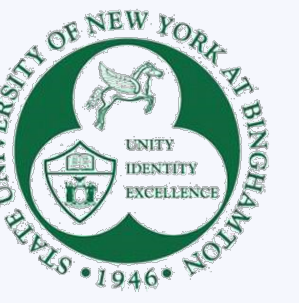


Task and Situation Structures for Service Agent Planning



Hao Yang¹, Tavan Eftekhari¹, Chad Esselink¹, Yan Ding², Shiqi Zhang²

¹Ford Motor Company, Dearborn, MI, USA, ²State University of New York at Binghamton, Binghamton, NY, USA

Problem

- Service Agents work in domains that classical AI planning methods find challenging:
 - ◆ States, conditions, and even goals are often not well defined
 - ◆ Various, unbounded, unpredictable environments
 - ◆ Multifarious situations
- Need solutions that enable scalable and adaptable planning systems for the Service Agents

Generic Task & Situation Structures

→ Generic *Task* Structure:

| Attribute: | Explanation |
|--------------|--|
| Task name: | string name of a <i>Task</i> class |
| Parent task: | null if no parent |
| Sub-tasks: | a list of sub-tasks, empty if leaf |
| Action: | the action of the <i>Task</i> |
| Specs: | detail specs for the action |
| Conditions: | preconditions for this <i>Task</i> |
| Effects: | effects after the <i>Task</i> is performed |
| Context: | a list of contexts of this <i>Task</i> |
| Goals: | goals to be verified |
| Est Time: | estimated execution time |

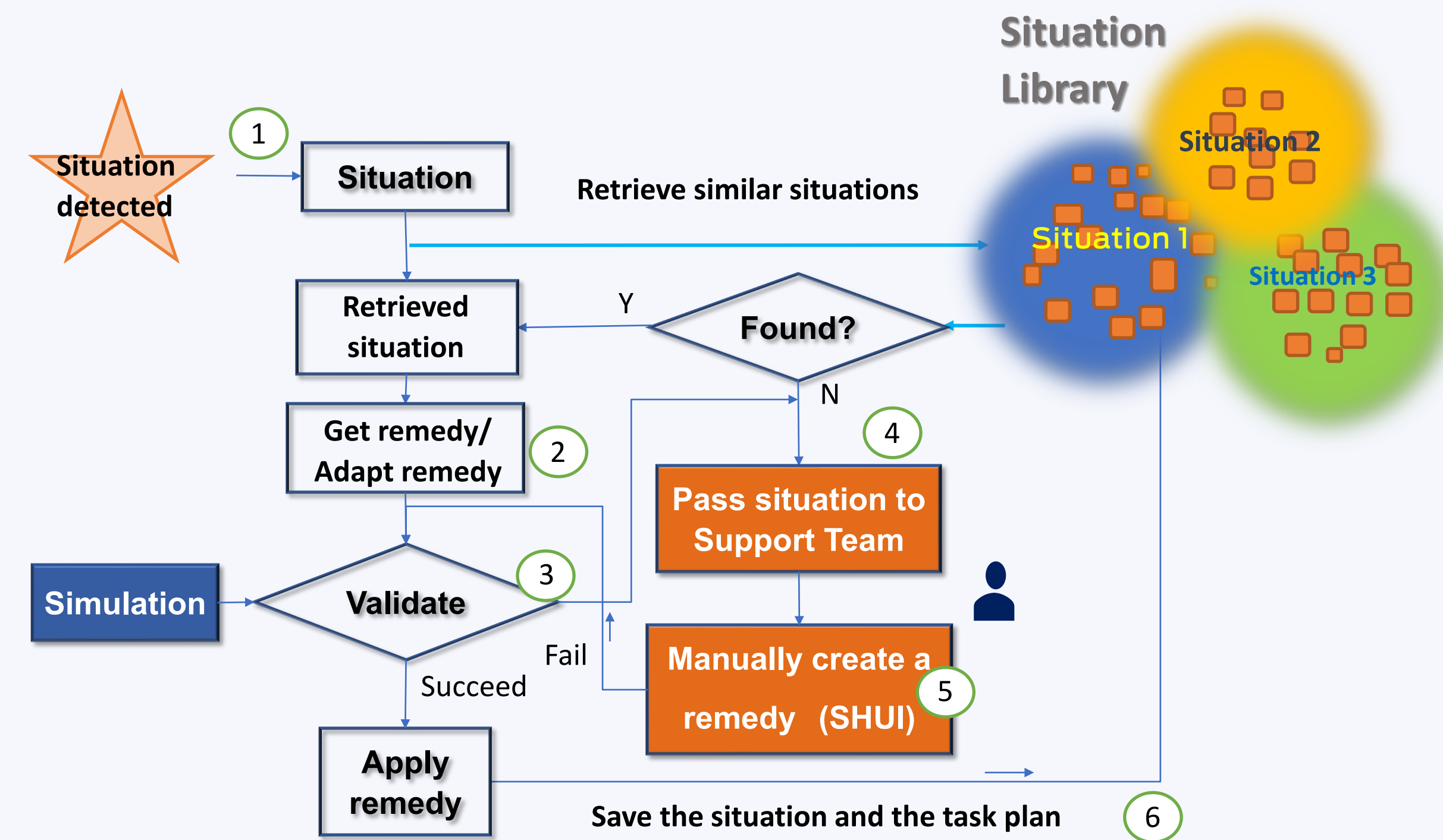
→ Generic *Situation* Structure:

| Attribute: | Explanation |
|------------|---|
| Name: | name of this <i>Situation</i> |
| Time: | time this <i>Situation</i> occurred |
| Task: | <i>Task</i> during which the <i>Situation</i> is logged |
| Context: | contexts while this <i>Situation</i> happened |
| Remedy: | a list of remedy actions to take |
| Logics: | how to set the Context and the Remedy |
| Goals: | new goals the repaired plan should satisfy |

Key Features of the Generic Structures

- It creates text-based, generic structures and syntax for *Tasks* and *Situations* of all variations.
- It embeds domain knowledge in executed cases and avoids the necessity of hard-coded or static domain rules.
- It offers a design that tolerates imperfections in the model (goals, conditions, states, actions, effects) and data.
- It uses context as additional attributes into guiding the search for solutions.

