC). IEEE, pp 137–144
13. Fox MS (2014) Foundation ontologies requirements for global city indicators. In: Workshop program of the twenty-eighth AAAI conference on artificial intelligence
14. Soroosh N, Mark SF (2014) An Ontology for open 311 Data
15. Gröger G, Plümer L (2012) CityGML—interoperable semantic 3D city models. *ISPRS J Photogramm Remote Sens* 71:12–33. <https://doi.org/10.1016/j.isprsjprs.2012.04.004>
16. BSI (2014) Smart city concept model-guide to establishing a model for data interoperability
17. WatERP (2013) D1.3 Generic ontology for water supply distribution chain
18. Laurie R (2014) SWIM—a semantic ontology for interoperability
19. CUAHSI CUAHSI Hydrologic Information System—Hydrologic Ontology for Discovery. <https://his.cuahsi.org/ontologyfiles.html>. Accessed 23 Oct 2015
20. SWEET SWEET Overview. <https://sweet.jpl.nasa.gov/>. Accessed 23 Oct 2015
21. Vilches-Blázquez LM, Ramos JA, López-Pellicer FJ et al (2009) hydrOntology: a global ontology of the hydrographical domain. <https://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/107-hydrontology>. Accessed 23 Oct 2015
22. INSPIRE Thematic Working Group Utility and Government Services (2013) D2.8.III.6 data specification on utility and government services—technical guidelines. European Commission Joint Research Centre
23. OGC WaterML2 (2016) A global standard for hydrological time series. <https://www.waterml2.org/>. Accessed 18 Jan 2016
24. D’Avanzo E, Lieto A, Kuflik T (2008) Manually vs semiautomatic domain specific ontology building
25. Pérez AG, de Figueroa Baonza MCS, Villazón B (2008) Neon methodology for building ontology networks: ontology specification. *Methodology* 1–18. <http://www.macs.hw.ac.uk/~yjc32/project/ref-ontology/ref-ontology%20building/1-2008-Gomez%20Per ez-NeOn-Methodology-OntologySpecificationv3.pdf>
26. Uschold M, Gruninger M (1996) Ontologies: principles, methods and applications. *Knowl Eng Rev* 11:93–136
27. Gruninger M, Fox MS (1995) Methodology for the design and evaluation of ontologies
28. Schreiber G, Wielinga B, Jansweijer W et al (1995) The KACTUS view on the “O”word. In: IJCAI Workshop Basic Ontol. Issues Knowl. Shar. Citeseer, pp 159–168
29. Mariano F, Asuncion G-P, Natalia J (1997) Methontology: from ontological art towards ontological engineering. Universidad Politecnica de Madrid, Madrid
30. Fernández-López M (1999) Overview of methodologies for building ontologies
31. Ali S, Khusro S, Chang H-T (2016) POEM: practical ontology engineering model for semantic web ontologies. *Cogent Eng* 3:1193959. <https://doi.org/10.1080/23311916.2016.1193959>
32. Janardhana R Punuru machine learning techniques for automatic ontology extraction from domain texts
33. Punuru J, Chen P (2007) Extraction of non-hierarchical relations from domain texts. In: *Comput. Intell. Data Min. 2007 CIDM 2007 IEEE Symp. On. IEEE*, pp 444–449
34. Punuru J, Chen J (2012) Learning taxonomical relations from domain texts using WordNet and word sense disambiguation. In: *Granul. Comput. GrC 2012 IEEE Int. Conf. On. IEEE*, pp 382–387
35. Doing-Harris K, Livnat Y, Meystre S (2015) Automated concept and relationship extraction for the semi-automated ontology management (SEAM) system. *J Biomed Semant.* <https://doi.org/10.1186/s13326-015-0011-7>
36. Giovanni M, Brian D, Markus H (2014) Systematic problem formulation in action design research: the case of smart cities. In: presented at the ECIS
37. Yin RK (2014) Case study research: design and methods, 5th edn. SAGE Publications Inc, Thousand Oaks
38. Herr K, Anderson GL (2014) The action research dissertation: a guide for students and faculty. Sage Publications, Thousand Oaks
39. Eisenhardt KM (1989) Building theories from case study research. *Acad Manag Rev* 14(4):532–550
40. Williams PA (2006) Making research real: is action research a suitable methodology for medical information security investigations? In: Australian information security management conference, p 82
41. Deshmukh N, Wadhwa S (2007) A meta model for iterative development of requirements leveraging. *IEEE*, pp 343–349
42. Machado RJ, Ramos I, Fernandes JM (2005) Specification of requirements models. In: *Eng. Manag. Softw. Requir.* Springer, New York, pp 47–68
43. Michigan State University Capturing the Requirements
44. Scrapy Scrapy | A Fast and Powerful Scraping and Web Crawling Framework. <https://scrapy.org/>. Accessed 30 Aug 2016
45. Steven B, Edward L, Ewan K (2009) Natural language processing with python. O’Reilly Media Inc, California

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.