

smashHit Semantic Model

Technical Essay



October 2022



This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 871477

Executive Summary

The objective of smashHit is to assure trusted and secure sharing of data streams from both personal and industrial platforms, needed to build sectorial and cross-sectorial services, by establishing a Framework for processing of data owner consent and legal rules and effective contracting, as well as joint security and privacy preserving mechanisms. The vision of smashHit is to overcome obstacles in the rapidly growing Data Economy which is characterized by heterogeneous technical designs and proprietary implementations, locking business opportunities due to the inconsistent consent and legal rules among different data-sharing platforms actors and operators. The Framework provides methods and tools, such as the smashHit platform or Automatic Contracting tool.

The following key points refer to those main innovative features brought to the smashHit Framework by the developed smashHit Semantic Model:

1	Motivation	3
2	Designing Semantic Models	4
3	Application of Semantic Models.....	9
3.1	Standardised representation of consent and contract	9
3.2	Automatic Contracting Tool	11
3.3	Data Use Traceability	11
3.4	Context Sensitivity Solution	11
3.5	Consent registration templates	12
4	Conclusion.....	14
5	Glossary	15

If you got curious about how all that is made possible, just continue on the following pages, enjoy the reading, and please contact us with your feedback or questions!

- smashHit support email: info@smashhit.eu
- smashHit project website: <https://smashhit.eu>

1 Motivation

The smashHit semantic model is one of the core components of the smashHit project and facilitates communication between other smashHit components, offering numerous benefits. However, before diving into detail, let's first understand the problem.

While the problem definition can be presented formally (or technically), we decided to introduce it in general layman terminology so that even those who are not technical will also get a glimpse of what we are solving and why we are doing it this way. Let us say that you own a few big fields. It does not matter how many big fields, you can assume as many as you would like. The key point here is to have fields that a single person cannot manage. You then hire, say 10 people, with the help of recruitment companies. These hired 10 people are from different regions of the world and speak different languages and have different cultural and societal backgrounds. Even though, if we were to assume, that they speak some common languages, there exist different challenges. The first challenge is related to the semantics of the communication between you, the employer, and the employee. Since the employees are from different societal and cultural backgrounds, they might understand differently of things that you wanted to communicate. The second challenge is the cooperation between the employees among each other. There are the well-known multilingual environments problems.

Now, at this point, you might be wondering, what has this thing to do with software or semantic model? What if we tell you, it is not only related to software problem but also the problem that we are solving using the semantic model. If you are thinking is this project about technology, how can multilingual environment relate to software, then you are not wrong. All we discussed is about people, language and field and said this is what we are solving and nothing about technology. Let us help you here with how the analogy of this people, field and language is related to what we are doing. The first challenge that we specified above is relating to the semantics of communication between the employer and employee. The same problem exists in the software as well. If two or more software modules do not communicate in a same common language with common semantics, then they don't work well, they fail or system breaks. It follows that we need something standardised. The second challenges that we specified above is about co-operation between the employees from different background itself. In software, we refer to this kind of problem as interoperability problem. Again, to solve this, we need standardisation. Semantic models, especially in the form of ontology, such as what is developed as a part of smashHit project does exactly this task, which is (i) to provide the standardised semantics for all software components and (ii) to facilitate interoperability. If we have to say in technical terminology, the semantic models provide the formalisation of the real-world concepts in a machine-readable format, thereby enabling the interoperability. The concepts in the case of this project are about the GDPR and data processing as for example, consent and the different types of data processing activities. The smashHit semantic model provides a standardised vocabulary that can not only be used in this project but in any other related project with similar scope.

Following up on the challenges mentioned above, and coming back to fields again, when there's a difficulty in communication and co-operation: for any task you might have to refer back to employee and employer multiple times, i.e., there will be multiple to-and-fro communications. This in terms of software can be related to the IO (Input-Output) operations. If there are many IO operations then it creates a dependency problem and is difficult to scale. This is another challenge that semantic models help in solving, partly or completely.

2 Designing Semantic Models

Following the motivation for the use of ontologies in our project, when designing the smashHit core semantic model we were faced with several challenges, which can be summarized by the following:

- There are already many ontologies that exist regarding consent, contract, privacy, and legal rights but none that models all topics.
- To determine whether we create a new ontology from scratch or extend the previous ones.
- Defining the main classes and subclasses for consent, contract, tracing, and tracking in the process of data provision and data consumption/use.
- The overall design of the data workflow in application use cases (e.g. in the smart cities and insurance domains).
- To define the relationship between classes to provide identification of the leakage of data and audit-proof logging of transactions.

The rest of the details, such as the use cases where it is applied and the results achieved will be explained in the subsequent sections.

