

Richtlinien Swissdec Standards: Anforderungen Darstellung DialogMessages

Die Richtlinien wurden in Zusammenarbeit mit folgenden Beteiligten erarbeitet:

- Suva
- Schweizerischer Versicherungsverband

Herausgeber

Swissdec
Postfach 4358
6002 Luzern

www.swissdec.ch

Inhaltsverzeichnis

1.	Einleitung.....	6
2.	DialogMessage in ELM.....	6
2.1	Test und Abnahme	6
2.1.1	Operationen und DialogMessage-Prozess	6
2.1.2	Darstellung der DialogMessages	7
3.	DialogMessage in KLE.....	8
3.1	Test und Abnahme	8
4.	Darstellung von standard und von freien DialogMessages.....	9
4.1	Standard und freie DialogMessages	9
4.2	Elemente der Story DialogMessage	9
4.3	Abfolge bei der Darstellung von DialogMessages	12
4.4	Antwort als Reactor mittels einer DialogMessage	13
4.4.1	Antwort auf eine DialogMessage mit enthaltenen Paragraph/Answer-Elementen	13
4.4.2	Verkettung von DialogMessages	15
5.	Standard DialogMessages in KLE	16
5.1	SimpleMessage «Mitteilung»	17
5.2	TaskWithDeadline «Aufgabe»	19
5.1	TaskWithConfirmation «Anfrage mit Bestätigung»	20
5.1	ContactPerson «Kontaktperson Anfrage»	21
5.2	WorkplaceDescription «Arbeitsplatz-Beschreibung»	22
6.	Anhang.....	24
6.1	Referenzen	24

Abbildungsverzeichnis

Abbildung 1: Auswahl Komplettierung	6
Abbildung 2: DialogMessage XML-Schema Struktur.....	9
Abbildung 3 Vorgehen bei der Darstellung der DialogMessage (Aktivitäts Diagramm)	12
Abbildung 4 Paragraph/Answer/DateTime-Element, andere Answer/<Type>-Elemente entsprechend	13
Abbildung 5 Struktur des DialogMessage/Previous-Elementes.....	15
Abbildung 6 Verkettung von DialogMessages	15
Abbildung 7: Beispiel einer Actor-GUI-Darstellung der DialogMessage «Mitteilung»	17
Abbildung 8: Beispiel einer Reactor-GUI-Darstellung der Standard DialogMessage «Mitteilung»	17
Abbildung 9: Beispiel XML-Instanz «Mitteilung» analog (REFBSPDIA-IDCH, 2018) 1_SimpleMessageActor.xml	18
Abbildung 10: Beispiel einer Reactor-GUI-Darstellung der DialogMessage «Aufgabe»	19
Abbildung 11: Beispiel einer Reactor-GUI-Darstellung der DialogMessage «Anfrage mit Bestätigung»	20
Abbildung 12: Beispiel einer Reactor-GUI-Darstellung der DialogMessage «Kontaktperson Anfrage».....	21
Abbildung 13: Beispiel Reactor-XML-Instanz Struktur der Standard DialogMessage «Kontaktperson Anfrage» ...	21
Abbildung 14: Beispiel einer Reactor-GUI-Darstellung der DialogMessage «Arbeitsplatz-Beschreibung» Teil 1...	22
Abbildung 15: Beispiel einer Reactor-GUI-Darstellung der DialogMessage «Arbeitsplatz-Beschreibung» Teil 2...	23

Übersicht der Änderungen

Kapitel	Änderung
Erste Version der Dialog-Message für ELM und KLE	

Konventionen in diesem Dokument

Folgende Schriftarten werden in diesem Dokument verwendet:

Text	Dokumentation
Text	Code
<Text>	XML-Element
(TEXT)	Referenz auf ein anderes Dokument

Die Verbindlichkeit von Anforderungen ist wie folgt definiert:

Verbindlichkeit	Wort
Pflicht	<i>muss, darf nicht</i>
Wunsch	<i>soll (sollte)</i>
Absicht	<i>wird</i>
Vorschlag	<i>kann</i>

Tabelle 1: Verbindlichkeit von Anforderungen

Achtung:

Für das konzeptionelle Verständnis genügen oft ältere Schemabilder, d.h. **verbindlich** sind immer nur die offiziellen¹ **XML-Files**.

¹ www.swissdec.ch

1. Einleitung

Dieses Dokument enthält Anforderungen zur Darstellung und Verarbeitung der DialogMessage, welche im Rahmen diverser Swissdec-Standards eingesetzt wird. Aktuell ist DialogMessage Teil von KLE (1.0) und ELM (5.0)

Eine DialogMessage wird dazu verwendet, einen Dialog zwischen verschiedenen Teilnehmern eines Standards zu ermöglichen. Dabei werden Nachrichten Schema-valid zwischen z.B. Unternehmen und Versicherer versendet. Abhängig vom verwendeten Standard bietet DialogMessage Möglichkeiten vom Austausch einfacher Informationen bis hin zur Umsetzung einer gesicherten Messenger-Applikation.

Hierbei wird zwischen Standard-DialogMessages und freien DialogMessages unterschieden.

Standard DialogMessages besitzen eine spezifizierte Struktur mit genau definierten Ids, freie DialogMessages setzen sich beliebig aus den Schema-Elementen der DialogMessage zusammen.

Freie DialogMessages erhöhen die Flexibilität des swissdec Leistungsstandards, indem sie neue Strukturen von Fragen und Antworten zwischen den beteiligten Partnern ermöglichen, ohne neue Versionen des swissdec Standards abwarten zu müssen. Falls sich ein Typ einer freien DialogMessage als allgemeiner Bedarf herausstellt, kann dieser Typ bei einer neuen Version dieses Dokuments als neue Standard DialogMessage spezifiziert werden.

Dieses Dokument enthält die Anforderungen an das generelle Vorgehen zur Darstellung von DialogMessages sowie Anforderungen an die Darstellung von vorgegebenen Standard DialogMessages.

An dieser Stelle sei darauf hingewiesen, dass Ziele und grundlegende Anforderungen der Story DialogMessage in den Richtlinien des Leistungsstandards (RL-IDCH, 2017), Kapitel 5, beschrieben sind und dieses Dokument eine Ergänzung dazu ist. Ohne die Lektüre der Ziele und Anforderungen an die Story DialogMessage in den Richtlinien (RL-IDCH, 2017) ist das Verständnis der Story DialogMessage für den Leistungsstandard unvollständig. Der Gebrauch von DialogMessage im Lohnstandard wird hier in seinem vollen Umfang beschrieben.

2. DialogMessage in ELM

2.1 Test und Abnahme

In ELM wird DialogMessage in eigenständigen Operationen abgehandelt, welche jeweils vom Transmitter ausgelöst werden. Es handelt sich hier um SOAP-Webservice-Aufrufe analog den anderen Prozesselementen einer ELM-Übermittlung. Die richtige Darstellung, sowie die richtige Durchführung eines Dialogs werden von Swisdec geprüft.

Auf Endreceiver-Seite steht es dem Endempfänger frei, welche Form von DialogMessage er verwendet. Er kann auf Standard-Messages zurückgreifen oder eigene, freie Messages definieren. Einschränkungen für bestimmte Domänen werden separat publiziert.

Standard-Messages in ELM werden zu gegebener Zeit von Swisdec publiziert.

Auf Transmitter-Seite **sollen** momentan keine Standard oder freien DialogMessages verfasst werden. Der Transmitter **soll** nur jeweils auf die DialogMessages vom Endreceiver reagieren.

