

Operationalising Foundational Ontologies for Enforceable Data-Use Policy: A Case of Privacy Policy

Alexandra Rowland^{1,2}

¹University of Twente, Drienerlolaan 5, Enschede, The Netherlands

²Kadaster, Hofstraat 110, Apeldoorn, The Netherlands

Abstract

Usage control seeks to govern the use of data after it has been shared, and in the semantic strand of that field a policy pairs a deontic component with a condition evaluated as a graph pattern over a knowledge graph. The mechanisms for expressing and evaluating such policies are well developed, and the deontic content of a policy has recently been given a foundational grounding. The vocabulary such a condition is written in must come from a model of the governed domain. Whether a model of that kind can be operationalised is the question posed in this paper. A single norm is traced from a fragment of a foundational privacy ontology, built in OntoUML based on the Unified Foundational Ontology (UFO), through a constraint stated over it, into a pattern evaluated against a graph. The passage is found to lose structure in four distinguishable ways, calling for four different responses. What the model says about how its facts must hold — that a role is borne only in virtue of an occasion, that a norm depends on the context it belongs to — is lost in a way that runs across them. The commitments the case exercises are among those the passage carries least well, and the paper closes by asking what the grounding secures if this is so, and what might compensate for what it does not.

Keywords

formal ontologies, usage control, privacy, OntoUML

1. Introduction

Data often has value beyond the purposes for which it was collected, and a party holding it may be willing to release this data provided its use remains well governed. The difficulty is that sharing and governance tend to come apart: once data has passed to another party, the terms under which it was released are no longer enforced by anything, and the holder is left to rely on contract and trust with third parties. Usage control as a field addresses this by attaching to the data a policy that continues to govern its use and by evaluating that policy at the point of use rather than at the point of access.

A policy of this kind is a set of rules, and a rule pairs a deontic component, which states that some action is permitted, forbidden or obligatory, with a condition determining when that statement applies. The mechanisms for expressing and evaluating such conditions are well developed. What is not generally supplied is the vocabulary a condition is written in: usage control assumes those terms are defined and represented elsewhere. Where nothing supplies them, a condition is written over whatever labels the data happens to carry, and such a condition may decide the cases before it correctly while being answerable to something other than the norm it is meant to enforce. Which of the two it is answerable to appears only when the case varies.

Supplying terms in which that separation can be drawn is the work of a domain ontology, and it is what the wider project undertakes: a foundational ontology of privacy, built in OntoUML on the Unified Foundational Ontology and resting on an ontology of secrecy that supplies the subject's interest [1]. That ontology is still under development, and whether it is right in every particular is a question the project continues to pursue. What we ask here, specifically, is whether a model of this kind can be carried into a form a system can evaluate. A model is not a condition: the ontology states what

ER2026: Companion Proceedings of the 45th International Conference on Conceptual Modeling: Doctoral Consortium, Tutorials, Posters and Demos, October 5-8, 2026, St. John's, NL, Canada

✉ a.c.s.j.rowland@utwente.nl (A. Rowland)

ORCID 0000-0002-2339-6357 (A. Rowland)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

a violation is, while a system needs a pattern it can match against a graph, and the transformation between the two does not preserve everything the model settles.

What of a foundational model's structure survives that transformation, and in how many different ways does the rest fail to survive it is the question posed in this paper. The work is situated in the semantic strand of usage control, where a policy is expressed in ODRL (the Open Digital Rights Language) grounded in OWL and its conditions are evaluated as graph patterns over a knowledge graph [2]. The remainder of the paper sets out a case that fixes what a condition must be able to test, states how a fragment of the ontology settles those terms, and traces a single norm through the passage from model to condition to see what arrives.

2. Problem Context

A condition determines when a policy statement applies, and it does so by testing what the system records. Where a permitted and an impermissible use differ in some recorded property (a different party, a different record, or a different category of data), the condition has something to test, and the mechanisms of usage control are sufficient for the task. The difficulty arises where two uses agree on every property a system records, and what distinguishes them is not carried by the record at all.