In order to overcome the challenges described above, we have followed best practices for the reuse of existing work in ontology engineering. We have investigated several existing ontologies and integrated published vocabularies, taxonomies and ontologies into the smashHit semantic model, from now on also mentioned as smashHit core ontology. For cases, where existing and integrated ontologies for modelling smashHit were incomplete, these ontologies were extended. The approach of the semantic model development followed was:

- specification of the main concepts to be used, coming from the smashHit components.
Note: the analysis of the already existing ontologies was made in parallel, overtaking said concepts from these ontologies.
- specification of the sub-classes for them (either reusing from the identified ontologies or from scratch)
- specification of the relationships (either reusing from the identified ontologies or from scratch)
- Verify that all requirements of the identified project use cases are covered within the smashHit Core ontology and make improvements as necessary
- iterate and add more details
- revisit and check the completeness of the ontology with the business cases' consent processes
- iterate and add more details

The mentioned external ontologies and vocabularies that were identified for reuse are: *GConsent*¹, *DPV*², *DCAT*³, *FIBO:Contract*⁴, *OntoSensor*⁵, *Prov-o*⁶, *CampaNeo*⁷ and *Languages, Countries, and Codes (LCC)*⁸.

The smashHit Core ontology contains the main smashHit concepts in the fields of:

- User identification such as *Agents* (including *Person* and *Organisation*) and respective identification categories (*PersonalDataCategory* which includes concepts such as *Name* or *Address*), see .
- Data identification such as *Metadata* or *Sector* which is a very important entity being the main identification and connection point to the data for which the data subjects will provide consent.
- Consent/Contract declarations such as *Consent*, *Contract*, *Status* and *Processing* (see Figure 2-2 and Figure 2-3)
- Description of the privacy context, expressed by such as *Purpose* and *Risk*.

Figure 2-1 shows the main entities overview of the smashHit Core ontology. In addition to these entities, relationships were also modelled in order to be able to relate entities, and later on, to model complex structures like the one we need to model a consent template and its use in the Automatic Contracting knowledge graph (see below in this paper). Also, the data identification entity *Metadata* connects to other entities such as *Resource* and *SensorDataCategory* for further categorization of the data in question.

¹ H. J. Pandit, C. Debruyne und P. McBennett, GConsent, A consent ontology based on the GDPR, ADAPT Centre, (Trinity College Dublin). DOI: 10.1007/978-3-030-21348-0_18

² H. J. Pandit, D. O'Sullivan und D. Lewis, „An Ontology Design Pattern for Describing Personal Data in Privacy Policies”, in: Proceedings of the 9th Workshop on Ontology Design and Patterns (WOP 2018), 2018.

³ <https://www.w3.org/community/dpvcg/wiki/>

⁴ <https://www.w3.org/TR/vocab-dcat-2/>

⁵ <https://edmcouncil.org/general/custom.asp?page=AboutFIBO4>

⁶ <https://mmisw.org/ont/univmemphis/sensor#>

⁷ <https://www.w3.org/TR/prov-o/>

⁸ https://www.digitale-technologien.de/DT/Redaktion/EN/Standardartikel/Smarte_Datenwirtschaft/Projekte/SDW_campaNeo_en.html

