

LemonTree
Sequenzdiagramme
Diff/Merge Schwierigkeiten

LieberLieber Software GmbH

Inhalt

1. Einleitung.....	3
2. Herausforderungen	4
2.1 Einfügen, Löschen und Verschieben von Nachrichten an verschiedenen Positionen	4
2.1.1. Einfügen von Nachrichten	4
2.1.2. Löschen von Nachrichten	5
2.1.3. Verschieben von Nachrichten	5
2.1.4. Workaround	6
2.2 Hinzufügen von Nachrichten an derselben Position	6
2.2.1 Workaround	6
2.3 Seiteneffekte am Sequenzdiagramm	7
2.3.1 Workaround	8
2.4 Fragmente.....	8
2.4.1 Workaround	8

1. Einleitung

LemonTree ist ein Diff and Merge Tool zur Versionsverwaltung von Modellen, die durch die Modellierungssprachen wie UML, SysML, BPMN etc. definiert sind. Es unterstützt Änderungen in Modellen nachvollziehbar nachzuverfolgen, und ermöglicht die parallele Zusammenarbeit mehrerer Teams oder Personen. LemonTree wird nahtlos in bestehende Versionierungssysteme, wie Git oder SVN integriert.

LemonTree unterstützt ebenfalls den Diff/Merge von Sequenzdiagrammen (Teil des UML Standards). UML Sequenzdiagramme sind Verhaltensdiagramme, welche unter anderem Interaktionen durch den Austausch von Nachrichten zwischen Objekten mittels Lebenslinien grafisch darstellen. Jedoch gibt es im Moment bei der Verwendung von Sequenzdiagrammen in LemonTree noch einige Herausforderungen auf, die beachtet werden müssen. In diesem Dokument werden die Probleme spezifiziert und Hinweise gegeben, wie im Moment damit umgegangen werden kann. Dazu wird ein Demonstrationsbeispiel zweimal verändert und anschließend mit LemonTree gemerged.

Das Demonstrationsbeispiel ist ein vereinfachter Ablauf eines Bankomaten. Dabei werden drei verschiedene Objekte angenommen: der Kunde, der Bankomat und die Bank. Zuerst fügt der Kunde die Bankomatkarte in den Bankomat ein. Anschließend wird der Pin abgefragt, welcher vom Bankomat an die Bank weitergeleitet wird. Wenn der Pin bestätigt ist, wird vom Bankomat die Auszahlungshöhe abgefragt. Nach der Eingabe des Kunden wird das Geld vom Bankomat ausgezahlt und die Karte ausgegeben. Abbildung 1 zeigt ein Sequenzdiagramm für den beschriebenen Prozess, welches als Basis dient.