Such cases are not marginal to the governance of privacy, whose norms are integral to the setting from which information originated and the capacity in which a party holds it. One record applied to two different uses illustrates this. A clinician conveys a record to an insurer for the coordination of a patient's care; the insurer later applies it to the pricing of that patient's risk. The first use is permitted, and the second is not, yet the party is the same, the record is the same, and the data is identical in both. What differs is the capacity in which the insurer holds the record, which is no longer that of a party to the coordination of care, and the setting the information came from, whose norms no principle of the latter use satisfies.

What sort of thing a context is, what it means for information to belong to one and what distinguishes the insurer's being an underwriter from acting as a party to the coordination of care are questions not supplied by policy languages when conditions are defined. These concepts are two of the five parameters defined by Nissenbaum's contextual integrity, which usefully support the identification of the appropriateness of the flow of information between settings [3]. However, for this to be usefully operationalised in usage control contexts, these questions should be answered. This is the work of a domain ontology.

How these terms are supplied bears directly on whether a condition holds up, as two renderings of the same case show. Suppose the recipient is modelled as a class the party falls under. The graph then records that the insurer is a party to the coordination of care, and the record is true: it was true when the conveyance occurred. It is also permanent. Nothing marks the assertion as one that lapsed when the flow it belonged to ended, so a condition asking whether the later use is licensed finds it, matches, and licenses the use. The rendering decides the traced case correctly and the varied case wrongly, and what it has missed is anti-rigidity: the insurer bears that capacity in virtue of the flow in which it received the record, and not as a standing property of the company.

Suppose instead that the information is modelled as a category attached to a record, as the data privacy vocabularies model it. A condition written over that category tests the label and not the information. Rename the field, or export the content under a schema that classifies it otherwise, and the norm lapses while the content is unchanged. Modelling the information as a proposition that belongs to a context of origin, and the record as a carrier that bears it, holds the norm to the content across every carrier; a condition ranging over the information rather than the carrier needs no restatement for each copy, export or message.

Neither rendering fails on the traced case. Both decide it correctly, as a condition written over labels also does. What the ontology secures is that the condition still answers to the norm once the case varies, and it is that which must survive into whatever form a system evaluates. A condition must be evaluated against something, and the data here are held as a knowledge graph, a setting adopted because the

vocabulary a condition draws on and the data it is evaluated against then sit in a single representation, which is what allows a domain ontology to ground a policy directly. Graph patterns follow from that choice rather than being adopted alongside it, since matching against assertions is how a graph is interrogated, and GUCON formalises the arrangement as the condition of a usage-control rule [2].

The representation is chosen because it permits an ontology to ground a policy, but a graph holds assertions and a pattern asks after assertions, while a foundational model states more than assertions. That a party bears a role only in virtue of some occasion, or that a norm could not exist apart from the context it belongs to, are not facts about instances. They are claims about how the facts must hold, and a pattern has nowhere obvious to put them. Whether they survive the passage, and what else fails to, is what the remainder of this paper examines.

3. Related Work

The concerns of this paper fall within usage control, the tradition that extends access control by governing the use of data beyond the moment of access and by treating enforcement as continuous and re-evaluated as the circumstances of use change over time or space. Two of that tradition's distinctions bear directly on what follows. The first is the separation of what a norm requires from the mechanism that secures it; the second, owed to Hilty and others, is the distinction between the controllable and the observable, between a norm a system may enforce in advance and one whose violation can only be detected after the fact [4]. This paper draws on both, though the question posed here is prior to both: not how a norm is enforced, but what a norm must be expressed in terms of before enforcement can arise at all. The predominant usage-control frameworks resolve the terms of a policy against attributes supplied at the point of decision [5]; the concern here is with the semantic strand of the field, in which a policy is expressed in ODRL grounded in OWL and its conditions are evaluated as graph patterns over a knowledge graph, and in which a domain ontology may be made to ground the policy in such a graph.

Within this strand, the means of expressing and evaluating policy are well developed. GUCON provides a generic framework in which a usage-control rule pairs a deontic component with a condition expressed as a graph pattern, evaluated against a knowledge base, and instantiable in ODRL, OWL and SHACL [2]. It is the enforcement mechanism defined there that the present work presupposes. The work itself observes that its instantiations require an ontology to express the elements a policy refers to, without defining what such an ontology must contain. The mechanism of the condition is provided for while the vocabulary in which a condition is written is not.