⁹ <https://www.omg.org/spec/LCC/1.2/About-LCC/>

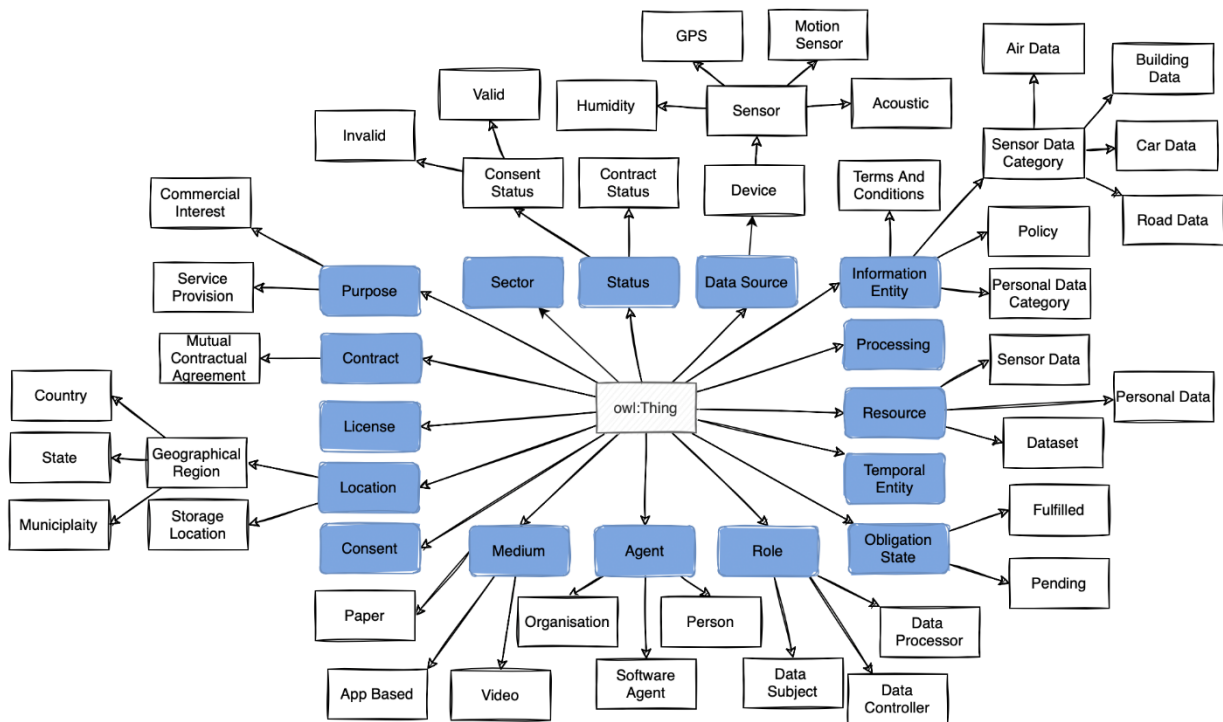


Figure 2-1: smashHit Core ontology main concepts overview

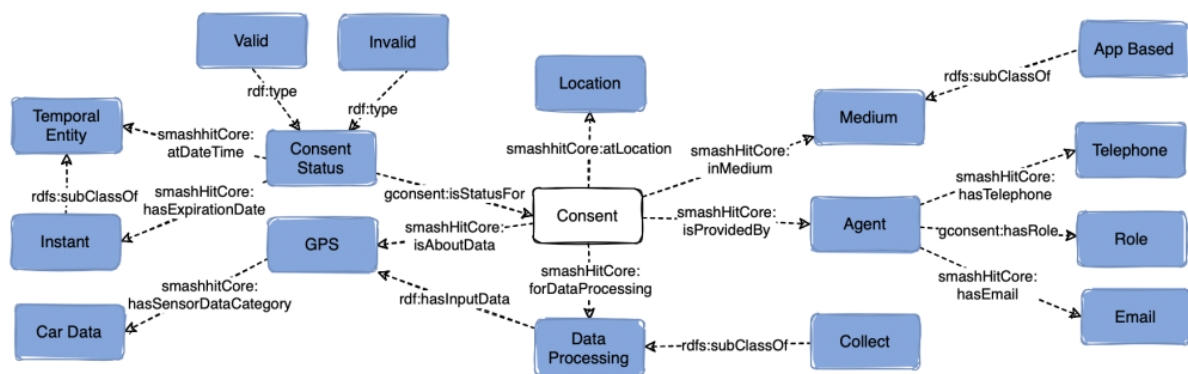


Figure 2-2: Overview of the class *Consent* and the classes related to it in smashHit Core ontology

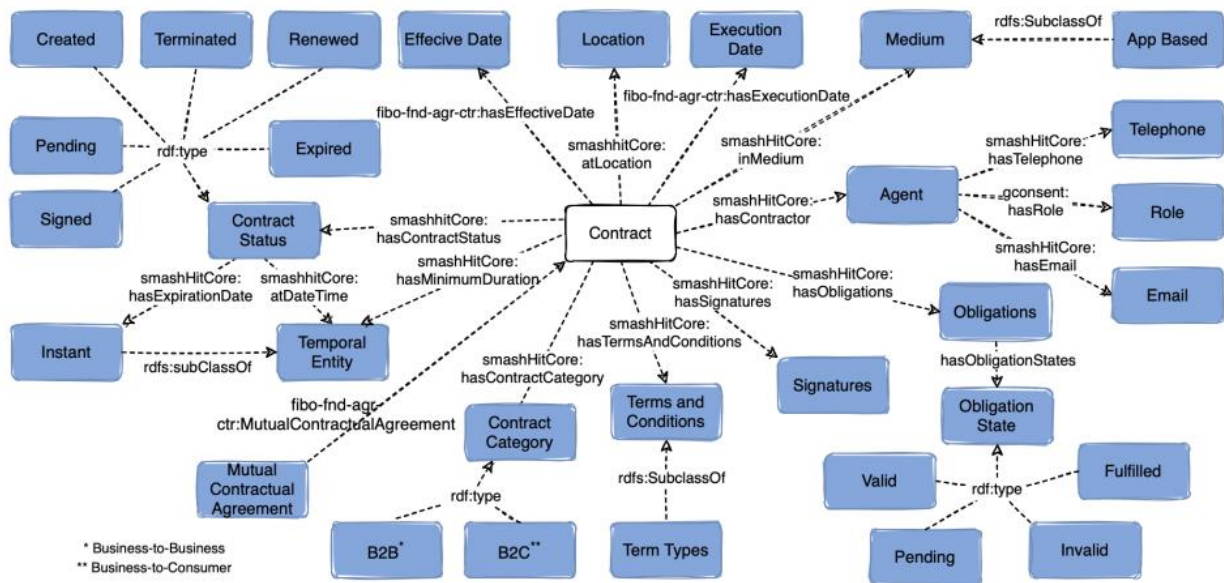


Figure 2-3: Overview of the class *Contract* and the classes related to it in smashHit Core ontology

