**Proposal for PbD-SE Discussion – Jan 9, 2013 – Agenda Item #3**

1. International use case and UML asset collection from the Healthcare standards domain to map PbD principles to privacy controls/processes/services and then to UML specifications.

Purpose: To leverage, aggregate, synthesize, and substantially extend existing work so that we may more quickly generalize from a domain where there has been foundational privacy standards work to all domains.

**Outline of approach:**

(1) Consider the PbD principles one at a time:

PbD Principle 3: Privacy Embedded into Design

*PbD* is embedded into the design and architecture of IT systems and business practices. It is not bolted on as an add-on, after the fact. The result is that privacy becomes an essential component of the core functionality being delivered. Privacy is integral to the system, without diminishing functionality.

It may be mapped directly to NIST 800-53 Appendix J’s AR-7, see <http://csrc.nist.gov/publications/drafts/800-53-rev4/sp800-53-rev4-ipd.pdf>, page J-9.

(2) Consider a major privacy process – Consent (may be described as an Agreement service e.g. in OASIS PMRM)

(a) Collect public standards assets for repository: e.g. HL7-based Consent Directive and HITSP Consent Directives package

(b) Collect use cases

Examples: <http://wiki.hl7.org/index.php?title=Consent_Directive_Use_Cases#Privacy_Consent_Directive_Use_Cases>

From Page 13, e-Health Ontario Consent Directive: Business View of Message Model for querying user consent – UML diagram showing a set of attributes that may be returned from across different classes when a query to view a user consent directive is issued. Attributes returned are Consent ID, User ID, Purpose of Use, Directive, Consent Type, RestrictDateTimeRange, DomainRestriction, DomainRestrictionQualifier

(c) Collect General Privacy Controls

Examples: NIST IP-1 Consent, Pg. J-15, <http://csrc.nist.gov/publications/drafts/800-53-rev4/sp800-53-rev4-ipd.pdf>

(d) With (a), (b), (c) as input, we can attempt to standardize consent privacy requirements at the software engineer’s level:

Example (as a start):

The software engineer shall provide means:

1. to authorize, (and/or automatically check for authorization for) the collection, use, maintaining and storage, sharing, and retention of PII prior to its collection,
2. for users to understand the consequences of accepting or declining the authorization of collection, use, storage and maintenance, dissemination, and retention of PII,
3. to obtain consent for new uses or disclosures of previously collected PII,
4. to ensure that any PII collection, storage and maintenance, dissemination, and retention aligns with current public notice of how PII is handled in the organization,
5. to ensure that individuals are aware of and consent to all uses, storage and maintenance, dissemination, and retention of PII not initially described in the public privacy notice that was in effect at the time the organization collected the PII.
6. to ensure that individuals providing consent are of legal age.

and so on..

We must also capture that the software engineer should incorporate in the above (a) the domain boundaries that encapsulate PII (PMRM can provide useful guidance here), (b) clear purpose specification, (c) data minimization, (d) knowledge that there may be third party-validation and verification of provided data etc.

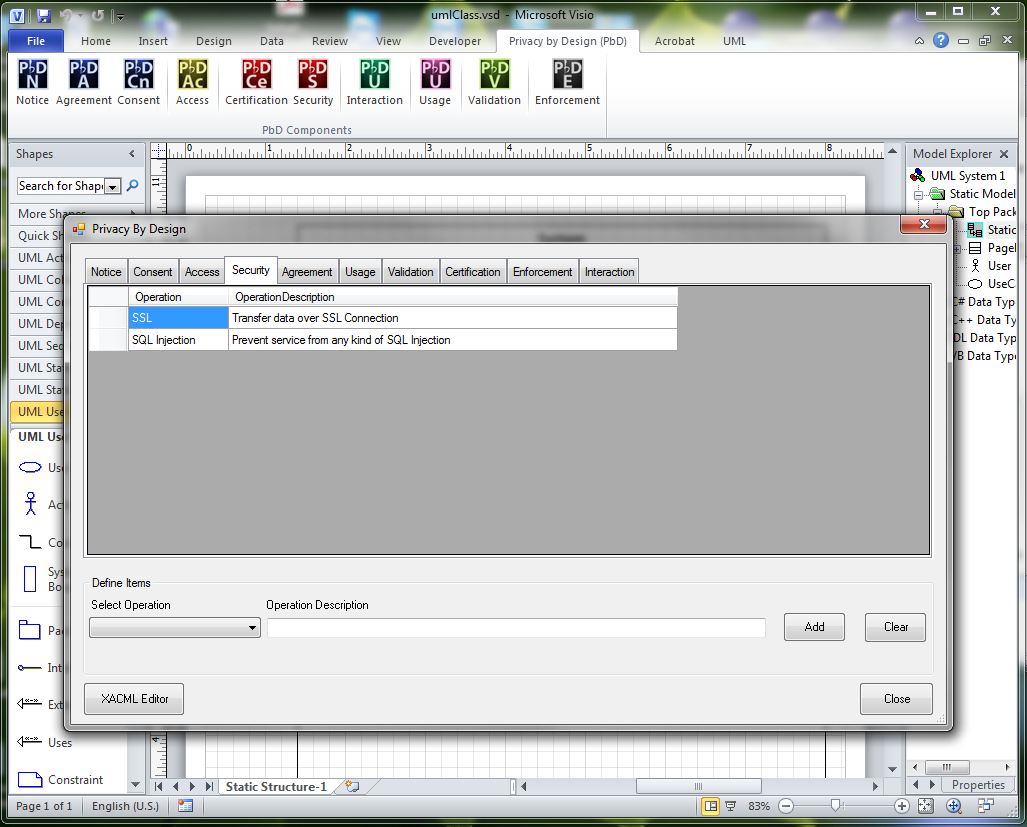
Purpose: With (d), software engineers will quickly grasp UML diagrams for privacy and will be able to pull them into their software designs.

(e) Compile and synthesize UML assets for the consent privacy process. Use methodologies such as OASIS PMRM to update and verify correctness of UML diagrams.

(3) Refine steps in (2), and repeat (2) for other privacy processes/services.

(4) Repeat for other PbD principles e.g. Privacy by default.

The below shows 3 screenshots to communicate my initial vision to add privacy functionality into UML diagramming tools, such as MS Visio, in order to educate, enhance programmer awareness, and increase privacy productivity for the software engineer. [A student programmer, S. Ali, executed the Visio implementation. I am not championing any particular vendor product. The illustration platform was simply available]. A future logical step is the provision of privacy APIs for increasing software developers’ privacy productivity.



Sc