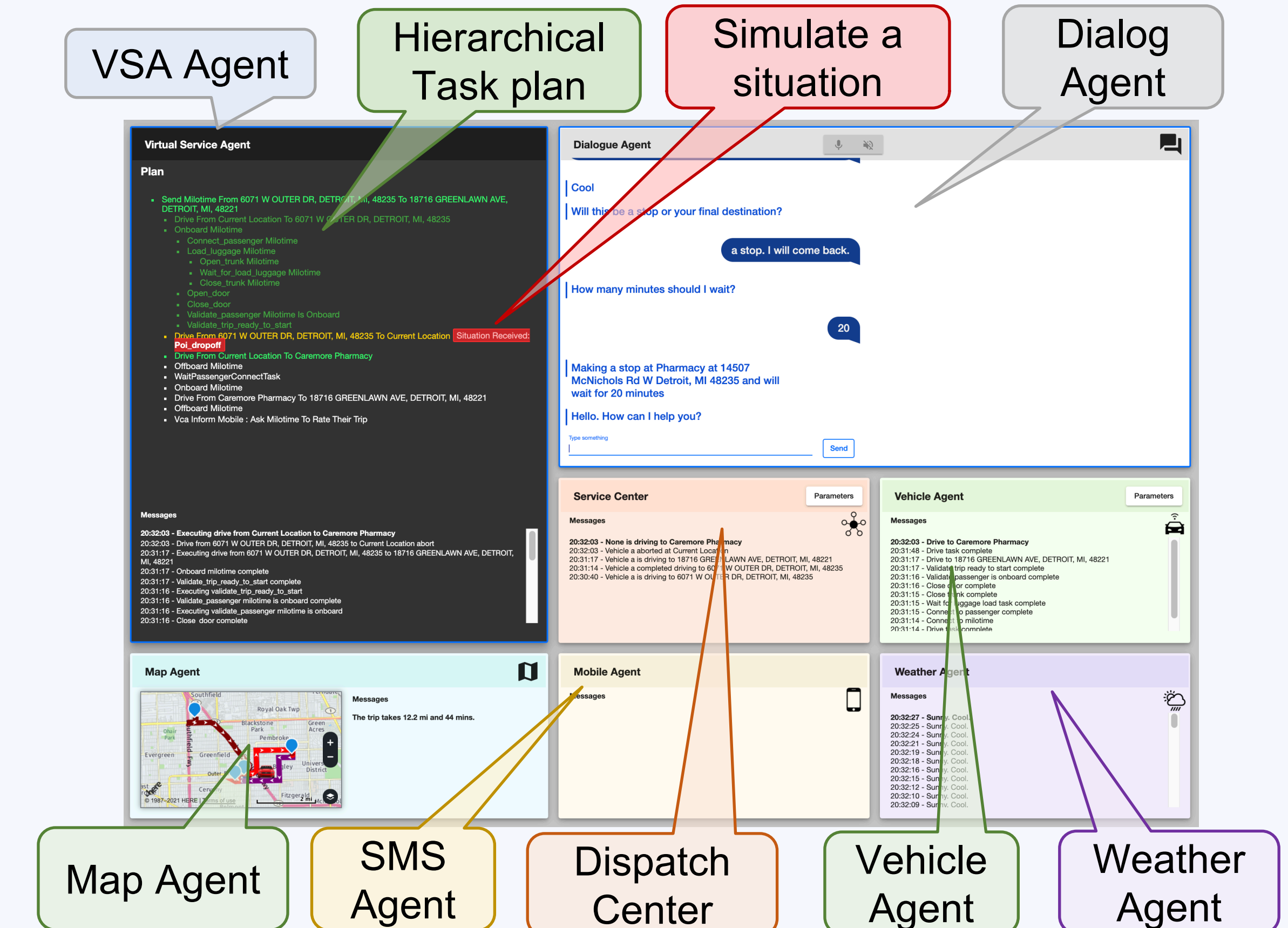
Method of Situation Handling (plan repair)



- ① *Situation* detected
- ② Retrieve a similar prior *Situation* and apply its Remedy to handle the situation
- ③ Validate if the modified plan passes validation
- ④ If failed, the executing task plan and *Situation* are sent to the remote support team
- ⑤ The remote support personal uses the tool (Situation Handling UI) to construct new Remedy for the *Situation*
- ⑥ The new *Situation* and its Remedy are saved for future reuse

Demonstration

- Virtual Service Agent (VSA): a demonstration platform for the ride hailing domain



Conclusions

- The generic *Task* and *Situation* structures allow us to introduce new Tasks and Situations into the system easily.
- The structures enable us to avoid hard-code domain rules and alleviate knowledge acquisition cost.

References

- Hammond, K. J. 2012. *Case-based planning: Viewing planning as a memory task*. Elsevier.
- Scholnick, E. K.; and Friedman, S. L. 1993. *Planning in context: Developmental and situational considerations*. International Journal of Behavioral Development
- Hao Yang, Tavan Eftekhari, Chad Esselink, Yan Ding, Shiqi Zhang, 2021, Task and Situation Structures for Case-based Planning, to appear: Proceedings of ICCBR