Modelleringsrapport

maandag 19 september 2022 16:25

- Vertekpunt is DPV versie 0.8.1.
- We namen de OWL-versie (10/9/2022) van DPV, ook DPVO genoemd. De standaard versie van DPV is de Concept versie maar OSLO werkt met klassen.
- Algemeen: van elementen die mappen namen we de naamgeving uit DPV over, klassen werden daarbij verhuisd naar een package DPV.
- OPMERKING: DPV maakt op veel plaatsen gebruik van subklassen waar we eerder een classificatie van Concepts verwachten. In die gevallen voorzien we een enumeratie in de plaats, bvb DataSubject.type vh type DataSubjectType.
- OPMERKING: OSLO-Consent is grotendeels gebaseerd op Gconsent, hoewel ook termen uit DPV zijn gebruikt.
- Hoewel de klasse PersonalDataHandling in DPV centraal staat (zie § 4.3 in de Primer) zijn alternatieve modellen mogelijk hoewel niet aanbevolen (zie §AlternateModelsToPersonalDataHandling in de Primer). Met dat doel hebben attributen doorgaans Thing als superklasse en niet PersonalDataHandling.
- Tegelijk echter wordt in §5.5. vd Primer geïllustreerd hoe een attribuut als hasDataSubject gebruikt kan worden als deel van Consent, reden is dat vooral bij een LegalBasis als Consent het om een instantie van Consent voor een specifiek datasubject gaat (bvb ConsentUser123), maw Consent wordt in dat geval niet louter gebruikt om het soort LegalBasis (in dit geval Consent, maar er zijn andere types bvb Contract, LegalObligation etc) aan te duiden.
- Consent.hasDataSubject vervangt nu isConsentForDataSubject en GConsent:Consent.isProvidedByDataSubject. DPV gaat er terecht van uit dat als het DataSubject niet diegene is die toestemming gaf, diegene die de toestemming gaf zelf geen DataSubject kan zijn (tenzij voor een andere Consent). Zie verder over Delegation.
- In dergelijk geval voorziet DPV het attribuut isIndicatedBy met als datatype Entiteit eerder dan DataSubject.
- ConsentStatus is in DPV een klasse met 2 subklassen ConsentStatusValidForProcessing en ConsentStatusInvalidForProcessing en een hoop instanties, bvb ConsentRequired, ConsentGiven etc. Net als in GConsent gaan we hier voor een enumeratie.
- Merk op dat PersonalDataHandling gemodelleerd kan worden voor meerdere instanties van Consent. Met PersonalDataHandling.hasDataSubject wordt dan een categorie van DataSubjects (bvb ServiceConsumers) eerder dan individuele DataSubjecten (bvb User123) aangeduid. Dit wordt geïllustreerd in §5.5 en 5.5.1 vd Primer.
- Omwille daarvan voegden we een attribuut type toe aan DataSubject zodat ook een categorie van DataSubjects kan worden aangeduid.
- Het vroegere GConsent.Consent.atTime is nu gemapped op DPV:Consent.indicatedAtTime.
- Problematisch in GConsent is dat de dataverwerking op zich niet wordt beschreven, de verschillende aspecten ervan (DataController, PersonalData, Processing, Purpose etc) worden met de Consent geassocieerd.
- Daardoor is alvast uitgesloten om met 1 Consent toestemming te geven voor meerdere dataverwerkingen, bvb in het geval dat de verwerking op basis van PersonalData van het type email is en deze gebruikt wordt om 1) Newsletters te sturen (Processing is storing) en 2) deze te delen met partners (Processing is sharing).
- Ook is het niet mogelijk om een dataverwerking te beschrijven alvorens er een instantie van Consent bestaat. Ipv dit af te dwingen dmv een kardinaliteit 1 zoals nu het geval is het beter om dit op te nemen in een beschrijving van de dataverwerking.
- En verder is het semantisch gezien vreemd om te zeggen dat een Consent instantie direct met PersonalData of een DataController geassocieerd kan worden zonder dat een dataverwerking is aangemaakt. Ook hier geldt dat dit beter niet wordt afgedwongen dmv een kardinaliteit 1 zoals nu het geval is.
- In DPV worden al deze aspecten wel samengebracht, mn in de klasse PersonalDataHandling. Deze klasse was aanwezig in OSLO-Consent, Processing en Purpose waren ermee verbonden maar PersonalData en DataController niet. We voegden deze associaties daarom toe (hasDataController en hasPersonalData). We schrapten meteen ook de

GConsent:Consent.isProvidedToController en GConsent:Consent.forPersonalData associaties.