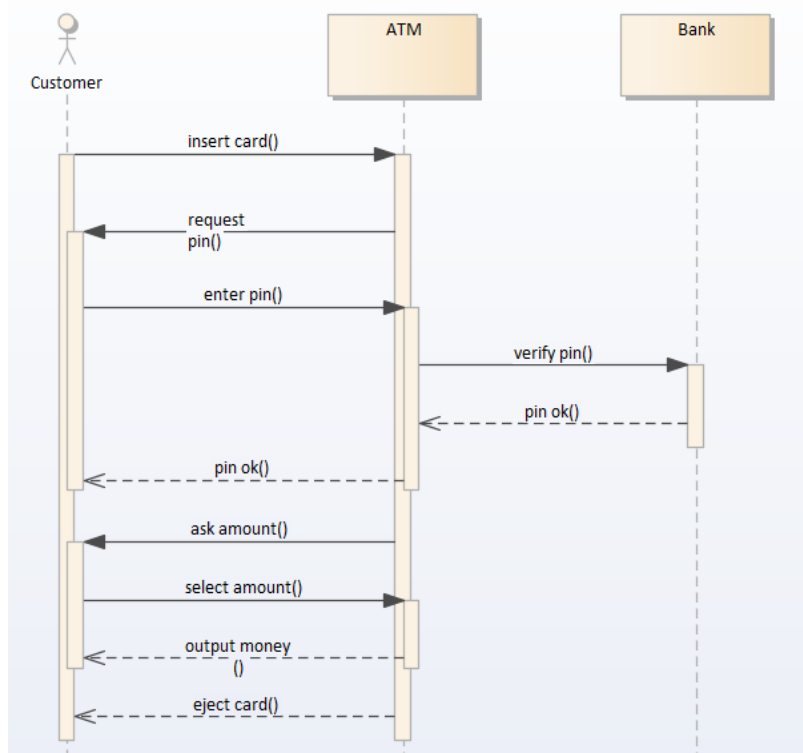


Abbildung 1: Vereinfachtes Sequenzdiagramm für die Bargeldbehebung an einem Bankomat

2. Herausforderungen

2.1 Einfügen, Löschen und Verschieben von Nachrichten an verschiedenen Positionen

2.1.1. Einfügen von Nachrichten

Durch das Einfügen und Löschen von Nachrichten können verschiedene unerwünschte Konflikte auftreten. Abbildung 2 zeigt ein Beispiel, wo auf der linken Seite **eine** Nachricht nach `insert(card)` und auf der rechten Seite **zwei** Nachrichten nach `pin ok()` eingefügt werden. Dies sollte aus logischer Sicht zu keinem Konflikt führen. Allerdings führt dies zu Konflikten bei allen Nachrichten, die sich nach der letzten eingefügten Nachricht befinden. In diesem konkreten Beispiel betrifft es die letzten vier Nachrichten.

Das Problem ist, dass durch das Einfügen einer Nachricht auch Parameter aller folgenden Nachrichten geändert werden. Passiert daher eine Änderung auf beiden Seiten, so kommt zu Konflikten.