2.1.1 Operationen und DialogMessage-Prozess

In erster Linie soll DialogMessage im ELM ein Prozessschritt als Alternative zur bisher verwendeten Completion werden. Der Transmitter löst zwar die Operationen zum Abrufen der DialogMessage aus, ist im Prozess aber ausschliesslich ein Reactor, der auf die DialogMessages vom Endreceiver reagiert. Dies führt zu folgendem Aufbau des Prozesses:

1. Die Übermittlung wird wie gewohnt mit DeclareSalary() und getStatus() durchgeführt. In der Response der Statusmeldung findet sich eine Information des Endempfängers, wie nun weiter vorzugehen ist:

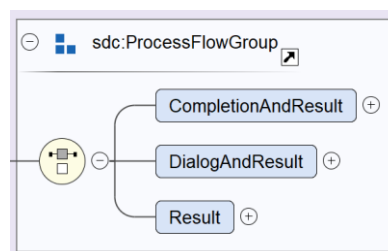


Abbildung 1: Auswahl Komplettierung

- a. CompletionAndResult: Der Endempfänger liefert dem Transmitter eine URL, über die auf dem Portal des Empfängers die Completion durchgeführt werden kann. Siehe ELM-Spezifikation, TransmitterRequirements.
 - b. DialogAndResult: Der Endempfänger ermöglicht Ergänzungen per DialogMessage.
 - c. Result: Der Endempfänger verzichtet auf Ergänzungen und Freigabe.
2. Im Falle von 'DialogAndResult' **muss** der Transmitter nach der Statusabfrage den DialogMessage-Prozess ausführen.
- a. Die Kommunikation wird mit getDialog() gestartet. Der Transmitter sendet dem Endreceiver die erhaltenen Login-Informationen (Credentials) und Informationen zur Empfänger-Institution.
 - b. Der Endreceiver antwortet mit getDialogResponse() und sendet in dieser Struktur die von ihm verlangten DialogMessages.
 - c. Der Transmitter reagiert mit ReplyDialog und sendet die verlangten Antworten auf die erhaltenen DialogMessages.
 - d. Der Endreceiver **kann** nun den Dialog beenden oder weitere DialogMessages an den Transmitter adressieren.

2.1.2 Darstellung der DialogMessages

Auf Transmitter-Seite **muss** ein generischer Renderer implementiert werden, welcher alle freien und Standard DialogMessages darstellen und beantworten können **muss**. Auch der Endreceiver **sollte** einen generischen Renderer implementieren, da in zukünftigen Versionen auch vom Transmitter her Mitteilungen per DialogMessage möglich werden könnten.

Anstelle des generischen Renderers **können** für Standard DialogMessages auf der Seite des Endreceivers spezielle Renderer implementiert werden.

ER [D] = Endreceiver mit Digitalisierungsbereich DialogMessages

	Transmitter als Reactor		ER [D] als Actor
	darstellen	Antworten wenn «Answer»- Element	verfassen und evtl. Antwort ver- arbeiten
Standard DialogMessage «Mitteilung»	muss	Nicht vorgesehen	kann
Weitere standard Dialog- Messages	muss	muss	kann
Freie DialogMessages	muss	muss	kann

Tabelle 2 Anforderungen zur Umsetzung von DialogMessages

3. DialogMessage in KLE

3.1 Test und Abnahme

In KLE wird DialogMessage als Story abgehandelt und innerhalb der Operation SynchronizeIncident() verwendet. Die richtige Darstellung der DialogMessage wird von Swissdec geprüft.

Sowohl auf Transmitter als auch auf Endreceiver-Seite **muss** die Standard DialogMessage «Mitteilung» vollständig umgesetzt werden.

Alle weiteren, in diesem Dokument spezifizierten Standard DialogMessages, **müssen** ebenfalls umgesetzt werden. Hier sind die untenstehende Tabelle und das Kapitel 5 «Standard DialogMessages» relevant. Ausführungen zu den Rollen Actor und Reactor siehe (RL-IDCH, 2017).

Auf Transmitter-Seite **sollen** momentan, ausser der «Mitteilung», keine Standard oder freien DialogMessages verfasst werden.

Sowohl auf Transmitter als auch auf Endreceiver-Seite, **muss** ein generischer Renderer implementiert werden, welcher alle freien und standard DialogMessages darstellen und beantworten können **muss**.

Anstelle des generischen Renderers **können** für standard DialogMessages spezielle Renderer implementiert werden.

ER [D] = Endreceiver mit Digitalisierungsbereich DialogMessages

	Transmitter als Actor	Transmitter als Reactor		ER [D] als Actor	ER [D] als Reactor	
	verfassen	darstellen	Antworten wenn «Answer»- Element	verfassen und evtl. Antwort ver- arbeiten	darstellen	Antworten wenn «Answer»- Element
Standard DialogMessage «Mitteilung»	muss	muss	Nicht vorgesehen	muss	muss	Nicht vorgesehen
Weitere standard Dialog- Messages	Nicht vorgesehen	muss	muss	kann	Nicht vorgesehen	Nicht vorgesehen
Freie DialogMessages	Nicht vorgesehen	muss	muss	kann	Nicht vorgesehen	Nicht vorgesehen

Tabelle 3 Anforderungen zur Umsetzung von DialogMessages

4. Darstellung von standard und von freien DialogMessages

4.1 Standard und freie DialogMessages

Mit DialogMessages kann in den Varianten standard und frei kommuniziert werden. Die Bezeichnung Standard-DialogMessage bezieht sich auf die in diesem Dokument spezifizierten Standard DialogMessages.

Standard DialogMessages werden mit generischen oder mit speziellen Renderern dargestellt. Freie DialogMessages werden nur mit einem generischen Renderer dargestellt. Ein generischer Renderer ist daher zentral für die Darstellung von DialogMessages.

4.2 Elemente der Story DialogMessage

Die Elemente (Datenstruktur) aus welchen sich eine DialogMessage zusammenstellen lässt, sind im zugehörigen Schema des Leistungsstandards (XSD-IDCH, 2017) und in den Richtlinien (RL-IDCH, 2017) und/oder im Schema des Lohnstandards V5.0 (XSD-SDCH, 2020) beschrieben.

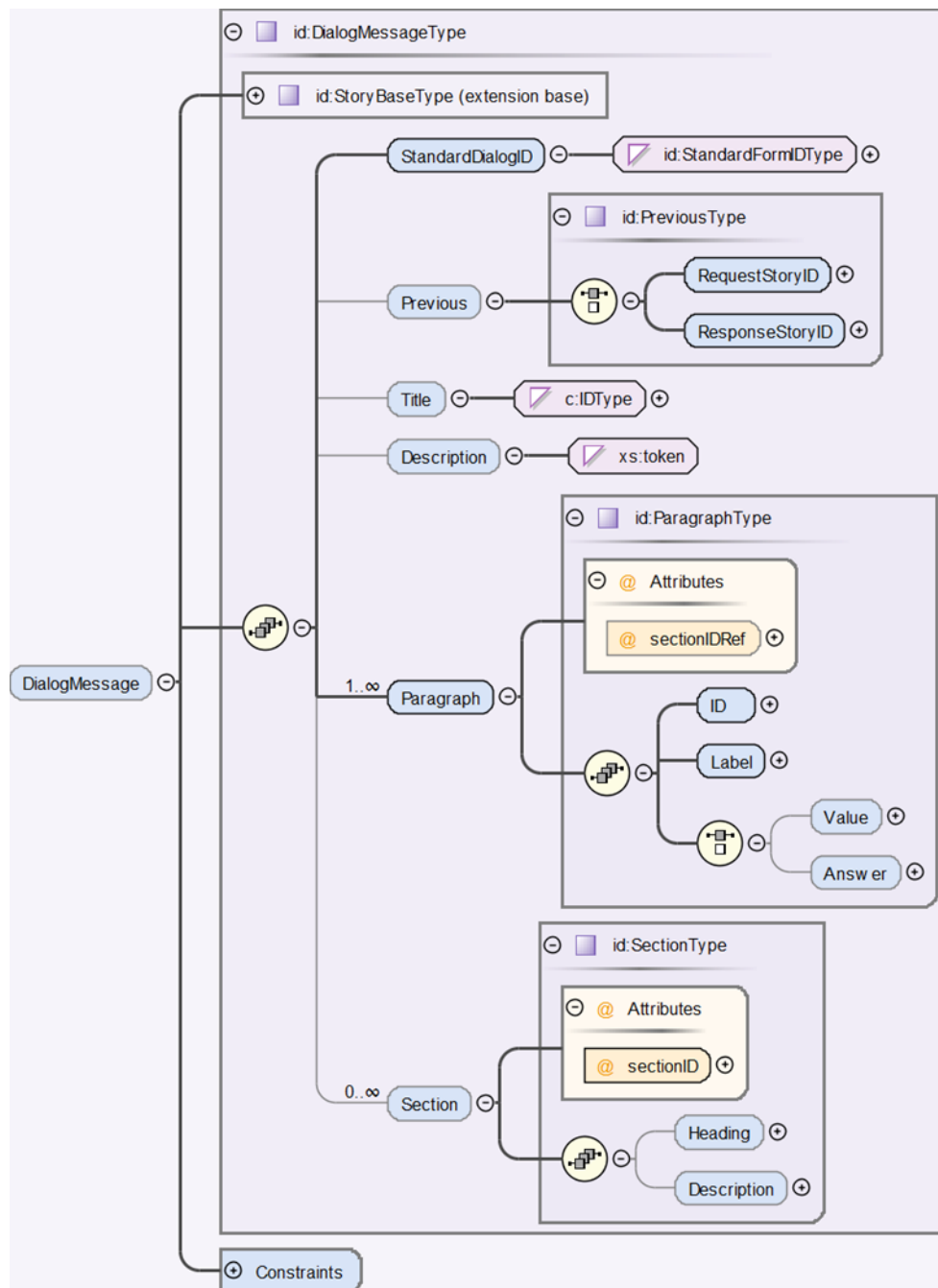


Abbildung 2: DialogMessage XML-Schema Struktur

Die GUI-Darstellung der einzelnen Elemente entspricht in etwa denjenigen eines Formulars z.B. in HTML oder einer Programmiersprache. Die Darstellung kann als einfacher Text oder aber durch Verwendung von GUI-Elementen wie Checkboxes, Dropdowns oder Kalenderkomponenten sein.