One of the main issues that we faced while developing the smashHit semantic model was how to improve interoperability with other ontologies used in the same area. To do this, we studied the possibility of relating the ontology with an existing top-level ontology (TLO), such as *BFO* or *DOLCE*. The conclusion at the time of the project was that no TLO in particular would be used. The reasons for this decision were: on the one hand none of the reused ontologies is based on a TLO, therefore there is no guidance in selecting one in particular in the same smashHit core topics, and that would best fit the purpose of the smashHit Core semantic model and on the other hand, the short time in which the smashHit Core ontology was needed for use.

Therefore, the decision was not to select and use a TLO in particular. Instead, we decided to follow good ontology design practices that will lead to a simple and structured ontology, which can be aligned to a TLO for interoperability and standardization purposes in the future. These ontology design principles are derived from the *Basic Formal Ontology TLO*¹⁰ and are extended and adapted for the smashHit Core ontology:

- Use single nouns and avoid acronyms
- Ensure univocity (univocal = having one meaning only) of terms and relational expressions
- Distinguish the general from particular
- Provide all non-root terms with definitions (*dct:description*)
- Use essential features in defining terms and avoid circularity
- Start with the most general terms in the domain
- Use simpler terms than the term you are defining (to ensure intelligibility)
- Do not create terms for universals through logical combination
- Structure ontology around *is_a hierarchy* and ensure *is_a* completeness

¹⁰ R. Arp, B. Smith and A. D. Spear, Building Ontologies with Basic Formal Ontology, The MIT Press, 2015

Main assumption is that the smashHit Core ontology focuses on consent, contracts, data processing etc. All associated concepts needed to follow the consent life cycle from request up to revocation¹¹.

These design guidelines and assumption have allowed for improved overall consistency (hierarchically) by analysing and restructuring the entities, analysis and provision of sound and consistent definitions to all entities and data properties, and overall, errors identification and correction.

One of the best practices for ontology engineering that was followed while developing the smashHit Core ontology is the documentation of the entities imported or created. Documentation is made in two steps, the first is when entities are added to the ontology, they need to have a description for the term, and the second, is using a tool to create documentation that can be consulted at any time. The tool used was *WIDOCO*¹² which uses *LODE*¹³ (Life OWL Documentation Environment), with some further customisation, to create the smashHit Core documentation. The current version of the smashHit Core ontology documentation can be seen under:

<https://smashhiteu.github.io/smashHitCoreV1>

¹¹ A. Kurteva, T. R. Chhetri, H. J. Pandit and A. Fensel, “Consent through the Lens of Semantics: State of the Art Survey and Best Practices,” *Semantic Web*, p. 1 – 27, 1 January 2021

¹² <https://github.com/dgarijo/Widoco>

¹³ <https://essepuntato.it/lode/>

3 Application of Semantic Models

3.1 Standardised representation of consent and contract

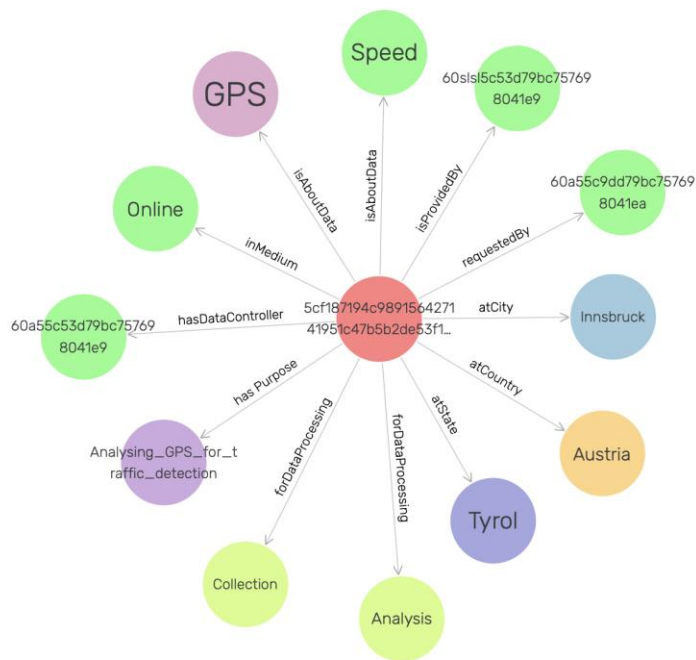
As we described in the introduction section, our solution is based on semantic technology, such as ontology. But what is a semantic model, and how is it beneficial compared with other data models like databases? A semantic model can be defined as the development of descriptions and representations of data in such a way that the latter's meaning is explicit, accurate, and commonly understood by both humans and computer systems. This definition encompasses a wide range of data artefacts, including metadata schemas, controlled vocabularies, taxonomies, ontologies, knowledge graphs, entity-relationship (E-R) models, property graphs, and other conceptual models for data representation. It represents the implicit meaning of the data by specifying the concepts and the relationships within the data. Information sources, such as relational databases contain a tremendous amount of structured data that can be leveraged to build and augment knowledge graphs. However, they rarely provide a semantic model to describe their contents.