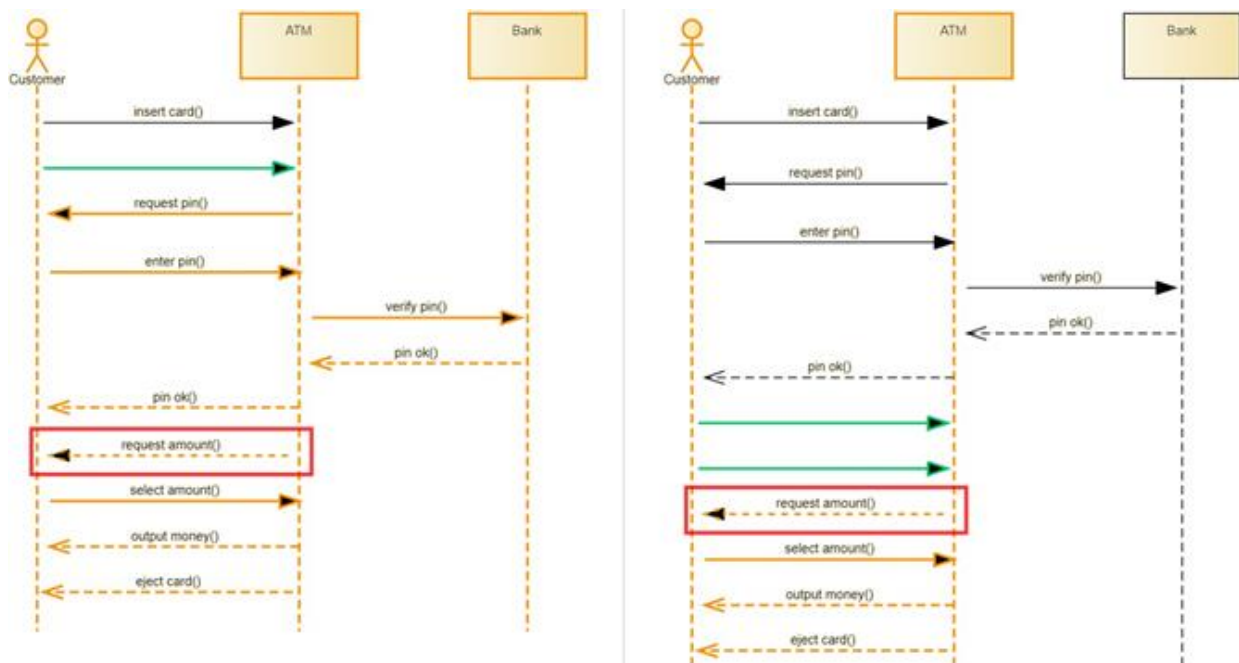


Abbildung 2: Konflikt hervorgerufen durch Einfügen verschiedener Nachrichten

<p>▲ PtEndY 367-390</p>	<p>▲ PtEndY 367-435</p>
<p>▲ PtStartY 367-390</p>	<p>▲ PtStartY 367-435</p>
<p>▲ SeqNo 78</p>	<p>▲ SeqNo 79</p>

Abbildung 3: Konflikte in Attributen durch mehrfaches Hinzufügen von Nachrichten

Abbildung 3 zeigt, dass folgende Attribute betroffen sind:

- PtStartY
Dieses Attribut beschreibt die Startposition auf der Y-Achse, bei welcher die Nachricht startet.
- PtEndY
Dieses Attribut beschreibt die Endposition auf der Y-Achse, bei welcher die Nachricht startet.
- SeqNo
Dieses Attribut beschreibt die Position der Nachricht in Relation zu den anderen Nachrichten.

PtStartY und PtEndY können auch als grafische Höhenangabe der Nachricht verstanden werden. Die SeqNo kann als die Reihenfolge der Nachrichten interpretiert werden.

2.1.2. Löschen von Nachrichten

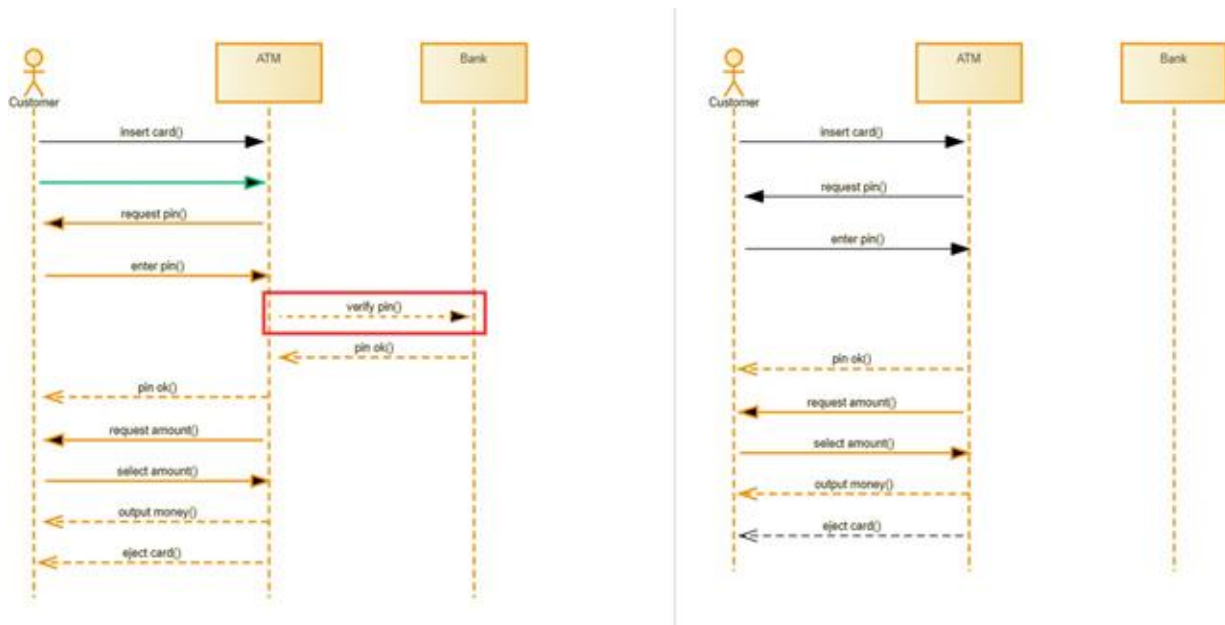


Abbildung 4: Konflikt durch die Kombination von Einfügen und Löschen von Nachrichten