Das Element «Paragraph/Label» wird sowohl als Label, als auch als alleinstehender Text verwendet.

Zu beachten: Werte unter Paragraph/Answer können einen vorgegebenen Default-Wert haben, welcher dargestellt werden muss.

Element / Attribut	Editierbar	Darstellung	Bemerkung
Creation	Nein	Kann	Datum und Zeit der Erstellung der DialogMessage
StoryID	Nein	Kann	StoryID der DialogMessage: ID muss für den Geschäftsfall eindeutig sein (pro IncidentCaseID im KLE, pro DeclarationID im ELM). Ausserhalb eines Geschäftsfalls muss sie ebenfalls eindeutig sein.
StandardDialogID	Nein	Kann	Kennzeichnung als standardisierte DialogMessage oder «notStandard»
Previous	Nein	Kann	RequestStoryID oder ResponseStoryID der vorangegangenen DialogMessage, je nach dem ob die DialogMessage in einem Request oder einer Response enthalten war.
Title	Nein	Muss	Optionaler Titel der DialogMessage. Muss dargestellt werden, wenn vorhanden.
Description	Nein	Muss	Optionale Beschreibung der DialogMessage. Muss dargestellt werden, wenn vorhanden, z.B. als Tooltip
Section/sectionID	Nein	Kann	
Section/Heading	Nein	Muss	
Section/Description	Nein	Muss	
Paragraph	Nein	Muss	Absatz mit Inhalt
Paragraph/SectionIDRef	Nein	Kann	Attribut welches die Zugehörigkeit eines Absatzes zu einer Section anzeigt
Paragraph/ID	Nein	Kann	
Paragraph/Label	Nein	Muss	Text in einem Absatz, Label für Value oder Answer
Paragraph/Value/String	Nein	Muss	
Paragraph/Value/Integer	Nein	Muss	
Paragraph/Value/Double	Nein	Muss	
Paragraph/Value/Boolean	Nein	Muss	Bei einer Gruppe von Booleans ist keine ausschliessende Auswahl möglich.
Paragraph/Value/Date	Nein	Muss	
Paragraph/Value/DateTime	Nein	Muss	
Paragraph/Value/YesNoUnknown	Nein	Muss	
<i>Answer/<Type>/Default: darf vom Actor gesetzt, vom Reactor in der Antwort aber nicht verändert werden</i>			
Paragraph/Answer/String/Default	Nein	Muss	
Paragraph/Answer/String/Value	Muss	Muss	
Paragraph/Answer/Integer/Default	Nein	Muss	
Paragraph/Answer/Integer/Value	Muss	Muss	
Paragraph/Answer/Double/Default	Nein	Muss	
Paragraph/Answer/Double/Value	Muss	Muss	
Paragraph/Answer/Boolean/Default	Nein	Muss	Bei einer Gruppe von Booleans ist keine ausschliessende Auswahl möglich.
Paragraph/Answer/Boolean/Value	Muss	Muss	Bei einer Gruppe von Booleans ist keine ausschliessende Auswahl möglich.

Element / Attribut	Editierbar	Darstellung	Bemerkung
Paragraph/Answer/Date/Default	Nein	Muss	z.B. Kalender mit Textfeld
Paragraph/Answer/Date/Value	Muss	Muss	z.B. Kalender mit Textfeld
Paragraph/Answer/DateTime/Default	Nein	Muss	z.B. Kalender mit Textfeld
Paragraph/Answer/DateTime/Value	Muss	Muss	z.B. Kalender mit Textfeld
Paragraph/Answer/YesNoUnknown/Default	Nein	Muss	z.B. DropDown
Paragraph/Answer/YesNoUnknown/Value	Muss	Muss	z.B. DropDown
Paragraph/Answer/Problem	(Muss)	(Muss)	Darf bisher nicht verwendet werden. Zukünftige Verwendung erst, wenn in einer standard DialogMessage in neuerer Version dieses Dokuments spezifiziert.

Tabelle 4 Darstellung verschiedener Datentypen der DialogMessage

4.3 Abfolge bei der Darstellung von DialogMessages

Das Aktivitätsdiagramm beschreibt das generische Vorgehen beim Empfang einer Response oder eines Requests mit enthaltenen DialogMessages. Die Antwort auf eine DialogMessage erfolgt, sofern notwendig, zu einem späteren Zeitpunkt in einem neuen Request-Response-Zyklus.

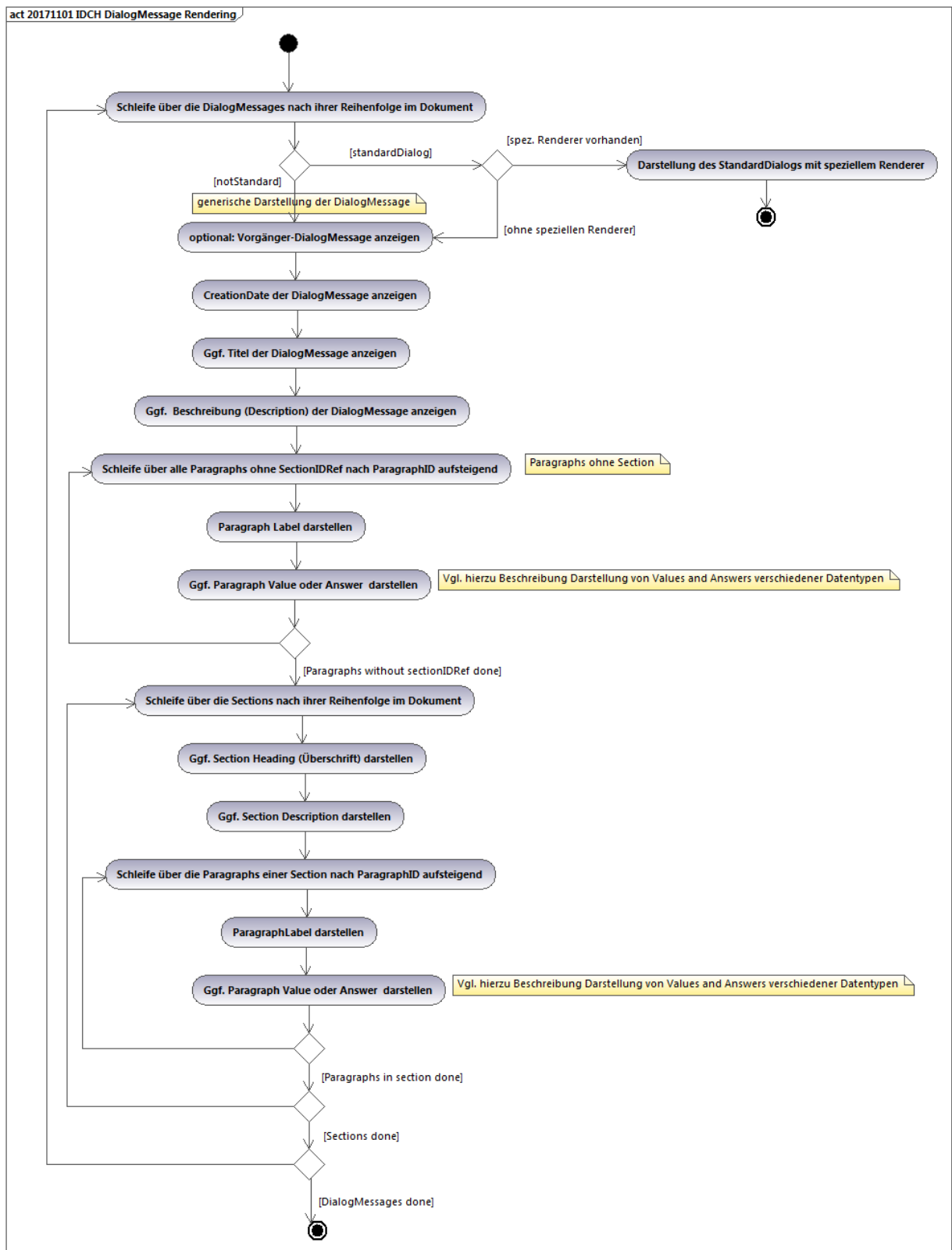


Abbildung 3 Vorgehen bei der Darstellung der DialogMessage (Aktivitäts Diagramm)

4.4 Antwort als Reactor mittels einer DialogMessage

Bei der Antwort eines Reactors auf eine DialogMessage, wird zwischen folgenden Situationen unterschieden.