For illustration, we take databases as an example of building data models. Later, we will compare it with semantic modelling. Creating a suitable data model in databases requires at least the following steps: (i) defining the purpose of the data. (ii) a proper normalisation of the data. (iii) reducing the redundancy. (iv) a proper naming convention. (v) to define constraint integrity properly. Due to this complexity, we can say that the data modelling process in databases is very complex. While creating semantic models on the other hand, is very simple. The ontology is used as a data model to create a semantic model by creating relationships between data when the data is organised. The data is organized into three essential parts—the first data element or subject, the relationship, and then the second data element or object. We can create classes, properties, and object properties with their relationships in ontology. Humans and machines commonly understood this semantic model. Further, the semantic model can aid the building of common solutions, foster interoperability, support knowledge discovery, and decision making. Following this, we created the semantic models for consent and contracts in the smashHit Core ontology.

The semantic model of consent is constructed as defined by GDPR's requirements (Article 7, Recital 32). In order to represent and store informed consent semantically within the knowledge graph (KG), the schema of the smashHit Core ontology has been followed. A KG instance of consent (graphical visualisation) is shown in Figure 3-1. The nodes of the graph represent the actual values, whereas the edges denote the relationship between nodes. The node with red colour is said to be a focus node and arrows from the focus node to other nodes describe the relationship between those nodes. For example, the arrow from the focus node to Tyrol denotes the relationship between consent and state — the location where the consent is granted. Similar, *hasDataController* represents a relationship between consent and the data controller.

The smashHit Core ontology goes beyond consent and addresses contract, which is another GDPR legal basis for data processing in smart cities and insurance domains. The main challenge is to bind GDPR rights with businesses, specifically in the form of digital semantic contracts. This creates an opportunity for small and medium enterprises (SMEs) to manage personal data in contracting services on scale. Our contracting solution is built in collaboration with both industry and legal experts. An instance of contract in the knowledge graph is presented in Figure 3-2.

To illustrate data processing between a data controller and a data processor in the smashHit, let us assume LexisNexis acts as a data controller and Infotripla Oy acts as a data processor (according to GDPR). In data processing where a contract is required, they must satisfy the requirements defined by GDPR (e.g., Art. 28, 32).



[5cf187194c989156427141951c47b5b2de53f1b3b8cea2f2e754edc01bf686ef](#) 

 5cf187194c989156427141951c47b5b2de53f1b3b8cea2f2e754edc01bf686ef

Types:

:ConsentID

RDF rank:

