

Automated service composition aims at automatically generating software solutions based on services to provide more complex functionality. This service composition process has been tackled with many AI techniques, but existing approaches make unreasonably strong assumptions such as a predefined data flow, are limited to tiny problem sizes, ignore non-functional properties, or assume offline service repositories.

In the first part of the talk, we present our algorithm that automatically composes services without making such assumptions. The algorithm incorporates a backward search approach that starts from an empty composition and prepends service calls to already discovered candidates until a solution is found. Available services are determined during the search process.

In the second part of the talk, we give an overview about why adaptivity becomes increasingly important when aiming for automated composition in dynamic and freely accessible environments such as service markets. We systematically derive dependencies among crucial processes such as service composition and service execution in a holistic view and briefly discuss the effects of changes in the environment according to the derived dependencies.