The deontic content of policy has likewise been grounded. An ODRL rule has two parts: a deontic component, which states that some action is permitted, prohibited or obligatory, and a constraint, which states the condition under which the rule applies. The recent grounding of ODRL in the legal-relations ontology UFO-L addresses the first of these, mapping each rule to the normative positions it establishes when a policy is in force [6]. The constraint is set aside there because it creates no normative position of its own.

On the side of the domain, the most developed privacy vocabularies address the information surrounding a policy rather than the policy itself. GConsent offers a detailed ontology of consent, its states, its provenance, and the party who grants it, but states expressly that it models the information associated with compliance rather than determining compliance, and does not represent policies or the conditions on which they turn [7]. The Data Privacy Vocabulary, which is the most widely adopted of these and is expressly designed to be used alongside ODRL, supplies taxonomies of personal data categories, processing operations, purposes, and legal bases [8]. What it provides are terms with which a condition may be annotated; what determines whether a given use offends a norm is not among them. Purpose, in such vocabularies, appears as a category to which a use is assigned rather than as a relation between a use and the interest it bears upon, and it is this relation that a category assignment leaves unrepresented. That a record was disclosed for the coordination of care and later turned to the assessment of risk is, in these terms, the assignment of two purpose categories to one datum: a description under which nothing distinguishes the permitted use from the offending one. The broader privacy and data-protection

ontologies share this orientation: PrOnto and comparable regulatory models render the roles, data categories, and obligations of a legal regime with precision [9], yet they presuppose rather than model the intentional structure on which a privacy norm ultimately rests.

What remains open is therefore a question of level rather than of absence. That a condition requires a vocabulary, and that the vocabulary must be supplied from somewhere, is recognised, and it has been addressed directly. Supplying the inputs an evaluator requires is not, however, the same as establishing what a condition is answerable to, and neither settles what becomes of a domain model's structure when that model is carried into a pattern a system evaluates. It is that passage, and what it costs, with which this paper is concerned.

4. Method

The work reported here is design science, and its artefact is the passage from a foundational model to a condition a system can evaluate. What is built is the fragment of Section 5, the constraint stated over it, and the pattern derived from that constraint; what is examined is what the second of these carries into the third.

The passage runs in two steps, each of which is a place where structure may be lost. The model becomes a vocabulary formalised in OWL, for which established practice exists in gUFO [10], the OWL rendering of the foundational ontology, though a decidable fragment of description logic cannot carry everything a foundational model expresses. The constraint then becomes a graph pattern, whose binding semantics answers to the activation of a rule and which is evaluated against a knowledge base a system maintains. The diagnostic follows a single norm through both steps and asks of each clause of the constraint three questions: whether it can be written as a graph pattern at all; whether what it tests is recoverable from the data, or inherited from a judgement made upstream in the modelling; and whether it resolves within the vocabulary the ontology supplies, or reaches into a layer the model does not define. The kinds of difficulty encountered are given by which of these a clause fails, and not by observation of the trace.

The fragment traced is smaller than the ontology it belongs to, and for a reason bearing on the method. The ontology is developed by ontological unpacking [11]: a candidate concept is proposed, given a provisional stereotype, and tested against concrete scenarios chosen to pull the alternatives apart, the stereotype being accepted only where the scenarios discriminate in its favour. Only those elements the present scenario exercises have been so tested, and where two readings survive the available cases the decision is recorded as open, as several are above.

Tracing a single norm cannot establish that the difficulties encountered are general, and no claim of that sort is made here. What the trace can establish is that difficulties of different kinds arise, which is enough for the argument the paper makes: were the loss of one kind, a single remedy might be sought, and the finding is that it is not. The claim is one of necessity rather than of exhaustiveness.

5. The Contextual Layer of the Privacy Model

The problem context sketched in Section 2 identified two main concepts that required definition: the capacity in which a party holds information, and the setting the information originated from. What follows states how they are preliminarily settled in a fragment of a larger ontology of privacy, developed in OntoUML based on the Unified Foundational Ontology (UFO) and built using a method of ontological unpacking [11]. Only those elements the case set out in the problem context exercises are laid out here. Beneath them lies a further layer, in which the subject holds a standing over information about them that does not depend on the setting they are in. The fragment presupposes that layer without representing it. It is there, too, that the subject's interest is given an ontological account, by way of the SECRET relator and the beliefs it bears [1], and it is that interest which makes the norms of a setting privacy norms rather than rules that happen to hold there.