Ähnliche Konflikte können durch das Löschen von Nachrichten auftreten, da auch hier dieselben Attribute bei den darauffolgenden Nachrichten verändert werden (siehe Abbildung 4). Im Unterschied zum Einfügen werden hier die Attributwerte kleiner, da Elemente nicht mehr vorhanden sind. Abhängig von der genauen Interaktion, könnte auch nur die SeqNo betroffen sein.

2.1.3. Verschieben von Nachrichten

Gleiche Konflikte können durch das Verschieben von Nachrichten hervorgerufen werden. Abbildung 5 zeigt ein Beispiel, wo die SeqNo nicht betroffen ist, aber PtStartY und PtEndY verändert werden.

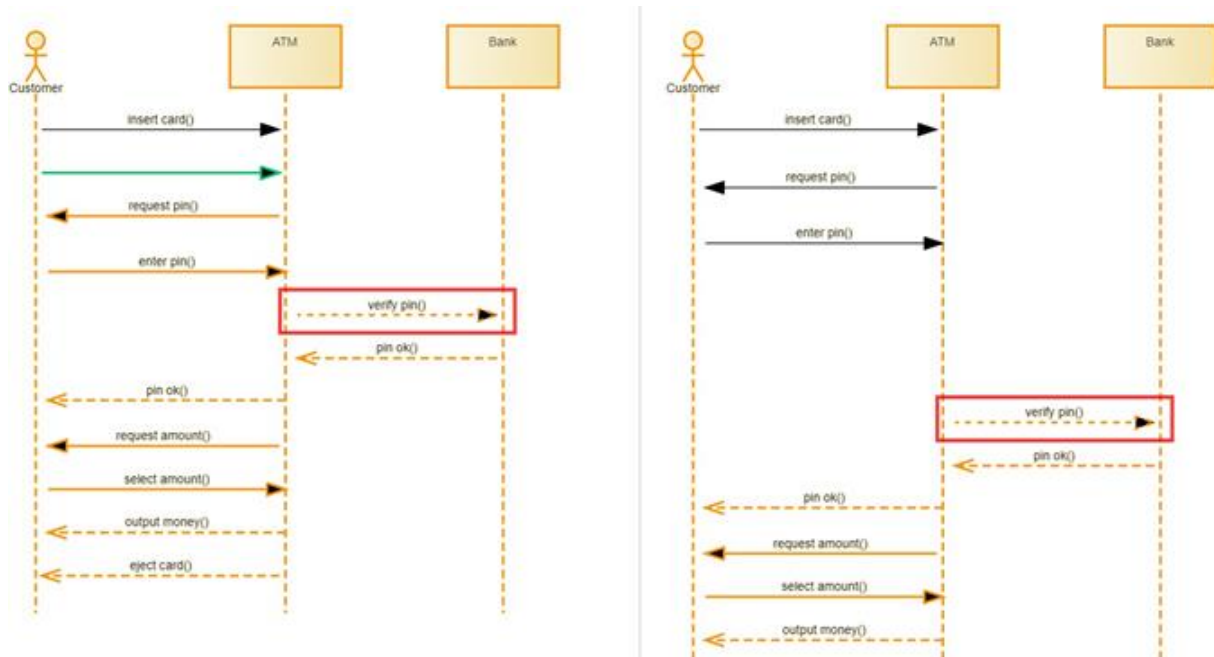


Abbildung 5: Konflikt durch die Kombination und Verschieben von Nachrichten

2.1.4. Workaround

Allgemein stellt die vertikale Verschiebung von allen folgenden Nachrichten, welche durch Hinzufügen, Löschen oder Verschieben ausgelöst werden können, ein Problem für die korrekte Anzeige in LemonTree dar, sobald dies auf beiden Seiten passiert.

