

Mixed Initiative Features of Semi-Automated Service Composition Systems: A Survey

Jan Schaffner



Agenda

- **Motivation**
- **Mixed Initiative Features**
- **Existing Approaches**
- **Evaluation**
- **Conclusion**

Motivation: Web Service Composition

- **Status quo: Compositions are created manually**
- **Experts are necessary for manual service composition**
 - **All conceivable alternative paths must be specified**
 - **All possible failures have to be considered**
 - **Maintenance is difficult and costly**
- **Benefits of automated composition:**
 - **Creation of individual plans for each case**
 - **Plans aligned with desired goal**

Motivation: Drawbacks of Automated Planning

- **Complete formal representations must be available**
 - **Domain knowledge and service descriptions**
 - **Initial state and goal state**

- **Automated planners plan with limited knowledge**

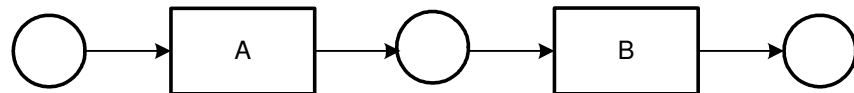
- **Transition from research to industry progresses slowly:**
 - **Juridical impediment**
 - **Human should be in the loop**

Motivation: Semi-Automated Composition

- **System should augment human planning skills rather than controlling the planning process**
- **Mixed initiative: Both human and computer contribute to the creation of the composed service**
- **Incomplete or incorrect semantic descriptions can be compensated by human planner**
- **Reasoning capabilities reduce problems akin to manual service composition**

Prerequisites: Modeling Service Capabilities

- Information space: **I**nputs and **O**utputs
- World state: **P**reconditions and **E**ffects
- Service composition can be conceived as set of states and transitions:
 - Transitions: Actual activities (“service calls”), characterized by IOPEs
 - States: Information space and world state at a given point in time



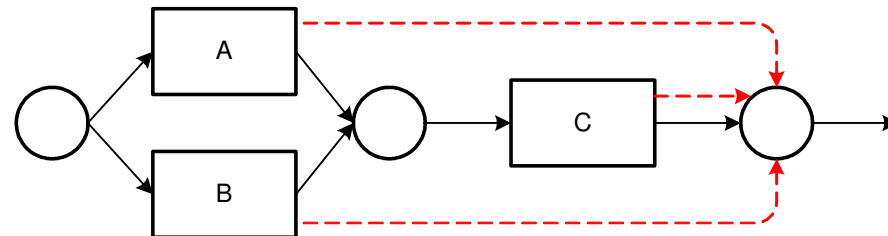
MI Feature I: Filter Inappropriate Services

➤ Problem:

- Number of available services can be very high

➤ Idea: Select services in the context of composition step

- State depends on outputs and effects of all precedent services
- Filter services that are not applicable in current state



➤ Extensions:

- Filter on nonfunctional properties after filtering on IOPEs
- Ordering according to “goodness of match”
- Ordering according to past user activity

MI Feature II: Suggest Partial Plans

➤ Problem:

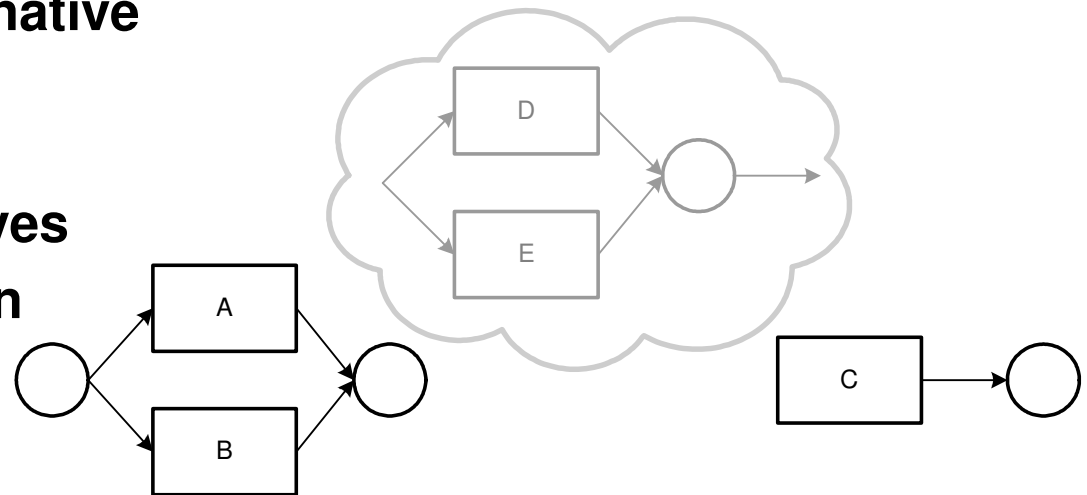
- User might have a clear idea only about specific parts of the composition