Capacity

The parties to a privacy relation are represented as bearers of roles rather than as instances of classes, and the roles are borne in virtue of participation in a particular flow. The insurer is a recipient, and a recipient of a particular kind, because it took part in the conveyance of that record on that occasion. The roles are anti-rigid: a party that is an insurer is an insurer necessarily, in the sense that it could not cease to be one and remain the entity it is. A party to the coordination of care, on the other hand, is one contingently and for as long as its participation lasts.¹

The data subject is a role of a further kind because the subject need not participate in the flow at all. In the case at hand the patient is neither sender nor recipient, and the model accordingly connects the subject to the flow through the information the flow carries rather than by participation.²

Origin

Within a flow, the information is distinguished from the carrier that bears it. The information is an abstract proposition, inherited from the secrecy layer's *SECRETUM*, and it belongs to exactly one context of origin. The carrier is any concrete thing (a record, a message, an utterance) that bears the information and may therefore participate in a flow.

Belonging to a context is, on this treatment, a relation the information stands in and not a category assigned to a record. The norm accordingly ranges over the information, which persists across carriers and forms, while a flow moves a particular carrier.

Norms, Flows and Violation

The norm is modelled as a social object that externally depends on a single context: the setting whose norms it expresses and without which it does not exist. The flow is modelled as an event in which sender, recipient and carrier participate, and it is a flow of the information belonging to its context that a norm may proscribe.³ A privacy violation is not modelled as a further event but as a relation between a flow and the norm it breaches. It holds when the information a flow carries is about a data subject and belongs to the context on which the norm depends, and when the flow occurs under no transmission principle. That the information is about a subject is not incidental: it is what distinguishes a privacy violation from the unauthorised crossing of a boundary as such, because the wrong is done in respect of the party the information concerns. The violation thus falls out of the structure rather than being separately asserted.⁴

The Boundary of the Model and the Deontic Layer

That a holder may not reuse a record for the purpose of underwriting is not simply a prohibition but the correlative of a claim the subject holds, and it is this relation, rather than a permission flag, that the norm asserts. Here the model meets the deontic grounding of policy provided by Mustafa et al. [6]. That work makes precise the normative positions a rule creates when a policy is in force; the model presented here makes precise, in parallel, what the condition of such a rule must be about.⁵ The ground

¹What confers a capacity, whether the principle under which information was conveyed, or a party's position or institutional membership independently of any conveyance, is not settled here. In the case traced below the two coincide, since the insurer holds the record as a party to the coordination of care in virtue of the arrangement under which it received it.

²Whether the data subject specialises *SECRET OWNER* or is a role in its own right is among the modelling decisions still to be formalised; this paper does not offer a resolution.

³Whether the application of information to an end proper to another setting is itself a flow, or an event of a further kind the model must accommodate, is a decision the present work leaves open; the analysis below does not depend on it.

⁴The precise typing of the violation, and its relation to the secrecy layer's treatment of *REVELATION* as an event, remain under development; the relational reading is adopted because the flow already plays, in the privacy layer, the role that *REVELATION* plays below.

⁵Whether the claim borne by the subject is identical to the assigner's right that the deontic grounding already pairs with each duty, or a distinct position grounded in the subject's interest, is a further modelling decision left open; the parallel drawn here does not depend on how it is settled.

on which a flow may nonetheless be licensed (i.e. the transmission principle under which it occurs) is itself a deontic position and is not modelled here. It is at this point that the domain model reaches the boundary of what it defines and hands off to the adopted deontic layer.