Aktuell gibt es dafür keinen einfachen Workaround um den Konflikt zu lösen. Man muss eines der beiden Diagramme auswählen und die fehlenden Informationen ergänzen.

2.2 Hinzufügen von Nachrichten an derselben Position

Wird zufällig genau an derselben Position jeweils eine Nachricht hinzugefügt, dann wird kein Konflikt angezeigt. Abbildung 6 zeigt genau einen solchen Fall. Es sollte eigentlich ein Konflikt angezeigt werden, da LemonTree nicht wissen kann, welche der beiden Nachrichten zuerst kommt. Gleichartige Probleme können auch beim Einfügen von mehreren Nachrichten an derselben Position auftreten.

2.2.1 Workaround

Da LemonTree in diesem Fall keinen Konflikt anzeigt, können die Diagramme einfach gemerged werden. Allerdings sollte der Modellierer danach die richtige Reihenfolge herstellen. Aktuell wird beim Merge einfach die grafische Position übernommen und dadurch die Ordnung bestimmt. Befinden sich zwei Nachrichten auf genau derselben Position, bleibt dies im zusammengeführten Modell ebenso. Alternativ, falls nur eine der beiden Nachrichten übernommen werden sollte, könnte der Modellierer dies schon beim Merge in LemonTree mit „Take this side“ entscheiden.