- Voor OSLO-Consent:Consent.inMedium vonden we niet direct een equivalent, behouden.
- De self-associaties uit Consent hebben we geschrapt (OSLO-Consent:Consent.isInvalidatedBy, .isPreviousConsentFor, isUpdatedConsentFor etc). Deze komen uit GConsent maar hebben betrekking op versies van dezelfde Consent. Dit gaat dus over historiek en is maw een implementatieding.
- Sowieso biedt Consent.indicatedAtTime en Consent.consentStatus voldoende mogelijkheid om versies van eenzelfde Consent (met verschillende statussen) of opeenvolgende Consents van bvb hetzelfde DataSubject te beschrijven.
- Het is zeker niet nodig om voor elke wijziging vd status van een Consent een nieuwe instantie van Consent te creëren, in praktijk zal een nieuwe versie van dezelfde instantie worden gemaakt.
- OSLO-Consent:Consent.isGivenFor en OSLO:Consent:LegalBasis.isLegalBasisFor kunnen beide vervangen worden door LegalBasis.hasPersonalDataHandling. Zie §5.5.1.1 in de Primer voor gebruiksvoorbeeld.
- OSLO-Consent:Consent.notice hebben we gemapped op DPV:.hasNotice, ondanks de definitie bij OSLO die volgens ons verkeerd is (een notice is niet gewoon een string met bjikomende info, het is hetgeen het DataSubject te zien krijgt qua tekst, checkboxes etc bij het verlenen vd Consent.
- Ipv op Langstring hebben we het bovendien gemapped op DPV:ConsentNotice. Merk op dat een Notice 1 vd TechnicalAndOrganizationalMeasures is vermeld in de PersonalDataHandling (het is meer in het bijzonder een OrganizationalMeasure vh type Notice vh type PrivacyNotice).
- Net als in §5.5.1.1 vd Primer gesuggereerd wordt kan de Notice meerdere vormen aannemen (document, url, string), te vermelden in de usageNote.
- Om die reden verwijderden we ook OSLO-Consent:Consent.consentDocument.
- De subklasse MinorDataSubject hebben we verwijderd. Idem met Delegation. De aanpak in DPV is nl als volgt: via de associatie Consent.isIndicatedBy kan naar een Entiteit (in praktijk een Persoon, Organisatie) worden verwezen die in plaats van het DataSubject de Consent gaf.
- OPMERKING: indicatedBy vervangt eerdere attributen, zie https://github.com/w3c/dpv/issues/21 voor de mapping. OPGELET: In DPVO is niet te zien dat deze oude attributen deprecated zijn, in DPV wel.
- Verder zijn subklassen als Child, Elderly etc voorzien om aan te geven dat het om een DataSubject gaat dat zelf geen Consent kon geven. De relatie tussen de Entiteit en het Datasubject volgt uit Entiteit.hasRelationWith.
- OPMERKING: De aard van de relatie kan dan weer niet worden meegegeven. DPV voorziet wel subklassen van DataSubject als ParentsOfDataSubject, maar dat gaat ervan uit deze dus het DataSubject zijn, ttz dat het hun PersoonlijkeData is die gebruikt wordt in de plaats van hun kind.
- DPV heeft een ganse subclassificatie van Datasubjecten. We gaan ervan uit dat deze via Datasubject.type kunnen worden beschreven. Dit attribuut voegden we eerder al toe met een gelijkaardig doel (zie hoger).
- TODO: relatie tussen Entiteit en Agent toelichten, toevoegen aan diagram.
- Ipv met OSLO-Consent:Consent.isContingentOn (dat er blijkbaar vooral is om Expiry te beschrijven) te werken voegden we uit DPV .hasDuration en hasFrequency toe.
- TechnicalAndOrganisationalMeasures modelleren we dmv een enumeratie, DPV gebruikt hiervoor subklassen.
- De attributen PersonalDataHandling.personalDataCategory/type hebben we geschrapt. Ze kwamen oorspronkelijk uit DPV waar ze intussen vervangen zijn door een subtypering van PersonalData. We vervangen deze subclassificatie door een enumeratie PersonalDataType via een attribuut PersonalData.type.
- Het attribuut PersonalDataHandling.liveData geeft aan of de dataverwerking gebeurt van livedata of van een snapshot in de tijd. Dit is een kenmerk van de PersonalData en hebben we daarom verhuisd naar de klasse PersonalData.
- Volgens Gconsent was PersonalData een metaklasse. Is niet zo in DPV en lieten we daarom weg.
- Right en Risk komen uit DPV maar hebben geen attributen in OSLO-Consent. Er is ook een

classificatie voor voorzien. We realiseren die via een type attribuut bij beide klassen.

- OPMERKING: DPV ziet voor Right en Risk attributen zoals hasImpact, hasSeverity etc. Deze zijn momenteel niet voorzien in het remapped2DPV model.
- In Gconsent is de DataProcessor een specialisatie van ThirdParty (zie bvb §4.3 in Gconsent). In DPV is dat niet zo.
- In DPV is de DataProcessor direct met de PersonalDataHandling verbonden, in Gconsent via Processing.involvesThirdParty.
- We passen het huidige model aan aan DPV: PersonalDataHandling.hasDataProcessor toegevoegd en DataProcessor is geen rechtstreeks subtype meer van ThirdParty.
- De klasse PersonalDataHandling heeft ook een relatie hasRecipient, om aan te geven aan wie de data ontsloten is. Dit kan de DataController zijn en/of de DataProcessor of evt zelfs een ThirdParty (zie punt 9 in artikel 4 vd GDPR).
- We voegden deze relatie toe en dus ook de klasse Recipient. TODO: checken of dit nodig is. Evt dan ook ThirdParty terug toevoegen? Want is naast DataController en DataProcessor een mogelijke Recipient. Zie §Recipient in de Primer.
- DataRetention is in DPV geen kenmerk vd PersonalData maar van Processing en het wordt StorageCondition genoemd. We pasten het model overeenkomstig aan.
- OSLO-Consent:DataRetention.expiryDate, .expiryTime, .startDate stemmen overeen met DPV:DataStorageCondition vh type StorageDuration. TODO: StorageCondition subklassen of beperken tot StorageDuration.
- Het attribuut DataRetention.processing is overbodig gezien PersonalDataHandling.hasProcessing.
- Analoog voor LegalBasis?
- En het attribuut liveData zit al bij PersonalData.
- TODO: Agent terug toevoegen.
- TODO: Filter. Komt niet voor in DPV.
- TODO: Illustratieve enumeraties.