TU фреймворк в автоматической обработке входящих заявок

Александр Тощев

Казанский (Приволжский) Федеральный Университет, Институт Механики и Математике имени Чеботарева, [atoschev@kpfu.ru](mailto:atoschev@kpfu.ru)

Максим Таланов

Казанский (Приволжский) Федеральный Университет, Институт Высшей школы ИТИС, [max.talanov@kpfu.ru](mailto:max.talanov@kpfu.ru)

*Введение*

Разработанный фреймворк предоставляет возможность обработки входящих сообщений на естественном языке, область его применения достаточно широка. От интернет помощников до интеграции в системы автоматической обработки инцидентов.

Модель базируется на тезисе о возможности понимания входящего запроса, а не слепой обработки. Понимание базируется на мышление. В теоритическую основу фреймворка легла модель мышления Марвина Мински[1]. В данной модели мышление делится на 3 категории:

1. Критик
2. Селектор
3. Путь мышления

*Критик*

Критик является по сути своей вероятностным предикатом. В упрощенным виде можно назвать их триггерами. После любого внешнего воздействия активируется несколько критиков. Они проверяют с помощью своего предиката сработали они или нет. После активации критик возвращает селектор. Например, с точки зрения входящих заявок может быть использован критик «Прямые инструкции». Примеры критиков:

1. Критик Выученных реакций
2. Рефлексивный критик
3. Само-рефлексивный критик

*Селектор*

Селектор возвращает ресурсы необходимые для обработки запросов. В узком смысле Селектор может вернуть либо другой критик, либо другой селектор.

*Путь мышления*

Путь мышления — это путь при помощи которого будет решена проблема. В фреймворке TU есть несколько типов путей мышления:

* Общая логика
* Специализированный путь мышления

Примерами специализированных путей мышления могут служить следующие пути мышления:

1. Система знает, как решить проблему, используя уже накопленные знания
2. Адаптация существующего решения (если проблема подобна)
3. Реформуляция, с помощью которой вырабатываются новый путь мышления, после решения проблемы, используя путь 2
4. Корреляция, с помощью который ведется поиск уже существующих знаний по входящим.
5. Помощь, путь при помощи которого система обращается к человеку за помощью

Архитектура фреймворка предполагает возможность использования нескольких путей параллельно, что подразумевает быструю обработку запросов. Таким образом можно избежать зависание системы в одном пути решения и гарантировать поиск наиболее оптимального решения.

Пример 1. Если проблема сгенерирована автоматически, система должна обработать ее согласна инструкциям из книги А.

Пример 2. Если описание проблемы уже есть в Базе Знаний, то можно использовать путь Адаптации (2), чтобы решить данную проблему.

С точки зрения программной архитектуры путь мышления проводит работу над данными в текущем контексте.