1. Es wird eine DialogMessage-Story mit «Paragraph/Answer»-Elementen empfangen. Die DialogMessage **muss** dargestellt werden und es **muss** eine Antwort mit der gleichen Struktur gegeben werden. Die Antwort wird wahrscheinlich in einem späteren Request-Response Zyklus geschickt.
2. Es wird eine DialogMessage-Story ohne «Paragraph/Answer»-Elemente empfangen. Die DialogMessage **muss** dargestellt werden. Auf diese DialogMessage muss nicht geantwortet werden. Es muss keine Datenstruktur kopiert werden.
3. Es wird eine DialogMessage empfangen, welche sich auf eine beliebige, vorangegangene Story (z.B. AccidentDescription, Agreement, DialogMessage) bezieht. Der Bezug wird durch das Element «Previous» hergestellt, s. Verkettung. Es muss vom Actor keine Datenstruktur kopiert werden.

4.4.1 Antwort auf eine DialogMessage mit enthaltenen Paragraph/Answer-Elementen

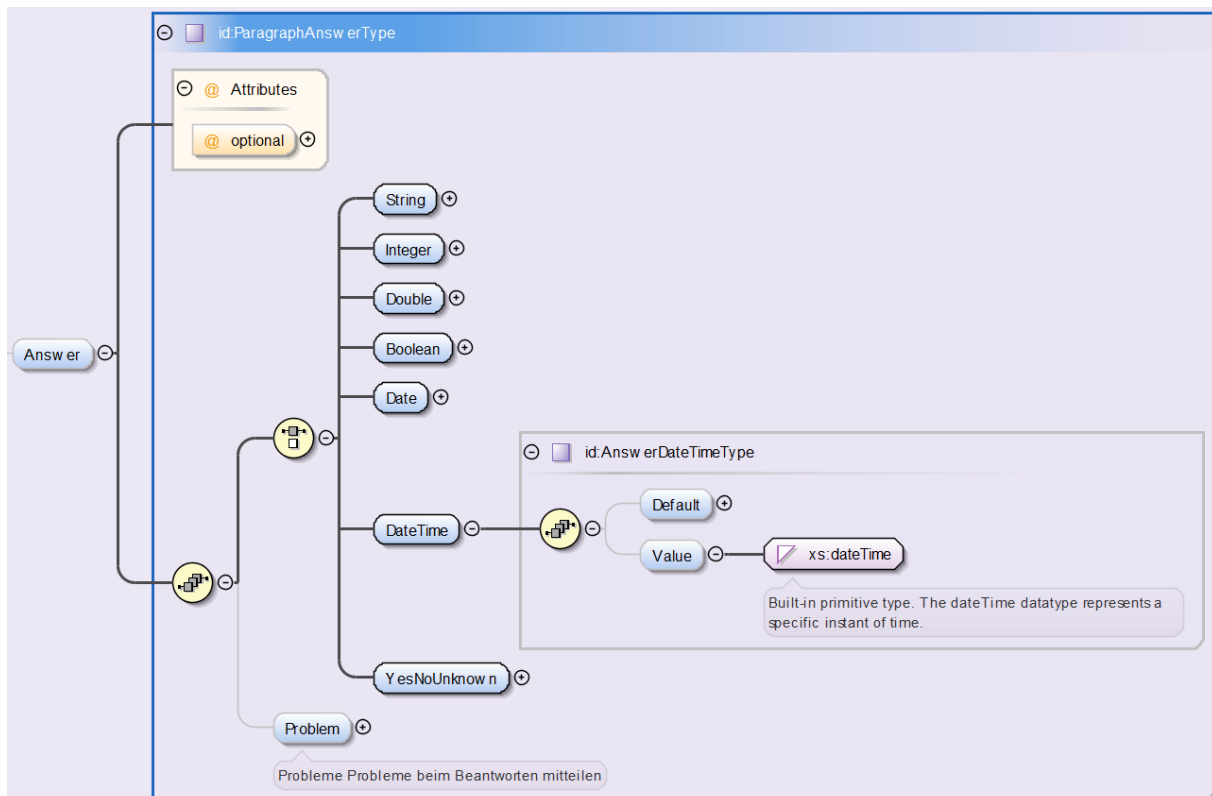


Abbildung 4 Paragraph/Answer/DateTime-Element, andere Answer/<Type>-Elemente entsprechend

- Die Antwort auf eine DialogMessage mit enthaltenen «Paragraph/Answer»-Elementen ist wiederum eine DialogMessage mit «Paragraph/Answer»-Elementen. Die Antwort wird wahrscheinlich in einem späteren Request-Response Zyklus erfolgen.
- In der Antwort-DialogMessage **muss** das Element «Previous», s. Kapitel 4.4.2 «Verkettung von DialogMessages» verwendet werden, so dass sowohl für den Actor, wie auch für den Reactor nachvollziehbar ist, auf welche DialogMessage-Story sich die Antwort bezieht.
Im Transmitter wird immer das Element «Previous/ResponseStoryID» verwendet, da die anfragende DialogMessage vom Endrezeiver immer in einer SynchronicIncidentResponse enthalten war.
Im Endrezeiver wird immer «Previous/RequestStoryID» verwendet, da die anfragende DialogMessage vom Transmitter immer in einem SynchronicIncidentRequest enthalten war.
- Bis auf «Creation», «StoryID» und «Previous» wird die gesamte DialogMessage mit enthaltenen «Paragraph/Answer»-Elementen gespiegelt.
- Die durch den Dialogteilnehmer geänderten Werte, ansonsten die Default-Werte, **müssen** bei der Antwort immer in die Paragraph/Answer/<Type>/Value Elemente eingetragen werden, nicht in die «Paragraph/Answer/<Type>/Default»-Elemente. Dabei wird <Type> durch den jeweils übergebenen Werte-Typ ersetzt, z.B. «Paragraph/Answer/Boolean/Value».

- «Answer»-Elemente, welche nur optional ausgefüllt werden müssen, sind durch das Attribut «optional» gekennzeichnet. Jedes andere «Paragraph/Answer/<Type>/Value»-Element **muss** bei der Antwort gesetzt sein. Das bedeutet, dass alle mit «Paragraph/Answer»-Elementen angefragten Werte ohne das Attribut «optional» Pflichtfelder sind, auch wenn diese nach XML-Schema leer zurückgesandt werden könnten.

4.4.2 Verkettung von DialogMessages

Die verschiedenen DialogMessages **können** miteinander verkettet werden. Sind in einer anfragenden DialogMessage «Answer»-Elemente enthalten, **muss** die Antwort-DialogMessage mit der anfragenden DialogMessage durch das Element «Previous» verkettet werden. Damit lässt sich in der Visualisierung eine ganze Kette von DialogMessages darstellen. In dieser Sicht werden einzelne DialogMessages für den Benutzer verständlicher, da diese in einem Ablaufkontext stehen und dann betrachtet werden können.

Die Verkettung wird mit dem Element «DialogMessage/Previous» gesteuert.

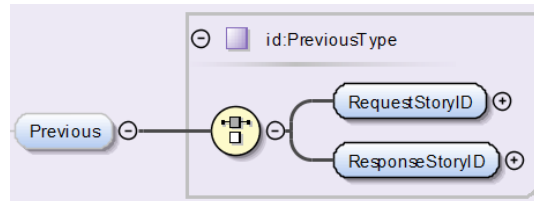


Abbildung 5 Struktur des DialogMessage/Previous-Elementes

1. Ein Actor initiiert im ersten Schritt einen Dialog und **kann** optional im Element «Previous» den Link zu einer bestehenden StoryID setzen. Dabei wird unterschieden zwischen einer beliebigen bestehenden Response- oder RequestStoryID.
2. Ein Reactor **muss** bei einer Antwort auf eine Actor DialogMessage immer ein «Previous» - Element setzen und zwar die StoryID der zuvor erhaltenen Actor DialogMessage.