➤ Idea:

- Connecting two services can be seen as planning problem
- Automatically generate partial plans
- User selects an alternative

➤ Extension:

- Ordering of alternatives based on aggregation of a nonfunctional property (e.g., cost)



MI Feature III: Check Validity

➤ Problem:

- User is likely to introduce errors into the composition

➤ Idea:

- Verify that composition is “well-formed”
 - inputs and preconditions of all services are satisfied
 - all services produce outputs or effects relevant for downstream services or produce end results
 - no redundant services
- Interleave creation and validation of composition

➤ Extension:

- Suggest appropriate actions to resolve errors

Existing Approaches: Web Service Composer

- **Tool for creating executable Web service compositions**
- **Knowledge base format**
 - **OWL-S ontologies**
- **Imposed planning strategy**
 - **Backward chaining**
 - **output-input subsumption**
- **Incompatible services are filtered at each step**
- **Further filtering based on nonfunctional properties in OWL-S service profiles**

Existing Approaches: CAT

- **Interactive composition of “computational workflows”**
- **Knowledge base format**
 - **Component ontologies**
 - **Domain ontologies**
- **Imposed planning strategy**
 - **Components can be added at any time**
- **Outputs of components can be declared as end result**
- **ErrorScan algorithm evaluates well-formedness and determines suggestions**

Existing Approaches: PASSAT

- **Planning tool based on hierarchical task networks (HTN)**
- **Knowledge base format**
 - **HTN extension for ontologies**
 - **Supports *world state***
- **Imposed planning strategy**
 - **Implies top-down plan refinement approach**
 - **Plan sketch facility**
- **User can invoke automated planning mode to expand open tasks**
- **Planned:**
 - **Guidance in the process of automated task expansion**
 - **Advice-mechanism based on high-level user policies**

Existing Approaches: IRS-III

- **Tool for user-guided interactive Web service composition and execution**
- **Knowledge base format**
 - **WSMO ontologies**
 - **WSMO goals represent Web services**
- **Imposed planning strategy**
 - **Backward chaining**
 - **Output-input subsumption**
- **WSMO mediators can be used to integrate semantic specifications from different parties**
- **Provides if-then-else control operator**

Evaluation: Supported Mixed Initiative Use Cases

	Web Service Composer	CAT	PASSAT	IRS-III
Filter Inappropriate Services	Considers only inputs / outputs	---	---	Considers only inputs / outputs
Extensions	Filtering on NFPs, Ordering by match-goodness	---	---	---
Suggest Partial Plans	---	---	HTN template expansion	---
Extensions	---	---	High-level policies for composition	---
Check Validity	---	Evaluates “well-formedness”	Tracks open information requirements	---
Extensions	---	Suggests fixes (ErrorScan)	Prioritized Agenda	---

Evaluation: Additional Criteria

	Web Service Composer	CAT	PASSAT	IRS-III
Imposed planning strategy	Backward chaining	None	Top-down refinement	Backward chaining
Knowledge base format	OWL-S	Proprietary	Proprietary	WSMO
Reasoning	Output-input subsumption	Output-input subsumption	Complete IOPEs taken into account	Output-input subsumption
Control constructs	Not provided	Not provided	Not provided	If-then-else construct
Composition are executable	Tool acts as Web service client	No	No	Orchestration engine
Modeling environment	Graphical	Textual	Textual	Graphical
Output format	OWL-S process model	Non-standard	Non-standard	Non-standard

Conclusion

- ➔ **Usability of existing approaches is weak**
- ➔ **Use case “suggest partial plans” only realized in context of HTN template expansion**
- ➔ **Extensibility of existing approaches is limited:**
 - **Most approaches only reason on inputs and outputs**
 - **Not using standards for semantic descriptions prevents usage of available domain knowledge**