

2016-07-07 Meeting notes

Date

07 Jul 2016

Attendees

- [Lynn Calahan \[X\]](#)
- [Jennifer Bond-Caswell](#)
- [Michael Uschold](#)
- [Dennis Wisnosky](#)
- [Unknown User \(mark_friend\)](#)
- [Lucy Opsitnik](#)
- [Viesturs Lenss \[X\]](#)
- [@gary?](#)

Agenda

[FIBO Loans FCT Meeting Agenda_070716.doc](#)

- 1) Use Case reminder.
- 2) Where we are on our road map.
- 3) Open Action Items
- 4) JIRA Issues Review

<https://jira.edmcouncil.org/browse/LOAN/?selectedTab=com.atlassian.jira.jira-projects-plugin:issues-panel>

- 5) Todays content discussion.

[MU-ModelingDiscussion-2016-06-30.pptx](#)

UML

Spreadsheet [LoansAnnotations Integrated_review070716.xlsx](#)

Protégé [loanCore-2016-07-06.rdf](#)

- 6) For next week.

Proceedings

SUMMARY:

Origination: Group prefers to have two properties for origination because there are two different roles, one of the (one or more) persons doing the originatin, and one for the organization that is involved in the origination. The fact that the roles are closely related does not justify using a single hasOriginator property. Everyone was happy with this.

- MU will make changes to ontology and suggest specific definitions to be reviewed.

hasGiver/hasGetter vs. hasObligor/hasObligee

No one likes the terms hasGiver and hasGetter. In the context of Commitments, the terms hasObligor and hasObligee have more specific meaning, and can be used instead.

MU will replace hasGiver/hasGetter with hasObligor/hasObligee and will make recommendation to FND.

hasPayor/hasPayee

The group recommendation from a few weeks ago was to use hasPayor/hasPayee for the obligation to make loan payments.

MU said that because the obligation to make loan payments is in fact an Obligation, there is therefore an Obligee and an Obligor. This was objected to, the feeling was that MU was trying to force the generic use of one property rather than the more specific examples recommended by the group.

MU attempted to clarify. Just like one can have a parent w/o saying what the gender is, there are more specific properties when needed: hasMother and hasFather. So you can be both a mother and a parent at the same time. By analogy, a more specific kind of Obligee is a Payee and a more specific kind of Obligor is a Payor, not one or the other, but both. So MU's proposal was not to replace Payor/Payee with Obligor/Obligee, but to ask whether it could be both.

JC made the observation that for an obligation to make loan payments, there is no payor or payee until an actual payment is made.

MU: this is an important distinction. Strictly, the payor and payee are properties of the payment events that discharge the obligations to make loan payments. It could be modeled that way, or not.

The group generally wanted new properties each with a very specific meaning, so they wanted to not (just) use hasObligee/hasObligor on the obligation to make loan payments, they wanted to use hasPayee/hasPayor.

On later reflection (after the meeting), MU noticed a problem.

If as JC says, there is no payee or payor on an obligation, (that it really is about that payment itself), then you should not be using hasPayor and hasPayee on the obligation. If you want to have a separate property that hangs off an Obligation that is specific to paying not just obligee/obligor, then a more accurate name would be hasIntendedPayor and hasIntendedPayee. The intended payee is exactly the obligee on that obligation, and the intended payor is exactly the obligor on that obligation (just like the mother is the parent). This can work.

- MU will rework this example this way and propose back to the group

Obligation to Report:

?: Nobody thinks of this as an obligation, it is a requirement under the law.

MU: that is fine, and we don't have to model it as an obligation if we don't want to, but semantically, that's exactly what it is. In English, if you are required to do something then you are obliged to it. I agree with Mike Bennets point about being first clear on what something means, and then decide how best to model it and later build applications.

LC?: it does not belong on the LoanContract

MU: ok, good point. [It turns out the current model says the LoanContract produces the obligation, so it is connected to but not part of the loan contract. I think this is correct]

- MU to reflect on this, make suggestion to group

hasCosts

LC: (in MU words from recollection) there are a number of costs that are not part of the loan contract, they are up front costs that must be paid in order for the loan contract to exist. Regulations require consistent price disclosures that can be compared across lenders, including various fees for various services.

MU: ok, this is another important distinction. These are still obligations to pay, but they are not produced by the loan contract, they are a pre-condition for it to exist.

LC: the lender and originator are bound by the list of costs, but they are not obligations until papers are signed.

MU: So it is essentially an offer, which becomes a template for a contract if one is signed. It includes a list of line items that are, in effect 'for sale' in a package that may result in a loancontract.

- MU to reflect on this distinction between the cost disclosures which are more like an offer, and the obligations that spawn from the contract.

Decisions

Obligor/Obligee shall remain connected to ObligationToFund

Continue with group recommendation for separate Obligor/Obligee and Payor/Payee party relationships at this time

Consider option of creating ObligationToPay class with sub-classes offline and bring back to group

Action items

- Michael Uschold on Origination MU will make changes to ontology and suggest specific definitions to be reviewed.
- Michael Uschold MU will replace hasGiver/hasGetter with hasObligor/hasObligee and will make recommendation to FND
-

- [Michael Uschold](#) On has payor/has Payee MU will rework this example this way and propose back to the group
- [Michael Uschold](#) On Obligation to Report MU to reflect on this, make suggestion to group SEE EMAIL STING ON THIS ONE
- [Michael Uschold](#) On hasCosts MU to reflect on this distinction between the cost disclosures which are more like an offer, and the obligations that spawn from the contract
- [Michael Uschold](#) - Add hasPayor and hasPayee to ObligationToPay
- [Lynn Calahan \[X\]](#), [Jennifer Bond-Caswell](#), [Michael Uschold](#) - review Obligor/Obligee and Payor/Payee concepts and relationships for appropriate construction and relationships
-