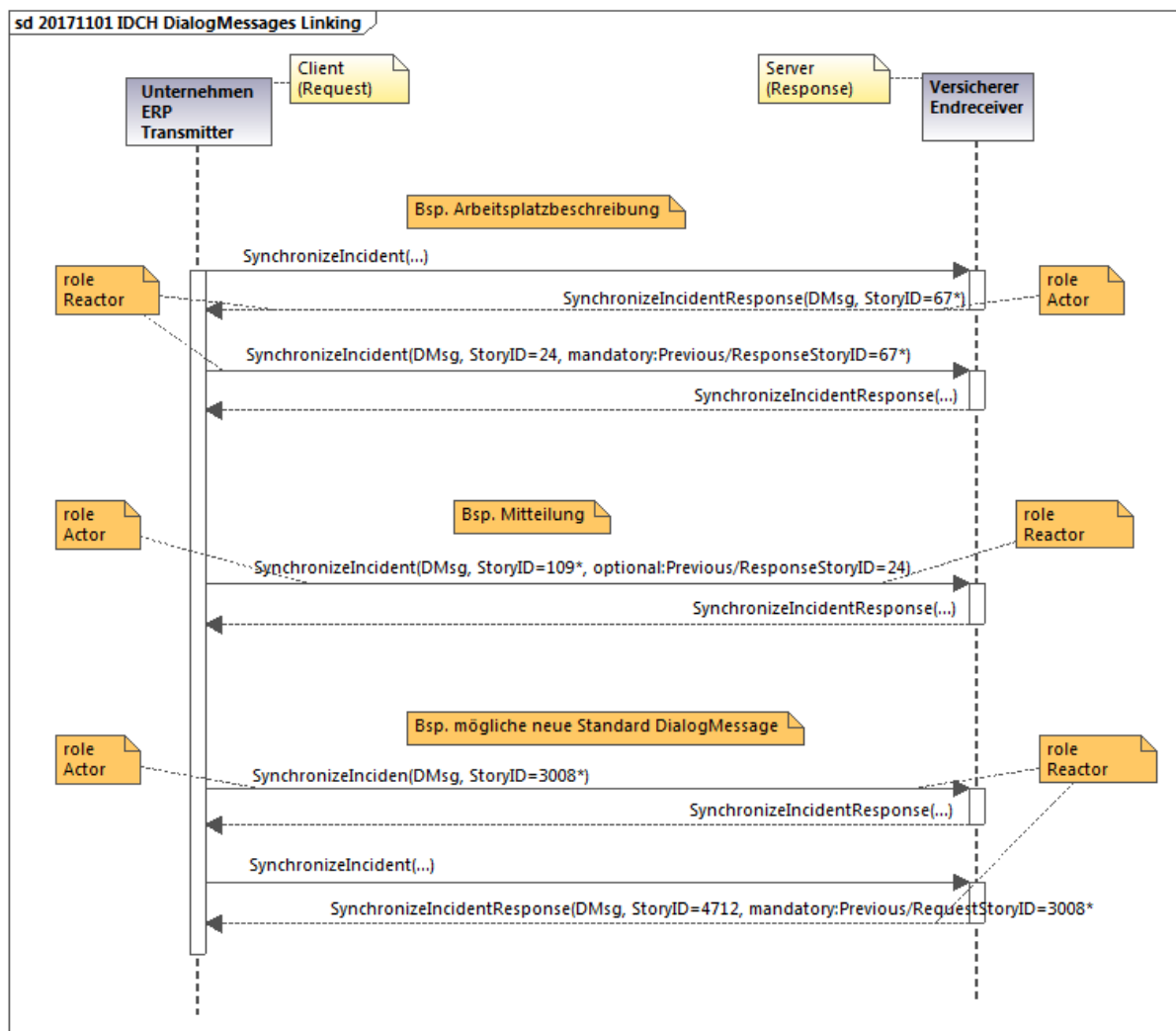


Abbildung 6 Verkettung von DialogMessages

5. Standard DialogMessages in KLE

Die folgenden DialogMessages sind standardisiert und **müssen** im Rahmen von KLE nach Tabelle 3 Anforderungen zur Umsetzung von DialogMessages erzeugt und dargestellt werden können. Die definierten Strukturen **müssen** eingehalten werden.

Eine DialogMessage **muss** die StandardDialogID ihres Typs im Instanzdokument setzen, s. nachfolgende Tabelle.

DialogMessage	StandardDialogID
SimpleMessage	0000.0001.0001-001
TaskWithDeadline	0000.0001.0002-001
TaskWithConfirmation	0000.0001.0003-001
ContactPerson	0000.0001.0004-001
WorkplaceDescription	0000.0001.0005-001
Freie DialogMessage	notStandard

Tabelle 5 StandardDialogIDs der DialogMessages

Die letzten drei Ziffern sind die Minor-Version der in diesem Dokument spezifizierten StandardDialogMessages. Sollte eine spezifizierte DialogMessage Änderungen erfahren, wird ihre Minor-Version erhöht.

Zu allen spezifizierten standard DialogMessages gibt es mitgeltende Beispiele in Form von Instanzdokumenten (REFBSPDIA-IDCH, 2018).

Damit Darstellung und Auswertung der standard DialogMessages sicher geparsed werden können, sind einige IDs fest vorgegeben. Folgende IDs **müssen** den Werten in den standard DialogMessage Referenzdokumenten (REFBSPDIA-IDCH, 2018) entsprechen:

- Alle sectionIDs **müssen** wie in den standard DialogMessage Referenzdokumenten gesetzt werden, inklusive der Gross- und Kleinschreibung und dem Hash-Zeichen, z.B. <Section sectionID="#ContactPerson">
- Alle Paragraph IDs von Paragraphs, welche eine sectionIDRef haben, **müssen** die gleiche ID wie in den standard DialogMessage Referenzdokumenten haben, z.B. hat der Name der Kontaktperson immer die ID=400
 <Paragraph sectionIDRef="#ContactPerson">
 <ID>400</ID>
 <Label>Name</Label>

- Bei den standard DialogMessages TaskWithDeadline, TaskWithConfirmation und ContactPerson gibt es einen Termin, bis zu welchem eine Antwort spätestens erwartetet wird. Der Paragraph mit diesem Termin **muss** immer die ID = 300 haben.
- Alle Paragraph/ID-Elemente **müssen** innerhalb des Elements «DialogMessage» eindeutig sein.

5.1 SimpleMessage «Mitteilung»

Das zugehörige Referenz-Instanzdokument ist «1_SimpleMessageActor.xml» aus (REFBSPDIA-IDCH, 2018)

Die Mitteilung ist eine einfache Nachricht eines Actors an einen Reactor und erfordert keine Antwort.

Das Element «DialogMessage/Title» **kann** vorhanden sein und enthält den «Betreff» oder das «Subject» der Aufgabe, analog einer E-Mail.

Eine Mitteilung **darf keine** Description- und keine Section-Elemente enthalten.

Die restlichen Paragraph-Elemente enthalten nur «ID» und «Label»-Elemente und **dürfen keine** anderen Elemente enthalten.

Es **kann** ein Element Previous mit der StoryID einer vorangegangenen Story enthalten sein.

Ausser der Betreff-Zeile («Title»-Element) **müssen** sich alle Texte in den «Paragraph/Label» Elementen befinden.

Alle Paragraph/ID-Elemente **müssen** innerhalb des Elements «DialogMessage» eindeutig sein.

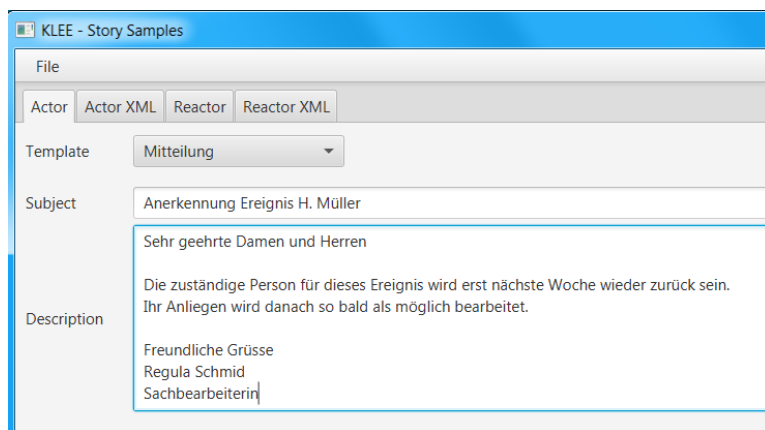


Abbildung 7: Beispiel einer Actor-GUI-Darstellung der DialogMessage «Mitteilung»