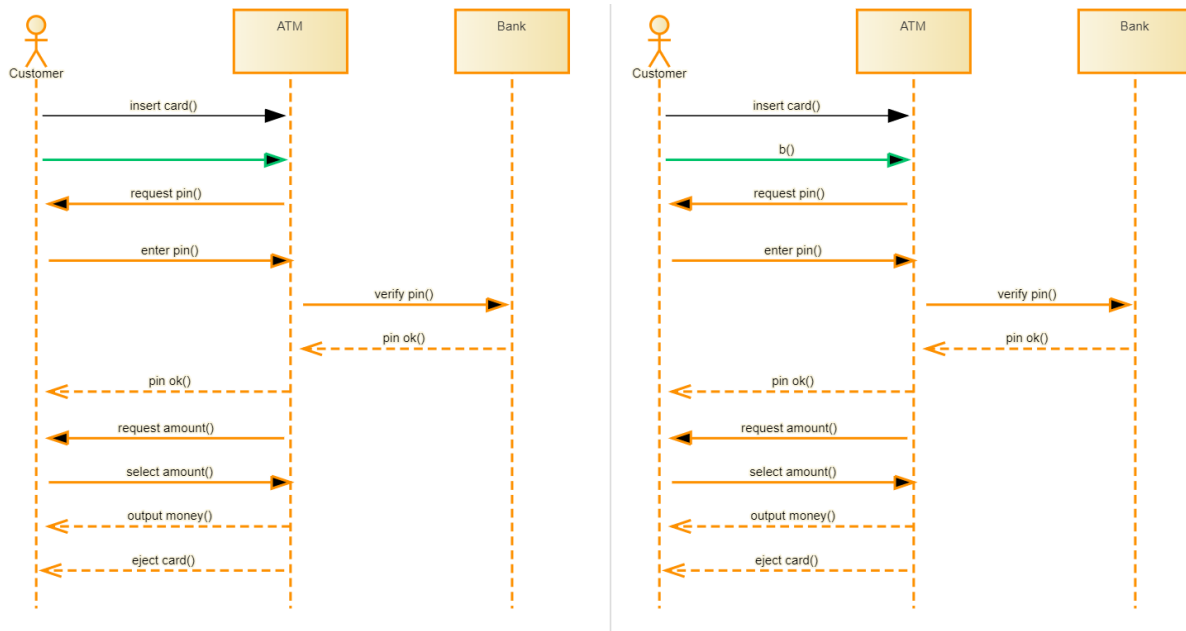


Abbildung 6: Kein Konflikt beim Einfügen von Nachrichten an derselben Position

2.3 Seiteneffekte am Sequenzdiagramm

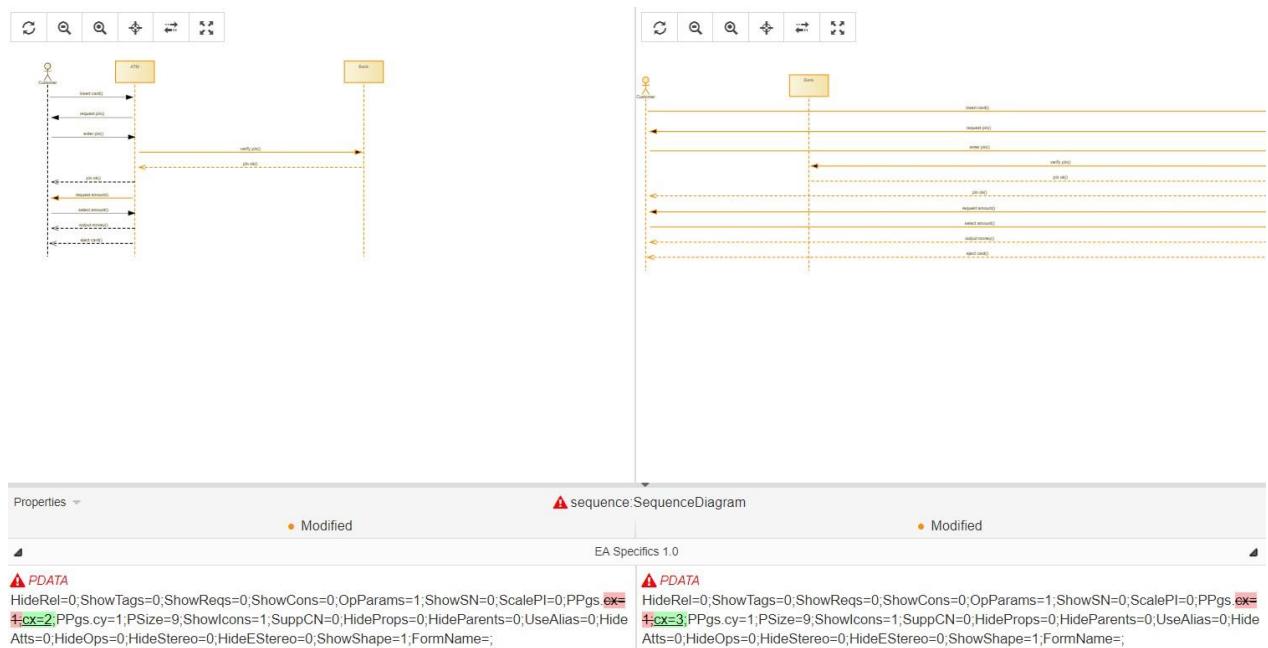


Abbildung 7: Änderung des Seitenindex durch Verschieben der Lebenslinien

Es können ebenfalls Konflikte durch das Verschieben von Objekten (Lebenslinien Aktoren, etc.) auf der X-Achse (horizontal) auftreten. Diese Änderungen führen nämlich zu Anpassungen im übergeordneten Sequenzdiagramm.

Zum Beispiel wird im **PDATA** Attribut des Sequenzdiagramms unter anderem der Seitenindex gespeichert, welcher durch das Verschieben von Objekten vergrößert oder verkleinert werden kann. Ein Beispiel für diesen Konflikt ist in Abbildung 7 dargestellt.

Ein solcher Konflikt kann ebenfalls durch das Hinzufügen oder Löschen von Objekten hervorgerufen werden. Außerdem könnten noch andere Aktionen Änderungen im PDATA hervorrufen, welche dann zu Konflikten führen können.