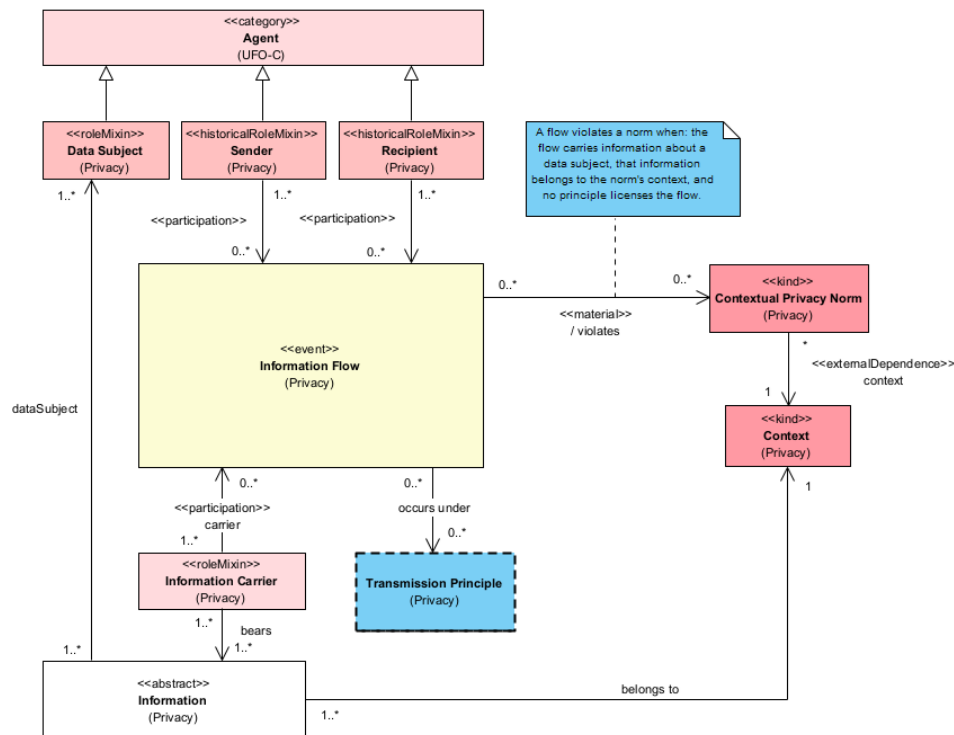


Figure 1: OntoUML fragment showing the contextual privacy model.

Figure 1 presents these elements as a single OntoUML fragment. Three conditions are stated of a violation: that the information a flow carries is about a data subject, that it belongs to the context on which the norm depends, and that no transmission principle licenses the flow. Each rests on a commitment taken above: that aboutness is a relation the information stands in, that belonging to a context is likewise a relation and not a category assigned to a record, and that whether a flow is licensed is a question the domain model does not itself answer. The commitment on which the case of the problem context turned, that a capacity is borne in virtue of a flow rather than held as a standing property, appears among none of them. It is these three conditions, and that absence, which the following section attempts to carry into a form a system can evaluate.

6. From Model to (Enforceable) Condition

The model states what a violation is. A system, however, cannot evaluate a statement of that kind; it requires a pattern to match against the data it holds and two transformation steps separate the model's representation and an enforceable condition. The model first becomes a vocabulary formalised in OWL, so that contexts, flows and subjects are terms a graph can be interrogated, and the constraint stated over the model becomes a graph pattern whose bindings answer to the activation of a rule. Using the methodology defined in a previous section. this section follows one norm through both steps, clause by clause, and asks of each clause whether it can be written as a pattern, whether what it tests is recoverable from the data or inherited from a decision made in the modelling, and whether it resolves within the vocabulary the ontology supplies. The content of the norm may be given in OCL over the fragment of Figure 1:

```
context InformationFlow
```

```

def: violatesNorm(n : ContextualPrivacyNorm) : Boolean =
  self.carrier.information->exists(i |
    i.dataSubject->notEmpty()
    and i.context = n.context)
  and self.occursUnder->isEmpty()

```

This, however, is a specification and not an implementation. It states what the model holds a violation to be, in the terms the model provides. What a system evaluates is a pattern, and the scenario of Section 2 supplies a graph for it to be matched against:

```

:flow1 a :InformationFlow ; :sender :clinician ; :recipient :insurer ;
      :carrier :record4471 .
:record4471 :bears :prop88 .
:prop88 :belongsTo :ctxCare ; :dataSubject :patient12 .
:norm3 a :ContextualPrivacyNorm ; :context :ctxCare .