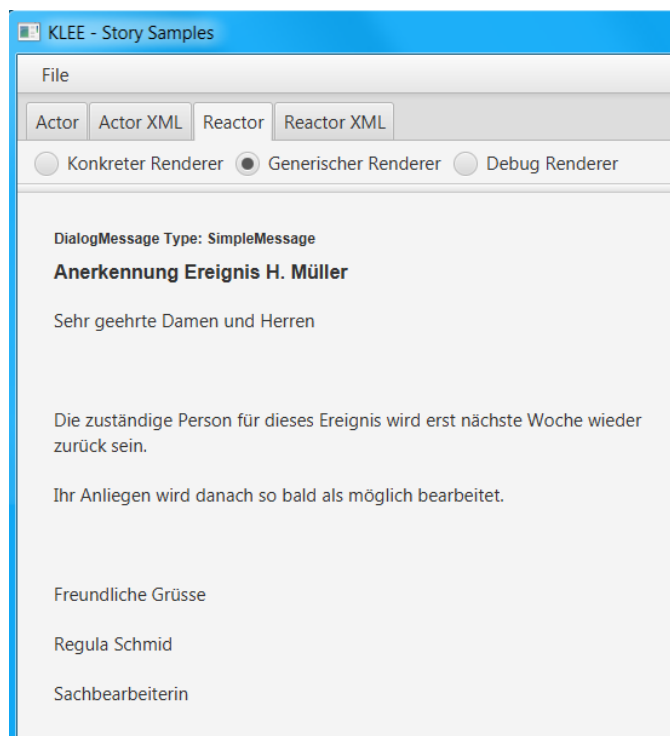


Abbildung 8: Beispiel einer Reactor-GUI-Darstellung der Standard DialogMessage «Mitteilung»

```
<DialogMessage>
  <Creation>2018-02-02T12:28:49.146+01:00</Creation>
  <StoryID>11151757092</StoryID>
  <StandardDialogID>0000.0001.0001-001</StandardDialogID>
  <Title>Anerkennung Ereignis H. Müller</Title>
  <Paragraph>
    <ID>100</ID>
    <Label>Sehr geehrte Damen und Herren</Label>
  </Paragraph>
  <Paragraph>
    <ID>101</ID>
    <Label> </Label>
  </Paragraph>
  <Paragraph>
    <ID>102</ID>
    <Label>Die zuständige Person für dieses Ereignis wird erst nächste Woche wieder zurück sein.</Label>
  </Paragraph>
  <Paragraph>
    <ID>103</ID>
    <Label>Ihr Anliegen wird danach so bald als möglich bearbeitet.</Label>
  </Paragraph>
  <Paragraph>
    <ID>104</ID>
    <Label> </Label>
  </Paragraph>
  <Paragraph>
    <ID>105</ID>
    <Label>Freundliche Grüsse</Label>
  </Paragraph>
  <Paragraph>
    <ID>106</ID>
    <Label>Regula Schmid</Label>
  </Paragraph>
  <Paragraph>
    <ID>107</ID>
    <Label>Sachbearbeiterin</Label>
  </Paragraph>
</DialogMessage>
```

Abbildung 9: Beispiel XML-Instanz «Mitteilung» analog (REFBSPDIA-IDCH, 2018) 1_SimpleMessageActor.xml

5.2 TaskWithDeadline «Aufgabe»

Das zugehörige Referenz-Instanzdokument ist «2_TaskWithDeadlineActor.xml» aus (REFBSPDIA-IDCH, 2018)

Die Aufgabe ist eine Anfrage eines Actors an einen Reactor mit dem Termin, bis zu welchem die Aufgabe erledigt sein soll. Sie erfordert keine Antwort.

Das Element «DialogMessage/Title» **kann** vorhanden sein und enthält den «Betreff» oder das «Subject» der Aufgabe, analog einer E-Mail.

Eine Aufgabe **darf keine** Description- und keine Section-Elemente enthalten.

Eine Aufgabe **muss** einen einzigen Paragraphen mit der *ID 300* und dem Termin in einem Paragraph/Answer/Date/Value Element enthalten.

Die restlichen Paragraph-Elemente enthalten nur «ID» und «Label»-Elemente und **dürfen keine** anderen Elemente enthalten.

Es **kann** ein Element Previous mit der StoryID einer vorangegangenen Story enthalten sein.

Ausser der Betreff-Zeile («Title»-Element) **müssen** sich alle Texte in den «Paragraph/Label» Elementen befinden.

Die Paragraph/ID-Elemente **müssen** innerhalb des Elements «DialogMessage» eindeutig sein.

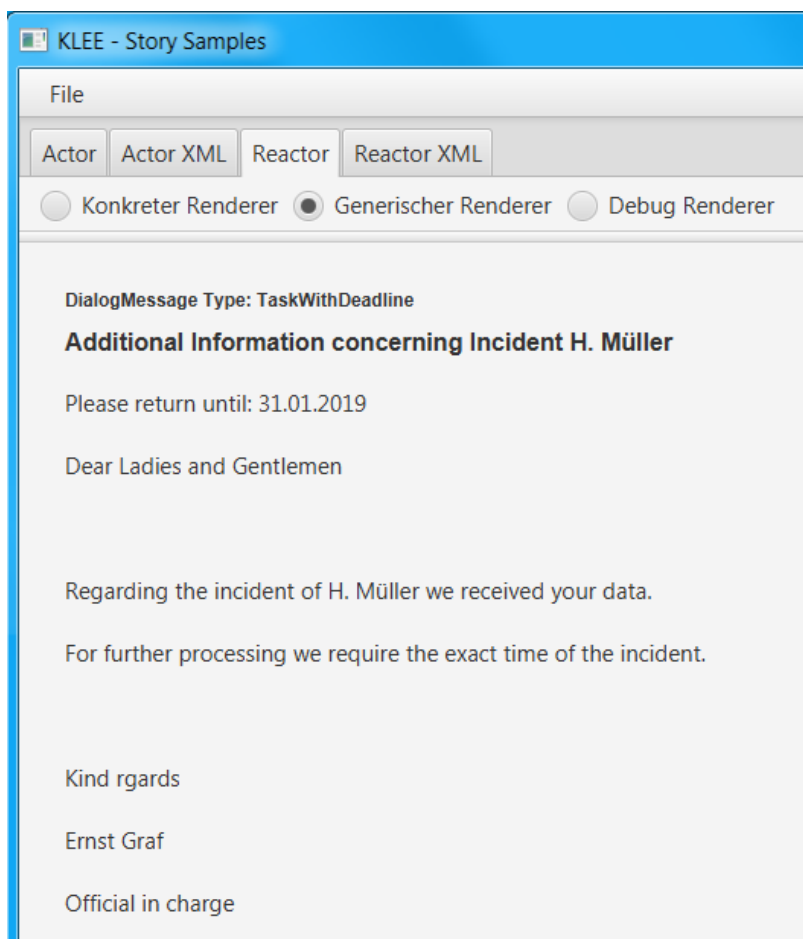


Abbildung 10: Beispiel einer Reactor-GUI-Darstellung der DialogMessage «Aufgabe»

5.1 TaskWithConfirmation «Anfrage mit Bestätigung»

Die zugehörigen Referenz-Instanzdokumente sind «3_TaskWithConfirmationActor.xml» und «3_TaskWithConfirmationReactor.xml» aus (REFBSPDIA-IDCH, 2018).

Die Anfrage mit Bestätigung ist von der Grundstruktur her die Standard DialogMessage Aufgabe mit einem zusätzlichen Boolean Value und es **muss** darauf geantwortet werden. Die Bestätigung der Anfrage entspricht in etwa einer Lesebestätigung in einem E-Mail Programm.

Das Element «DialogMessage/Title» **kann** vorhanden sein und enthält den «Betreff» oder das «Subject» der Anfrage mit Bestätigung, analog einer E-Mail.

Eine Anfrage mit Bestätigung **darf keine** Description- und keine Section-Elemente enthalten.

Eine Anfrage mit Bestätigung **muss** ein «Paragraph»-Element mit der ID 300 und dem Termin in einem «Paragraph/Answer/Date/Value» Element enthalten.

Eine Anfrage mit Bestätigung **muss** ein «Paragraph»-Element mit der ID 310 und dem Wert der Bestätigungsanforderung in einem «Paragraph/Answer/Boolean/Default» Element mit dem Wert true enthalten. In der Antwort **muss** der Paragraph das Element «Paragraph/Answer/Boolean/Value» enthalten.

In der Antwort **muss** das Element Previous mit der StoryID der anfragenden DialogMessage gesetzt sein.

Ausser im «Title»-Element **müssen** sich alle Texte in den «Paragraph/Label» Elementen befinden.

