Adaptive Choreographies

Ivan Lanese
Computer Science Department
University of Bologna/INRIA
Italy

Joint work with Mila Dalla Preda, Jacopo Mauro and Maurizio Gabbrielli
Choreographies

- Allow to describe the behavior of a distributed communicating system at the very abstract level
- Composed by interactions of the form \( a \rightarrow^o b \)
- Using different operators
  - Sequential composition ;
  - Parallel composition ||
  - Nondeterministic choice +
- There are approaches extending choreographies with additional information (data, recursion, ...)
- Very similar to global types in multiparty session types
Choreography projection

- Allow to automatically derive from a choreography the description of the behavior of each participant
- Nearer to the implementation
- Preserves the semantics: when interacting, the participants behave as specified by the choreography
Participants description

- Locations corresponding to participants, containing their code
- Basic operations: input $o@a$ and output $\bar{o}@a$
- Composed using
  - sequential composition ;
  - parallel composition $|$
  - nondeterministic choice $+$
Examples

- The projection of $a \rightarrow b; b \rightarrow c$ is
  $$[\overline{o}@b]_a || [o@a; \overline{o'}@c]_b || [o'@b]_c$$

- The projection of $a \rightarrow b; c \rightarrow d$ is
  $$[\overline{o}@b]_a || [o@a]_b || [\overline{o'}@d]_c || [o'@c]_d$$
  - Not well-behaved
  - Syntactic connectedness conditions exist to avoid this problem
  - Choreographies can be transformed so to satisfy the conditions
Adaptation

- Systems should live for long periods of time in ever-changing environments
- Users can change their minds
- The system should adapt to satisfy new requirements
Grey box adaptation

- Adaptation happens at runtime
- Adaptation details not known when the system has been designed or even started
- The system should provide an interface to interact with an adaptation middleware
- At runtime the adaptation middleware will send new code to the system
Adaptation and choreographies

- Lots of works on adaptation exist
- Not many formal approaches
- Very little guarantees on the properties of the system after adaptation takes place
  - Which parts of the behavior are changed and which are preserved?
  - Is it still safe?
- Using choreographies we can guarantee safe adaptation
Our approach, architecturally

- A system is obtained as projection of a choreography
- The system may run on its own
- The system may interact with an adaptation middleware
  - Composed by possibly distributed adaptation servers
  - Requires a dedicated interface from the application
- A single adaptation may involve many participants
  - Need for some coordination protocol
Adaptation rules

- The adaptation servers contain rules which can be applied to update the running system
  - Conditions checking whether adaptation is applicable/useful can be specified
  - Each rule may involve multiple participants
  - New rules can be added at any moment
- Application of a rule involves code mobility from the adaptation server to the system
- Rules can be added at any time + rules contain code → the system may not contain code to cope with every possible adaptation when it is started
Our approach, semantically

- Extend choreographies with adaptation scopes
  - Part of the choreography to which a rule may be applied
  - But the choreography does not specify which rule

- The system is executed together with a set of applicable rules (abstracting adaptation servers) and an environment

- Adaptation rules include
  - The new code for the scope
  - Information on when and where the rule can be applied

- Using projection we can derive the new code for each participant

- A coordination protocol is required to apply the updates
Our approach, graphically
Our approach, graphically
Our approach, graphically
Advantages of the approach

- The effect of adaptation at the choreography level is clearer
  - Abstract, global view
  - Good properties by construction
- Adaptation is applied to the running system
  - Projection allows to bridge the gap
Adaptation scopes

- An adaptation scope has the form \( \{C, \Delta\}_{r, l, k} \)
- \( C \) is the choreography to be executed if no adaptation is performed
- \( \Delta \) is a set of non functional properties guaranteed by the current choreography
- \( r \) is the role leading the adaptation process
- \( l \) is a label of the scope
- \( k \) is a key ensuring the scope is uniquely identified
Adaptation rules

- An adaptation rule has the form $l, c \rightarrow C$
- $l$ is the label of scopes the rule can be applied to
- $c$ is an applicability condition
  - May involve non-functional properties of the current scope and environment conditions
- $C$ is the new choreography to be executed if adaptation succeeds
Projecting a scope

- Essentially homomorphic
  - On the leader we also keep the set of involved participants
  - On other roles, $\Delta$ is not needed

- Semantics of the leader
  - Check whether there is a rule targeting the scope whose applicability condition holds
  - If so, download the code for all the participants and send it to them
  - If not, tells the other participants that no adaptation is needed

- Semantics of other participants
  - Wait for instruction from the leader
  - If adaptation is needed, execute the new code
  - If not, execute the current code
Results

- A choreography and its projection have the same traces
  - Under all possible adaptations
  - With environments and sets of applicable rules that may change at any moment during the computation
- The adapted system is deadlock free by construction
- Note that we are adapting in a coordinated way a distributed system
Current and future work

- Going towards an implementation
  - Language with data, if-then-else and while
  - More at the level of choreographies according to Montesi & Carbone terminology
  - Projecting to Jolie

- Current choreographies can be seen as types for the new ones

- Which is the impact of adaptation on refinement?

- Different design choices on adaptation to be explored
  - When is adaptation applied?
  - Which applicability conditions are allowed?
End of talk

Thanks!

Questions?