0

:GrantedAtTime

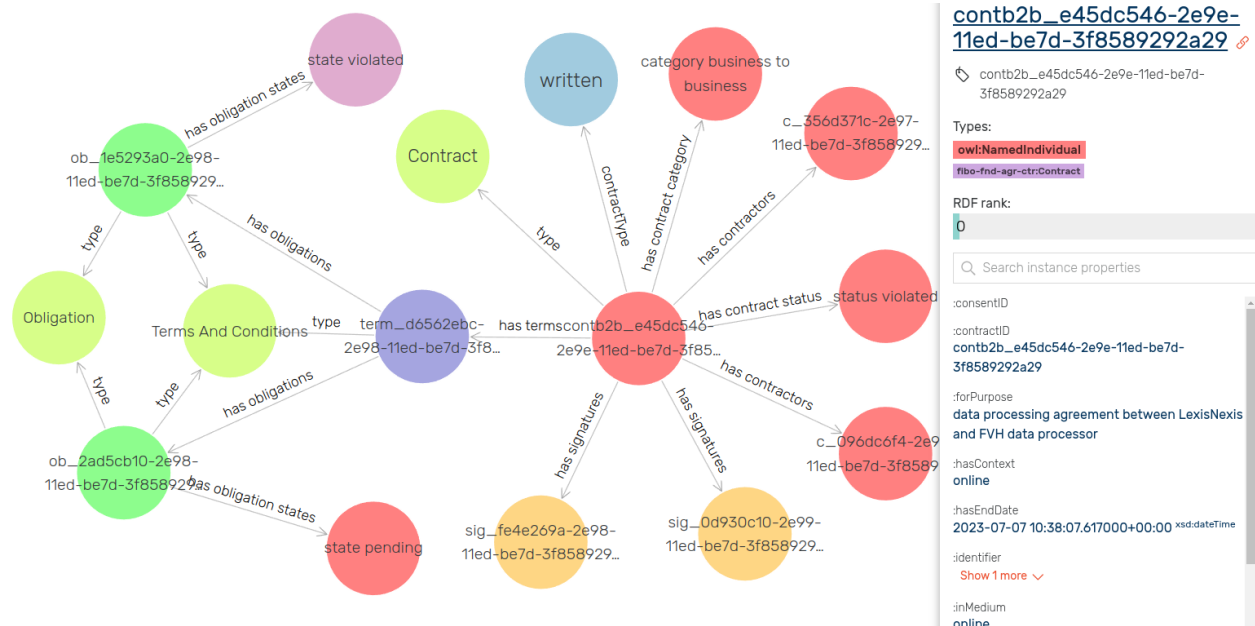
2021-05-21T09:03:31.859000+00:00

:hasExpiry


2021-05-21T09:03:31.859000+00:00

Figure 3-1: Knowledge graph instance of consent (graphical visualisation)

For instance, the data processor must notify the data controller if there is a breach of a contract. The insurers can view the information about the data storage and its usage.



[contb2b_e45dc546-2e9e-11ed-be7d-3f8589292a29](#) 

 contb2b_e45dc546-2e9e-11ed-be7d-3f8589292a29

Types:

owl:NamedIndividual

fiibo-fnd-agr-ctr:Contract

RDF rank:

0

Search instance properties

:consentID

:contractID
contb2b_e45dc546-2e9e-11ed-be7d-3f8589292a29

:forPurpose
data processing agreement between LexisNexis and FVH data processor

:hasContext
online

:hasEndDate

2023-07-07 10:38:07.617000+00:00 xsd:dateTime

:identifier

Show 1 more 

:inMedium
online

Figure 3-2: Knowledge graph instance of contract (graphical visualisation).

Figure 3-2 represents a B2B (business-to-business) contract between a data controller and a data processor. The central red node represents the focus node, while other red nodes denote the collection of values. For instance, the focus node has a relationship (has a contract category) with category business to business. It also shows the contract has a relationship with terms and conditions, which have a relationship with obligations.

3.2 Automatic Contracting Tool

Once the semantic models of consent and contracts are constructed, the next question could be how we can use these models in the real world to test their applicability. In smashHit, the Automatic Contracting Tool (ACT) supports the automatic generation of consent documents and execution of contracts based on specific terms and conditions in compliance with GDPR (as is modelled by the smashHit Semantic Model). It is a stand-alone module, which could be reused by any of the smashHit components or by external service providers via an API. The ACT provides the following functionalities:

- Semi-automatic consent/contract creation and annotation in the legal knowledge graph based on smart cities and insurance domains.
- Semi-automatic consent status update, namely, consent granting and revocation.
- Automatic GDPR compliant consent/contract document generation.
- Consent and contract modelling, specifically terms and conditions with knowledge graphs.
- Compliance verification in the case of a broken consent chain and contract breaches.
- Traceability of consent and contracts within the KG via relationships between concepts.
- Contracting User Interface (UI).
- B2C and B2B contracts modelling and implementations.