```

The constraint becomes a pattern over these terms. The navigation of the OCL becomes a chain of triple patterns, its existential quantification becomes the binding of a variable, and its requirement that no transmission principle apply becomes a negation:

```

?flow :carrier ?c . ?c :bears ?i .
?i :dataSubject ?s ; :belongsTo ?ctx .
?norm :context ?ctx .
FILTER NOT EXISTS { ?flow :occursUnder ?tp }

```

The pattern then fills the constraint defined by the rule: a prohibition on the use of the record by the insurer, active for those bindings under which the pattern matches, which an engine of the kind described in Section 3 evaluates against the graph. The passage is complete, and what it has carried can now be examined.

The flow and its participants pass into the pattern as they stand. That a party stands as recipient of a flow, and that a carrier participates in it, are assertions a graph may hold and a pattern may match; the participants are recoverable as the participations of the event, and nothing in the clause reaches outside the domain vocabulary. Where structure crosses in this way, nothing further is required of the condition. The distinction between information and carrier survives in the same manner. Because the pattern ranges over the information rather than the record that bears it, it is indifferent to the form a flow takes and to the number of carriers bearing the same content, and a condition written over carriers would require restatement for each.

The clause requiring the information to belong to the norm's context crosses on a condition. It can be written as a pattern and amounts to an identity between two nodes of the graph. What a pattern cannot establish is which context a given item of information belongs to. That assignment is a judgement made based on the ontology rather than a fact recoverable from the data, and the pattern inherits it without being in a position to make this decision or to record that it was made. The clause is therefore answerable to a decision nowhere visible in what the system holds, and the condition depends on it silently either way. What is required is not a refinement of the pattern but discipline in modelling: the judgement must be made deliberately and recorded where it can be found.

The requirement that the information be about a data subject is representable as a triple asserting that some information concerns some party, but representability is not the same as availability. Systems record what data they hold and, on occasion, to whose account a record belongs. They do not generally record what a proposition is about. The clause may be written, and the pattern will match wherever the relation is present, yet its evaluation presupposes a graph maintaining a relation few systems presently maintain. The difficulty here is not ontological, and no refinement of the model will close it.

Interestingly, last clause fails differently because whether a flow occurs under a transmission principle is not a fact about the domain but a deontic question, which the model defers to the adopted deontic

layer. The clause can be written but it does not resolve within the vocabulary the ontology supplies. Rather, it reaches across to a layer the ontology does not define, and what it requires is a defined seam rather than further domain modelling.

The four are distinguished by where a clause fails rather than by observation. Structure may cross into a pattern as an assertion; it may cross while carrying a judgement the graph cannot make; it may be expressible and yet not maintained; or it may fail to resolve in the domain vocabulary at all. A method for carrying foundational models into enforceable conditions would otherwise treat these as one problem and seek a single remedy. They are four, and one of them is not a modelling problem.

A loss of a different kind runs alongside them. A pattern matches assertions, and the model states more than assertions. That a violation is derived rather than asserted is carried by the pattern itself, since what the model holds to follow from other facts is what the query computes from them; the derivation survives in the engine and not in the graph and is unavailable to anything reading the graph without running the rule. That the norm depends on exactly one context survives only in its consequence, as a cardinality a validation language can enforce, which preserves the well-formedness the dependence requires without preserving the dependence. And that recipient is a role a party bears in virtue of a flow, rather than a class the party falls under, does not survive at all: it is recorded in the OWL rendering as a metatype, since gUFO provides for the distinction [10], but the metatype is inert in evaluation. A triple typing a party asserts the type as it would assert any other, and a pattern matching it cannot tell that the classification was never essential to the party.

Roles, contexts, flows and the information they concern therefore pass into a pattern as assertions. The aboutness a privacy norm turns on can be expressed but is seldom recorded. The licensing of a flow can be expressed only by pointing to a layer beyond the model. And the capacity in which a party holds information, on which the case of Section 2 turned, appears in no clause.

7. Discussion and Open Questions

Two commitments were identified in Section 2 as those on which a condition's holding up depended: that a capacity is borne in virtue of participation in a flow rather than held as a standing property, and that information belongs to a context of origin rather than carrying a category assigned to a record. The first appears in no clause of the constraint at all, and is recorded in the OWL rendering only as a metatype that evaluation ignores. The second survives the transformation, but carries with it a judgement the graph cannot make and does not record. What makes the grounding most valuable is, unfortunately, what the transformation carries least well.