Die Paragraph/ID-Elemente **müssen** innerhalb des Elements «DialogMessage» eindeutig sein.

Die Antwort auf die Anfrage mit Bestätigung ist ebenfalls eine DialogMessage. Sie ist eine Kopie der anfragenden DialogMessage, vergleiche Kapitel 4.4 «Antwort als Reactor mittels einer DialogMessage» und enthält die Bestätigung im «Answer»-Element:

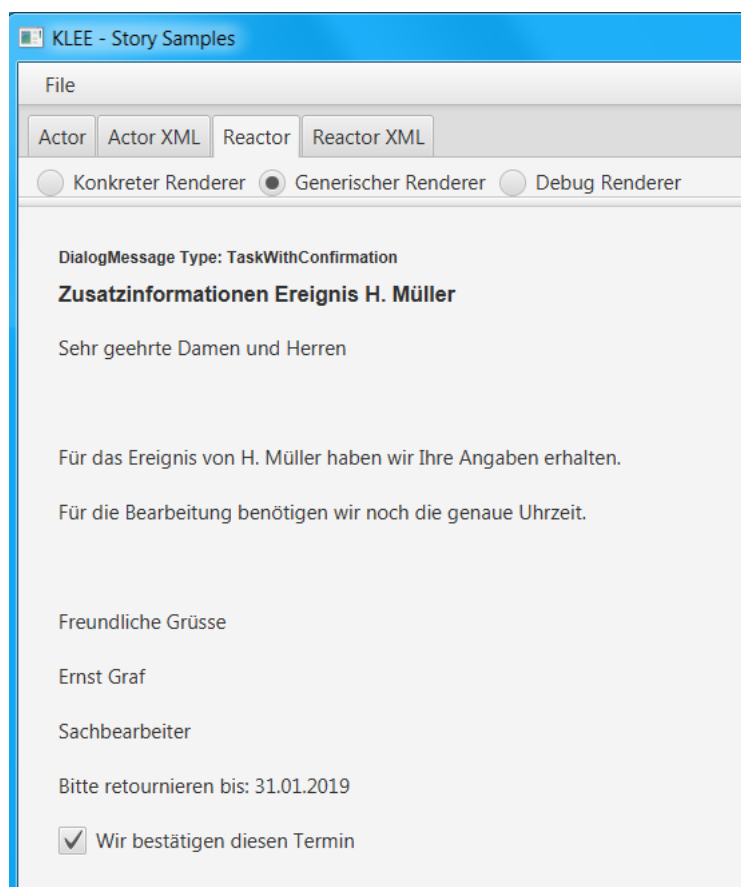


Abbildung 11: Beispiel einer Reactor-GUI-Darstellung der DialogMessage «Anfrage mit Bestätigung»

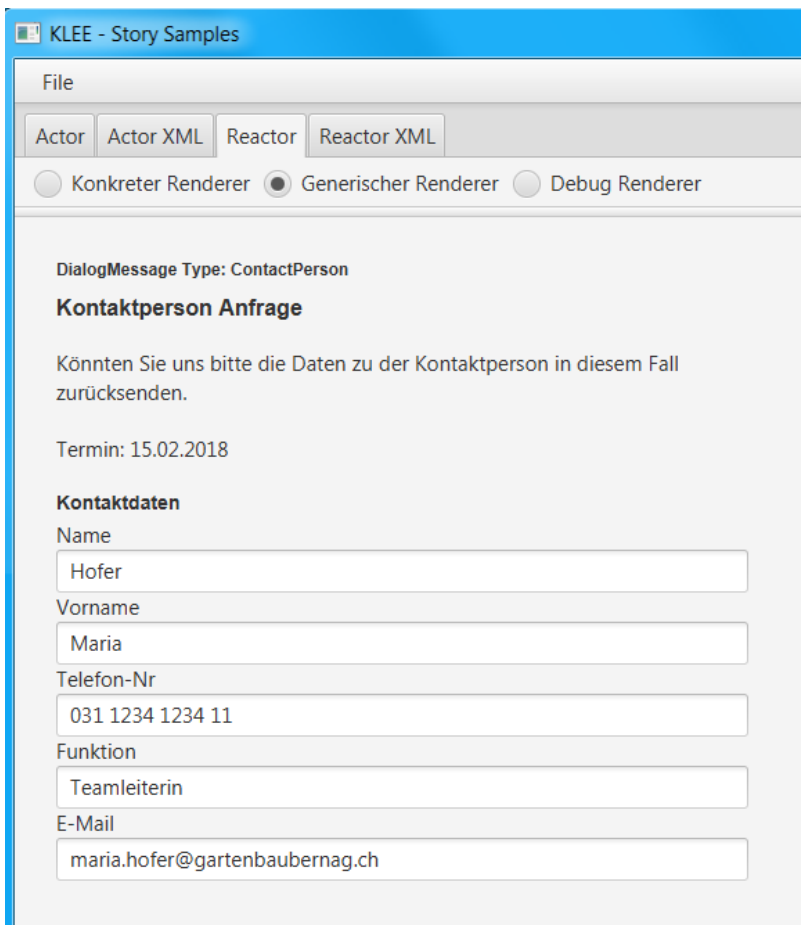
5.1 ContactPerson «Kontaktperson Anfrage»

Die zugehörigen Referenz-Instanzdokumente sind «4_ContactPersonActor.xml» und «4_ContactPersonReactor.xml» aus (REFBSPDIA-IDCH, 2018).

Paragraph-IDs, sectionIDs, sectionIDRefs und die Reihenfolge von sections und Paragraphs **müssen** denjenigen aus den zugehörigen Referenz-Instanzdokumenten entsprechen. Das Element Previous mit der ResponseStoryID der anfragenden DialogMessage, **muss** in der Reactor-DialogMessage enthalten sein.

In der «Kontaktperson Anfrage» können folgende Elemente optional sein:

- optional ist «DialogMessage/Title» welches, wenn vorhanden, ein «Betreff» oder «Subject» analog einer E-Mail enthält.
- optionale Paragraph-Elemente, welche einzig je ein Label-Element für Erläuterungen enthalten.



The screenshot shows a web application titled 'KLEE - Story Samples'. It has a 'File' menu and four tabs: 'Actor', 'Actor XML', 'Reactor', and 'Reactor XML'. Below the tabs are three radio buttons: 'Konkreter Renderer', 'Generischer Renderer' (which is selected), and 'Debug Renderer'. The main content area displays 'DialogMessage Type: ContactPerson' and the title 'Kontaktperson Anfrage'. The text reads: 'Könnten Sie uns bitte die Daten zu der Kontaktperson in diesem Fall zurücksenden.' followed by 'Termin: 15.02.2018'. Under the heading 'Kontaktdaten', there are several input fields: 'Name' (Hofer), 'Vorname' (Maria), 'Telefon-Nr' (031 1234 1234 11), 'Funktion' (Teamleiterin), and 'E-Mail' (maria.hofer@gartenbaubernag.ch).

Abbildung 12: Beispiel einer Reactor-GUI-Darstellung der DialogMessage «Kontaktperson Anfrage»

Die Antwort auf die Kontaktperson Anfrage ist ebenfalls eine DialogMessage. Sie ist eine Kopie der anfragenden DialogMessage, wie unter Kapitel 4.4 «Antwort als Reactor mittels einer DialogMessage» beschrieben und enthält die benötigten Angaben in den einzelnen Answer/Value-Elementen:

Das Previous-Element mit der ResponseStoryID oder RequeststoryID der anfragenden Story **muss** enthalten sein.