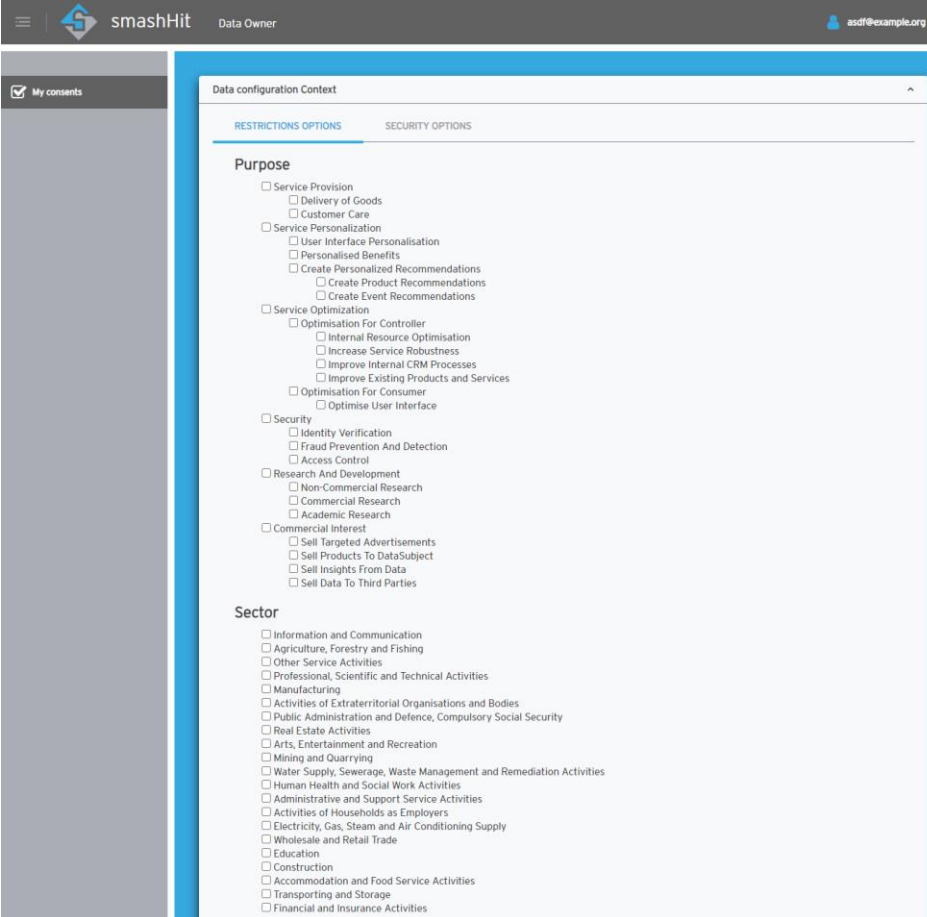
3.3 Data Use Traceability

Another application of the semantic model is the Data Use Traceability component. The Data Use Traceability component facilitates the tracing and tracking of data, in order to secure personal and industrial data sharing among companies and provide transparency to the data subject. Moreover, the Data Use Traceability component also tackles the risk of data leakage. For example, data can be fingerprinted or watermarked to enable re-identification of this data in case of a data leak. To provide the functionalities, we defined the classes and properties for Data Use traceability within smashHit Core ontology. We extended existing ontologies on data traceability reusing common properties and adding new ones, specific to our use case. We added new classes fingerprint, hash, and watermark specified in the semantic model as subclasses of processing class. We then added the properties *has description*, *has method*, and *has parameter* to the processing class. We also added properties *has fingerprint*, *has hash*, *has watermark* to the resource class. Thus, the semantic model can provide a common understanding of the concepts used in the Data Use Traceability component.

3.4 Context Sensitivity Solution

The Context Sensitivity Solution allows data subjects to configure context sensitive options associated with the active consents, which will be monitored continuously during consent's lifetime. Within the UI, the data subject will be offered with some options as seen in the following figure. Here, the user can select their desired context preferences associated with the consents, in which we are making use of the semantic model concepts *Purpose* and *Sector*.

After the new options are processed and stored by the CSS, any existing consents for the user will pass a compliance check to detect any conflicts with the new preferences. Any conflicts detected by the CSS will be reported via email to the data subject, so that they can manually review each of the offending consents and choose if they want to issue a revoke request to the consent.



The screenshot shows the 'Data configuration Context' interface in the smashHit application. The interface is divided into two main sections: 'RESTRICTIONS OPTIONS' and 'SECURITY OPTIONS'. The 'RESTRICTIONS OPTIONS' section is currently active and displays a list of checkboxes under two categories: 'Purpose' and 'Sector'.

Purpose

- ☐ Service Provision
 - ☐ Delivery of Goods
 - ☐ Customer Care
- ☐ Service Personalization
 - ☐ User Interface Personalisation
 - ☐ Personalised Benefits
 - ☐ Create Personalized Recommendations
 - ☐ Create Product Recommendations
 - ☐ Create Event Recommendations
- ☐ Service Optimization
 - ☐ Optimisation For Controller
 - ☐ Internal Resource Optimisation
 - ☐ Increase Service Robustness
 - ☐ Improve Internal CRM Processes
 - ☐ Improve Existing Products and Services
 - ☐ Optimisation For Consumer
 - ☐ Optimise User Interface
- ☐ Security
 - ☐ Identity Verification
 - ☐ Fraud Prevention And Detection
 - ☐ Access Control
- ☐ Research And Development
 - ☐ Non Commercial Research
 - ☐ Commercial Research
 - ☐ Academic Research
- ☐ Commercial Interest
 - ☐ Sell Targeted Advertisements
 - ☐ Sell Products To DataSubject
 - ☐ Sell Insights From Data
 - ☐ Sell Data To Third Parties


Sector

- ☐ Information and Communication
- ☐ Agriculture, Forestry and Fishing
- ☐ Other Service Activities
- ☐ Professional, Scientific and Technical Activities
- ☐ Manufacturing
- ☐ Activities of Extraterritorial Organisations and Bodies
- ☐ Public Administration and Defence, Compulsory Social Security
- ☐ Real Estate Activities
- ☐ Arts, Entertainment and Recreation
- ☐ Mining and Quarrying
- ☐ Water Supply, Sewerage, Waste Management and Remediation Activities
- ☐ Human Health and Social Work Activities
- ☐ Administrative and Support Service Activities
- ☐ Activities of Households as Employers
- ☐ Electricity, Gas, Steam and Air Conditioning Supply
- ☐ Wholesale and Retail Trade
- ☐ Education
- ☐ Construction
- ☐ Accommodation and Food Service Activities
- ☐ Transporting and Storage
- ☐ Financial and Insurance Activities

