Proposal

Subject: Choreography and composition
Date: 12th July 2004

Author: Gary Brown (Enigmatec Corp)

ISSUE:

Current approach to composing a sub-choreography is based on 'performing it', and sharing variables across the boundary. Subsequent activities in the parent choreography may then vary depending on the values returned from 'performing' the choreography. This is very functional, and therefore does not easily support the interaction based nature of a choreography.

Specific problems are:

1) Approach is brittle, as based on shared variables defining next state. This results in a loose binding of the interactions of the sub-choreography with the parent choreography. If the sub choreography changes, it is hard to detect impact on parent choreography. In general this approach will be difficult to model check.

2) Performed choreographies are atomic - they must complete before being able to continue. This restricts the types of choreographies that can be used.

3) It is not possible to interact with a choreography. Some choreographies may represent a more complex business transaction that requires further interactions.

4) Difficult to handle multiple outcomes - e.g. if the sub-choreography results in a set of parallel paths, then it would be difficult to enable the parent choreography to continue each of the concurrent business flows that resulted.

PROPOSAL:

Sub-choreographies are defined because they perform a self contained business task, which may result in multiple outcomes. For example, in Peter Furniss's recent example, he wanted to re-use a 'credit check' choreography, but perform different tasks depending upon the outcome (i.e. good or bad credit).

The only way to solve this problem at the moment is to record the outcome of the sub-choreography in a state variable, and then inspect that variable in the outer choreography. This approach is very brittle, as changes to the composed choreography would be undetectable in the outer choreography - i.e. it would not be easy to perform model verification.

The proposed approach is to compose the parent and sub-choreography based on the fact that the parent choreography would be playing one of the participants/roles in the sub-choreography. For example, in the credit check example, the parent choreography would be the role requesting the credit check. Therefore, to indicate how and where the parent choreography interactions with the sub-choreography, it would include the interactions associated with that role (from the sub-choreography) in the parent choreography.

Therefore two choreographies could be 'bound' by a common set of interactions related to a shared "relationship".

In the simple case, where a sub-choreography only has two participants (with a single relationship between them), this may mean that all the interactions in the sub-choreography are included in the parent choreography - and therefore you may wonder what the benefit would be of composing the sub-choreography. In this situation, although all the interactions may be duplicated, the benefit is that the sub-choreography then acts as a definition for the purpose of conformance checking (i.e. model checking). If the sub-choreography was to change, we would then be able to detect the difference.

However, where the sub-choreography includes many participants, then the parent choreography would only have to include the interactions associated with one of the "relationships", to identify how the parent choreography interacts with the child choreography.

The benefit of this approach is that it enables the parent choreography to supply 'callback' channels to the sub-choreography, to enable the sub-choreography to initiate interactions that will result in other activities being triggered in the parent choreography.

There would be no need to expose state variables across this interface, and changes in the observable behaviour of the sub-choreography would be detectable when verifying the parent choreography.

BENEFITS:

1) Model checking would be easier based on composition of the choreographies - enabling incompatible changes in sub-choreographies to be detected.

2) Parent choreography can 'interact' with the sub-choreography, at the beginning but also at any stage during the sub-choreography. Consistent approach based on interactions as opposed to a 'functional' approach based on sharing state variables.

3) Possible for parent choreography to result in multiple outcomes initiated by interactions from the sub-choreography.

4) Equivalent to process algebra composition - matching compatible interactions on shared channels (relationships).

COMPOSITION PROPOSAL - EXAMPLE

Payment Processing

A choreography could be defined for performing payment processing on behalf of a supplier. Many supplies use the services of intermediaries (such as WorldPay) to offer a range of payment options to its customers, which also involves doing any necessary authorisation. When the payment has been made, the supplier is notified (so they can supply the goods) and the customer is notified.

```
<interaction from="supplier" to="payproc" operation="processPayment" >
        <passed channel="customer" />
</interaction>
```

```
<interaction from="payproc" to="creditcardcomp" operation="authorize" />

<choice>
	<sequence>
		<interaction from="creditcardcomp" to="payproc" operation="authorized" />
		<interaction from="payproc" to="supplier" operation="paymentSuccessful" />

		<!-- Now send a message to the customer to indicate that their payment has
			been approved -->
		<interaction from="payproc" to="customer" operation="paymentAuthorized" />
	</sequence>

	<sequence>
		<interaction from="creditcardcomp" to="payproc" operation="unauthorized" />
		<interaction from="payproc" to="supplier" operation="paymentFailed" />
	</sequence>
</choice>
```

Goods Distribution

A supplier will use a distributor company to deliver purchased goods to a customer. This choreography is only between two participants, so it will be duplicated

```
<interaction from="supplier" to="distributor" operation="shipGoods" />

<choice>
	<sequence>
		<interaction from="distributor" to="supplier" operation="delivered" />
	</sequence>
	<sequence>
		<interaction from="distributor" to="supplier" operation="undelivered" />
	</sequence>
</choice>
```

Composed Choreography - the Supplier

```
<interaction from="customer" to="supplier" operation="purchaseShoppingList" />

<interaction from="supplier" to="payproc" operation="processPayment" >
	<passed channel="customer" />
</interaction>

<choice>
	<sequence>
		<interaction from="payproc" to="supplier" operation="paymentSuccessful" />

		<!-- Now send a message to the customer to indicate that their payment has
			been approved -->
		<interaction from="payproc" to="customer" operation="paymentAuthorized" />

		<interaction from="supplier" to="distributor" operation="shipGoods" />

		<choice>
			<sequence>
				<interaction from="distributor" to="supplier" operation="delivered" />

				<interaction from="supplier" to="customer" operation="goodDispatched" />

			</sequence>
			<sequence>
				<interaction from="distributor" to="supplier" operation="undelivered" />

				<interaction from="supplier" to="customer" operation="deliveryFailed" />
			</sequence>
		</choice>

	</sequence>

	<sequence>
		<interaction from="payproc" to="supplier" operation="paymentFailed" />
	</sequence>
</choice>
```

In this example, the "Payment Processing" choreography is integrated around the role of "payproc" and "customer" (because the customer is supplied in an interaction with payproc, and therefore any interactions performed by the "Payment Processing" choreography on customer would also become visible in the composed choreography).

As the "Goods Distribution" choreography only has two participants, its interactions are duplicated in the composed choreography - although this does not necessarily have to be the case. If the sub-choreography offered two alternate paths, then the composed choreography may only use one of the paths, in which case it would not include any interactions related to the other path. Although the complete choreography is included (in this case), the benefit is that if any changes are made to the "Goods Distribution" choreography, we would be able to verify whether they had an observable impact on this composed choreography.

The benefits of this approach are:

This approach enables complex choreographies to be combined in an interleaved manner based on shared behaviour, as opposed to being atomic units.

Follow-on activities can be performed in the appropriate flow, as opposed to being based on the evaluation of state variable values, making the composed choreography simple to understand.

The aim of composition should not be to save on the number of lines of CDL. It is to ensure that individual (self contained) choreographies can be combined to construct higher level choreographies, where the boundary between the choreographies can easily be verified for correctness.

TO DO:

Before this can be formally proposed, we would need to determine how a sub-choreography is referenced from a parent choreography. Should it be in the definition of the role or relationship? Should it be in some construct that is associated with the relevant channel name associated with the relationship that binds the choreographies?