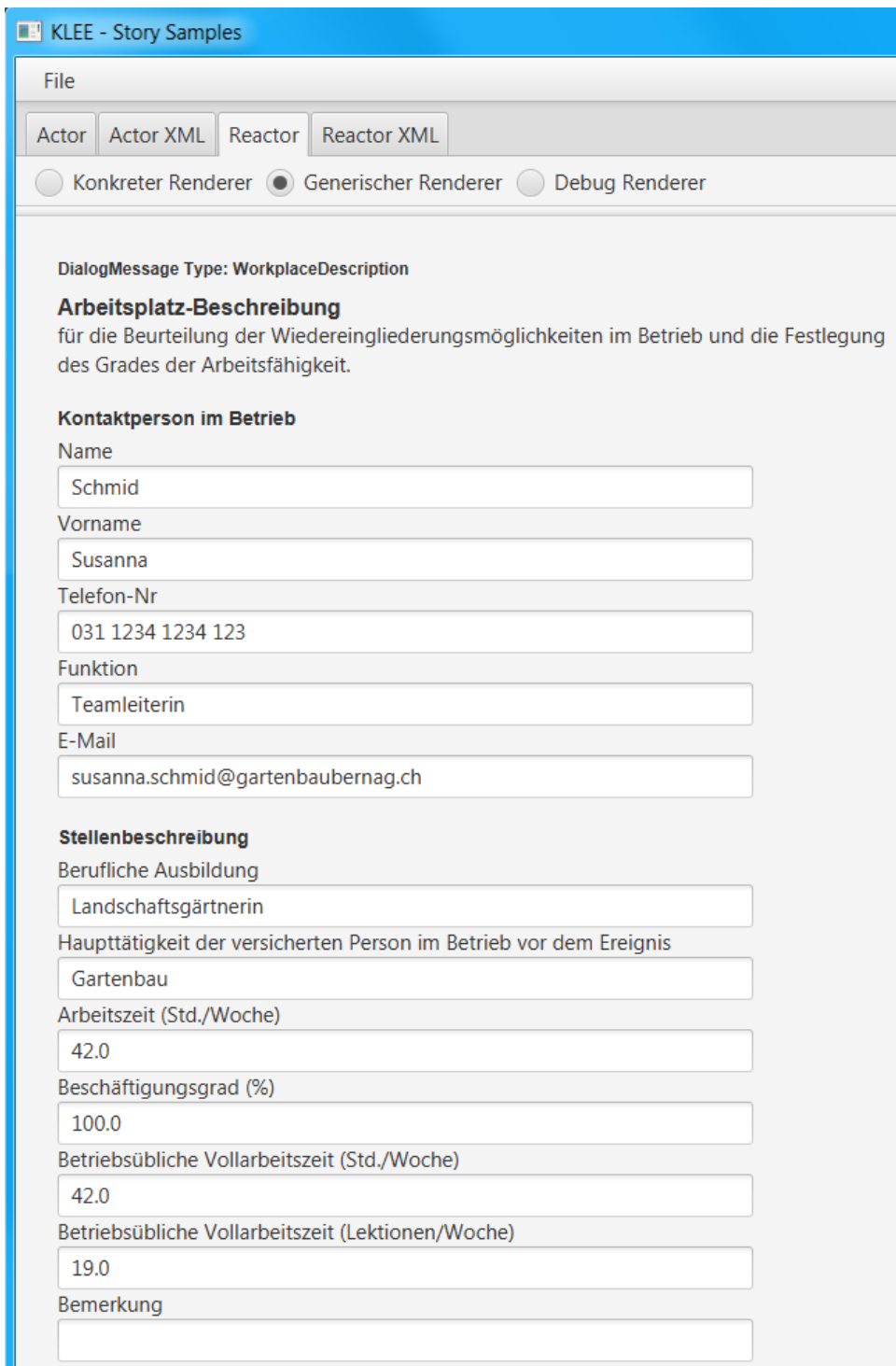
5.2 WorkplaceDescription «Arbeitsplatz-Beschreibung»

Die zugehörigen Referenz-Instanzdokumente sind «5_WorkplaceDescriptionActor.xml» und «5_WorkplaceDescriptionReactor.xml» aus (REFBSPDIA-IDCH, 2018).

Paragraph-IDs, sectionIDs, sectionIDRefs und die Reihenfolge von sections und Paragraphs **müssen** denjenigen aus den zugehörigen Referenz-Instanzdokumenten entsprechen. Das Element Previous mit der ResponseStoryID der anfragenden DialogMessage, **muss** in der Reactor-DialogMessage enthalten sein.

Optionale «Answer»-Elemente sind durch das Attribut «optional» gekennzeichnet.

Die Paragraph/ID-Elemente **müssen** innerhalb des Elements «DialogMessage» eindeutig sein.



The screenshot shows a web application titled "KLEE - Story Samples". It has a "File" menu and four tabs: "Actor", "Actor XML", "Reactor", and "Reactor XML". Below the tabs are three radio buttons: "Konkreter Renderer", "Generischer Renderer" (which is selected), and "Debug Renderer". The main content area displays a "DialogMessage Type: WorkplaceDescription". Under this, there are several sections with form fields:

- Arbeitsplatz-Beschreibung**
für die Beurteilung der Wiedereingliederungsmöglichkeiten im Betrieb und die Festlegung des Grades der Arbeitsfähigkeit.
- Kontaktperson im Betrieb**
 - Name: Schmid
 - Vorname: Susanna
 - Telefon-Nr: 031 1234 1234 123
 - Funktion: Teamleiterin
 - E-Mail: susanna.schmid@gartenbaubernag.ch
- Stellenbeschreibung**
 - Berufliche Ausbildung: Landschaftsgärtnerin
 - Haupttätigkeit der versicherten Person im Betrieb vor dem Ereignis: Gartenbau
 - Arbeitszeit (Std./Woche): 42.0
 - Beschäftigungsgrad (%): 100.0
 - Betriebsübliche Vollarbeitszeit (Std./Woche): 42.0
 - Betriebsübliche Vollarbeitszeit (Lektionen/Woche): 19.0
 - Bemerkung: (empty field)

Abbildung 14: Beispiel einer Reactor-GUI-Darstellung der DialogMessage «Arbeitsplatz-Beschreibung» Teil 1

Spezielle Anforderungen / Rahmenbedingungen

<input type="checkbox"/>	Sitzarbeitsplatz
<input checked="" type="checkbox"/>	Steharbeitsplatz
<input type="checkbox"/>	Sitz-Steharbeitsplatz
<input type="checkbox"/>	Bildschirmarbeitsplatz
<input type="checkbox"/>	Visuelle Kontrolltätigkeit
<input checked="" type="checkbox"/>	Kundenkontakt
<input checked="" type="checkbox"/>	Teamarbeit
<input checked="" type="checkbox"/>	Lärm
<input checked="" type="checkbox"/>	Wärme, Kälte, Dämpfe
<input type="checkbox"/>	Geruch
<input checked="" type="checkbox"/>	Staubbelastung
<input checked="" type="checkbox"/>	Häufige Haltung in verdrehter Stellung
<input checked="" type="checkbox"/>	Präziser Handeinsatz mit erhöhtem Kraftaufwand
<input checked="" type="checkbox"/>	Erhöhtes Unfallrisiko für beeinträchtigte Mitarbeiter
<input type="checkbox"/>	Heben < 5kg
<input checked="" type="checkbox"/>	Heben > 5kg
<input checked="" type="checkbox"/>	Heben > 10kg
<input checked="" type="checkbox"/>	Heben > 15kg (teilweise)

Andere spezielle Anforderung / Rahmenbedingung

Maschinen und Geräte bedienen

Arbeitszeit

<input checked="" type="checkbox"/>	Tagesbetrieb fix
<input type="checkbox"/>	Tagesbetrieb gleitend
<input type="checkbox"/>	Nachtschichten notwendig
<input type="checkbox"/>	Ausschliesslich Nachtschichten

Andere spezielle Arbeitszeiten

Schonarbeitsplatz

<input type="checkbox"/>	vorhanden
<input checked="" type="checkbox"/>	nur temporär möglich
<input type="checkbox"/>	nicht vorhanden

Beschreibung des möglichen Schonarbeitsplatzes

Verkauf von Gartenartikeln

Kontaktaufnahme

☐ durch den Arzt erwünscht

mit Versicherung erfolgt am

14.11.2019

Weitere Informationen

Wir bitten Sie, uns wenn immer möglich auch einige Fotos des angestammten Arbeitsplatzes der verunfallten Person zur Verfügung zu stellen. Diese benötigt unser Kreisarzt oder der behandelnde Arzt für das bessere Verständnis zur konkreten Tätigkeit Ihres Mitarbeiters. Bitte übermitteln Sie uns die Bilder via swissdec Leistungsstandard (IncidentStories/Attachment) bis am: 31.05.2019

Bemerkungen

Besten Dank.

Abbildung 15: Beispiel einer Reactor-GUI-Darstellung der DialogMessage «Arbeitsplatz-Beschreibung» Teil 2

6. Anhang

6.1 Referenzen

REFBSPDIA-IDCH, s. (15. 02 2018). StandardDialogMessageInstances. (swissdec, Hrsg.) Bern, Schweiz.

RL-IDCH, s. (09. November 2017). Richtlinien für den Leistungsstandard-CH. (swissdec, Hrsg.) Bern, Schweiz.
Von <https://www.swissdec.ch/de/releases-und-updates/richtlinien-kee> abgerufen

XSD-IDCH, s. (02. 11 2017). IncidentDeclaration.xsd. Bern, Switzerland.