2.3.1 Workaround

Im Moment kann zur Auflösung des Konflikts eines Seitenindexes einfach der größere Wert übernommen werden. Allgemein gilt hier, dass der Modellierer nach dem Merge gegebenenfalls Änderungen nachbessern muss.

2.4 Fragmente

Beim Einsatz von Fragmenten kann es zu einer Vielzahl an diversen Problemen kommen.

Erstens, kann das Verschieben von Lebenslinien zu demselben Problem führen, welches schon in Sektion 2.1 beschrieben ist. Weiteres können viele Situation auftreten, in denen kein Konflikt angezeigt wird, jedoch ein sinnvoller Merge nicht möglich ist. Beispiele hierfür sind unter anderem Überlappungen von Fragmenten, Verkleinern (Elemente fallen aus dem Fragment raus) oder Vergrößern (Elemente werden hinzugenommen).

Abbildung 8 zeigt ein Beispiel für überlappende Fragmente, welche separat an verschiedenen Positionen eingefügt wurden.

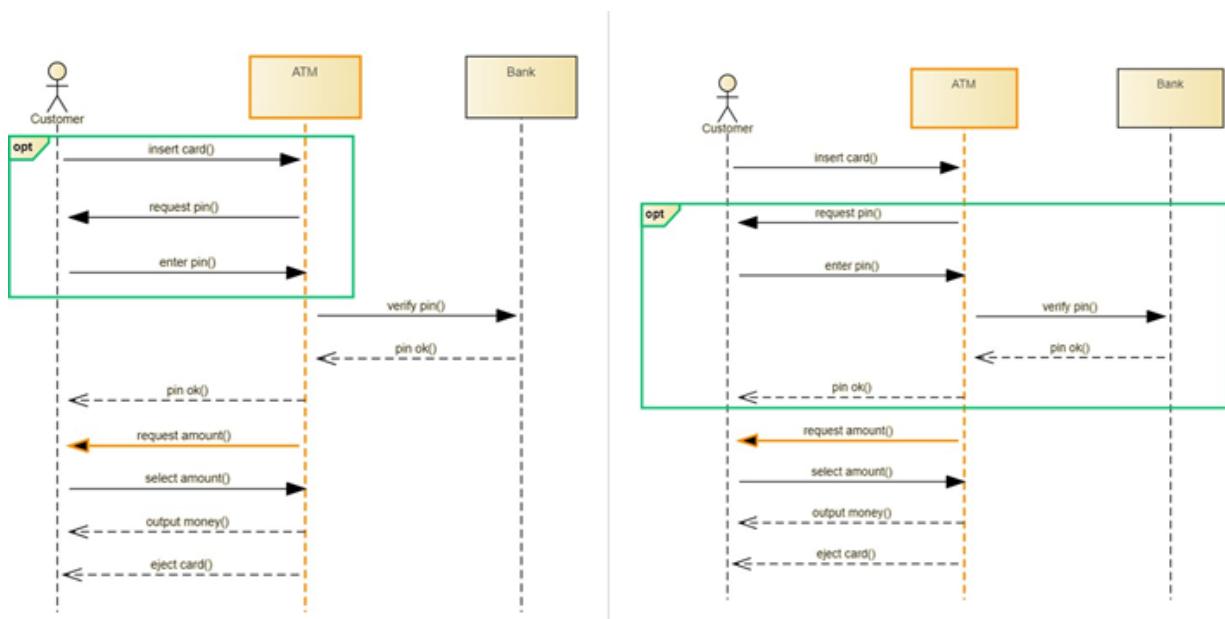


Abbildung 8: Einfügen von überlappenden Fragmenten

2.4.1 Workaround

Im Moment kann bzgl. Fragmenten nur der Modellierer selbst nach dem Merge den richtigen Workflow und logischen Ablauf wiederherstellen.