It may be asked, then, what the grounding is worth. A condition written over labels would have failed in the same places: it has no account of the capacity in which a party holds information, and none of the setting the information came from. The difference is that it records nothing of its failure. Such a condition looks complete. Its assumptions are invisible, and it decides the varied case wrongly with no indication that anything is missing. A condition derived from a model fails knowably, and each of its gaps can be named (the judgement inherited, the relation unrecorded, the clause reaching beyond the vocabulary) because something upstream committed to what ought to have been carried. Legibility is what makes compensation possible, since one cannot build machinery against a gap one cannot see. The added value in the grounding is not that everything survives but that what does not survive is known.

Several compensations suggest themselves, and none is void of difficulty. The first is to arrange the graph so that the pattern's matching coincides with what the model requires. Capacity may be attached to the flow rather than to the party, so that a condition asking what capacity the recipient of a given flow bears finds nothing where no capacity was conferred. The pattern acquires no new power by this; the facts are simply arranged so that matching them yields the answer the model would give. But the arrangement is invisible. Nothing in the graph records that it is load-bearing, and a later decision to flatten the capacity onto the party, for the convenience of querying, would break the condition without any indication that it had. That a compensation should consist in circumventing the mechanism rather

than in extending it also raises the question of whether the mechanism is the right one, which the present work is not in a position to settle.

The second is to record the inherited judgement where it is made. The assignment of information to a context of origin is a decision, and a decision may be documented. Where the graph is virtual, and its contents are produced from relational sources by mappings rather than asserted directly, the mapping is the natural place for such a decision to live: it is there that source data acquires its ontological interpretation, and there that the interpretation may be inspected. This moves the judgement rather than removing it, and nothing obliges a mapping to justify what it asserts, but a judgement in a known place is of a different order from one distributed across a graph.

The third is to enlist validation alongside the pattern. A validation language can enforce the cardinality that the norm's dependence on a single context requires and so preserve the well-formedness the dependence demands. What it preserves is a consequence and not the dependence itself, and it offers nothing at all for anti-rigidity, which is the loss that matters most.

The fourth is to define the seam at which the condition hands off to the deontic layer, rather than allowing the clause to trail into it. Whether this is the right response depends on a question the present work has not settled: whether the boundary is a division of labour, with the licensing of flows properly belonging to a grounding of the deontic content of policy, or a defect in an ontology of privacy that declines to model the grounds on which a flow may be licensed.

The remaining difficulty has perhaps no compensation of this kind. That systems do not record what a proposition is about is not a modelling issue and will not be repaired by a better model. Whether it is a gap that better practice would close or a sign that aboutness is not the sort of thing a system readily records, is not clear, and the answer bears directly on whether a condition of the kind traced here could be evaluated in practice at all.

8. Research Context and Plan

The work reported here is part of a doctoral project on policy-aware information systems. The project pursues this in knowledge graph systems for three reasons. Ontological grounding is best supported there, since the vocabulary a policy condition draws on and the data it is evaluated against sit in the same representation. Federation comes naturally, so policies can govern data spread across parties without first consolidating it. And because data and metadata are held together, a policy can travel with what it governs rather than being maintained separately. Privacy and security are the domains in which the approach is developed and tested.

A foundational layer is in place. An ontology of secrecy, which supplies the intentional structure the privacy model builds on, has been accepted at ER 2026 [1]. The privacy layer, of which this paper presents the contextual fragment, is still being built. What the fragment sets aside, the claim a subject has over information about them independently of any setting, is the core focus at the time of writing.

The next step is the one this paper is focused on. The model and its constraints are specifications: the OntoUML fragment says what the domain contains, and the OCL says what, in those terms, a violation is. Neither is something a system evaluates. For that, each must be given an implementable counterpart. The model becomes a vocabulary formalised in OWL so that contexts, flows, and subjects are terms a graph can be interrogated about. The constraint becomes a graph pattern, expressed in SPARQL, whose binding semantics answer to the activation of a rule, while SHACL serves to validate the well-formedness the model requires. That pattern then fills the condition slot of a policy rule in ODRL, a prohibition on some action by some party *when* the pattern matches, which an engine evaluates against the graph. Carrying both transformations out for the ontology as a whole, rather than for the single norm traced above, is the immediate task.