Figure 3-3: User preferences configuration options

3.5 Consent registration templates

Another application of the smashHit semantic model is in the communication between the external data subjects looking to collect consent for their applications and the smashHit platform. In order to be able to gather consent from data subjects, the data controllers and processors need to first make sure that the consent request and registration is “understood” by the smashHit platform and therefore this application registration follows a template that is compliant with the smashHit Core ontology, i.e. as shown before a consent request has mandatory fields (see Figure 2-2 and an example in Figure 3-1) and these are followed by the external data subjects when planning the communication with the smashHit platform (see Figure 3-4).


smashHit
System Administrator
admin@smashhit.aif-mobility.eu

- Data Owners
- Organizations
- Applications
- Consent templates

Go Back

Consent Template Details

ID	e2fb60afd201af000a48a693	
NAME	Email address for managing user account and consent information	
CREATED	August 16, 2022 at 11:17:35 AM GMT+2	
ORGANIZATION	[REDACTED]	
PURPOSE	Service Provision	
PURPOSE DESCRIPTION	Your email address is used to manage your user account and the consent information you provide. This information is stored in the smashHit system.	
PERSONAL DATA	CATEGORY	DATA
	feedback	feedback
	Contact	Email Address
SECTOR	Information and Communication	
DATA PROCESSING	Use, Store, Share	
AGENTS	APPLICATION	ROLE
	[REDACTED]	dataController

Edit Consent Template

Figure 3-4: Example of a data controller consent template for their App

4 Conclusion

From the preceding discussion, it is evident that communication is one of the most challenging aspects, especially for distributed systems. Furthermore, we have learned from the discussion that standardisation is the answer. As demonstrated by the applications of semantic models, semantic models can aid in standardisation and solve communication or interoperability issues between all software modules and stakeholders. However, semantic models present their own set of challenges, and if these challenges are not addressed, the models will not provide any benefits and will instead cause problems.

The following are the key takeaways from this technical essay:

- Semantic models can help with standardised representation of information, thereby helping in addressing the interoperability problem.
- Always follow the best practices while building the semantic model, i.e., ontology. For example, always look for existing information that can be reused and never try to reinvent the wheel.
- Involve all the stakeholders early in the design process and agree on common design principles and conventions. This is even more important when you are dealing with legal stuff, such as GDPR, where the early involvement of a legal team is absolutely necessary.

We were able to show the validity of the developed model, as the smashHit Core ontology was build and tested in the fields addressed by smashHit within two dimensions: i) internally by the smashHit software components using the semantic model as a formal guidance for their internal data model, ii) between the two business cases and the smashHit Framework, i.e. for the sake of modelling the consent templates for three different applications (i.e. in the vehicle insurance and smart city domains).

We are also quite convinced, that the developed smashHit Core Ontology models all relevant elements in the field of consent and contract. This being said, our model can and should be applied in other frameworks and solutions dealing with the same challenges.

5 Glossary

ACT: Automatic Contracting Tool

Agent: Entity that bears some form of responsibility in the context of a consent

B2C: Business-to-Consumer

B2B: Business-to-Business

Consent: As per Article 4(11) of the GDPR, 'consent' of the data subject means any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her

Consent template: A template defining the consent to be requested by a smashHit application, including: purpose, personal data, data processing, and agents

CSS: Context Sensitivity Solution

GDPR: Abbreviation for 'General Data Protection Regulation', a legal norm on EU level adopted in 2016, which is directly applicable within its scope and lays down rules for the processing of personal data so as to protect natural persons' fundamental rights and freedoms, in particular their right to the protection of personal data.

KG: Knowledge Graph

Personal data: Any information which are related to an identified or identifiable natural person (GDPR Art.4 (1))

Purpose: The purpose of Data Handling

SMEs: Small and Medium size Enterprises

TLO: Top-Level Ontology – ontology which consists of very general terms that are common across all domains

UI: (Contracting) User Interface



➤ **Our vision** - Solving Consumer Consent & Data Security for Connected Car and Smart City

➤ **Further information**

This document is part of the smashHit Methodology. The complete set of documents, including user/developer guides as well as a concept white paper created within this scope can be found on our website:

<https://smashhit.eu/publications>

➤ **Our consortium**



Funded by the Horizon 2020 Framework Programme of the European Union

Every effort has been made to ensure that all statements and information contained herein are accurate, however the smashHit Project Partners accept no liability for any error or omission in the same.

© 2022 Copyright in this document remains vested in the smashHit Project Partners.