Two further questions have been deliberately set aside for the moment. The first is enforcement: whether available mechanisms can act on conditions of this kind and where, following Hilty et al.'s distinction, what can be controlled gives way to what can only be observed [4]. The second is transfer: whether this kind of grounding serves domains beyond privacy, which presupposes the privacy work

now in progress. Both belong to later stages of the project, and neither can be usefully taken up until the move from model to condition is better understood.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude Sonnet 4.5 for sentence polishing, paraphrasing, and peer review simulation where required. After using this tool, the author reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] A. Rowland, G. Moura Amaral, T. Prince Sales, G. Guizzardi, R. Lemmens, What secrets are: Towards a formal ontology of secrecy, in: Forthcoming, 2026. Accepted, ER 2026.
- [2] I. Akaichi, G. Flouris, I. Fundulaki, S. Kirrane, GUCON: A Generic Graph Pattern Based Policy Framework for Usage Control Enforcement, in: A. Fensel, A. Ozaki, D. Roman, A. Soylu (Eds.), Rules and Reasoning, volume 14244, Springer Nature Switzerland, Cham, 2023, pp. 34–53. URL: https://link.springer.com/10.1007/978-3-031-45072-3_3. doi:10.1007/978-3-031-45072-3_3.
- [3] H. Nissenbaum, Privacy in Context: Technology, Policy, and the Integrity of Social Life, Stanford University Press, Stanford, California, 2010.
- [4] M. Hilty, D. Basin, A. Pretschner, On Obligations, in: D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, S. D. C. Di Vimercati, P. Syverson, D. Gollmann (Eds.), Computer Security – ESORICS 2005, volume 3679, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 98–117. URL: http://link.springer.com/10.1007/11555827_7. doi:10.1007/11555827_7.
- [5] I. Akaichi, S. Kirrane, A comprehensive review of usage control frameworks, Computer Science Review 56 (2025) 100698. URL: <https://linkinghub.elsevier.com/retrieve/pii/S1574013724000819>. doi:10.1016/j.cosrev.2024.100698.
- [6] D. M. Mustafa, C. Lange, G. Guizzardi, D. Collarana, C. Quix, S. Decker, What Does ODRL Mean? A Cross-Level Ontological Grounding of Permissions, Prohibitions, and Duties in UFO-L, 2026. URL: <http://arxiv.org/abs/2606.24344>. doi:10.48550/arXiv.2606.24344, arXiv:2606.24344.
- [7] H. J. Pandit, C. Debruyne, D. O’Sullivan, D. Lewis, GConsent - A Consent Ontology Based on the GDPR, in: P. Hitzler, M. Fernández, K. Janowicz, A. Zaveri, A. J. Gray, V. Lopez, A. Haller, K. Hammar (Eds.), The Semantic Web, volume 11503, Springer International Publishing, Cham, 2019, pp. 270–282. URL: https://link.springer.com/10.1007/978-3-030-21348-0_18. doi:10.1007/978-3-030-21348-0_18.
- [8] H. J. Pandit, B. Esteves, G. P. Krog, P. Ryan, D. Golpayegani, J. Flake, Data privacy vocabulary (DPV) – version 2, in: G. Demartini, et al. (Eds.), The Semantic Web – ISWC 2024, volume 15233 of *Lecture Notes in Computer Science*, Springer, Cham, 2025, pp. 171–193. doi:10.1007/978-3-031-77847-6_10.
- [9] M. Palmirani, M. Martoni, A. Rossi, C. Bartolini, L. Robaldo, PrOnto: Privacy Ontology for Legal Reasoning, in: A. Kó, E. Francesconi (Eds.), Electronic Government and the Information Systems Perspective, volume 11032, Springer International Publishing, Cham, 2018, pp. 139–152. URL: https://link.springer.com/10.1007/978-3-319-98349-3_11. doi:10.1007/978-3-319-98349-3_11.
- [10] J. P. A. Almeida, G. Guizzardi, T. P. Sales, R. A. Falbo, gUFO: A lightweight implementation of the unified foundational ontology (UFO), <http://purl.org/nemo/doc/gufo>, 2019. Accessed: 2026-07-24.
- [11] G. Guizzardi, N. Guarino, Explanation, semantics, and ontology, Data & Knowledge Engineering 153 (2024) 102325.