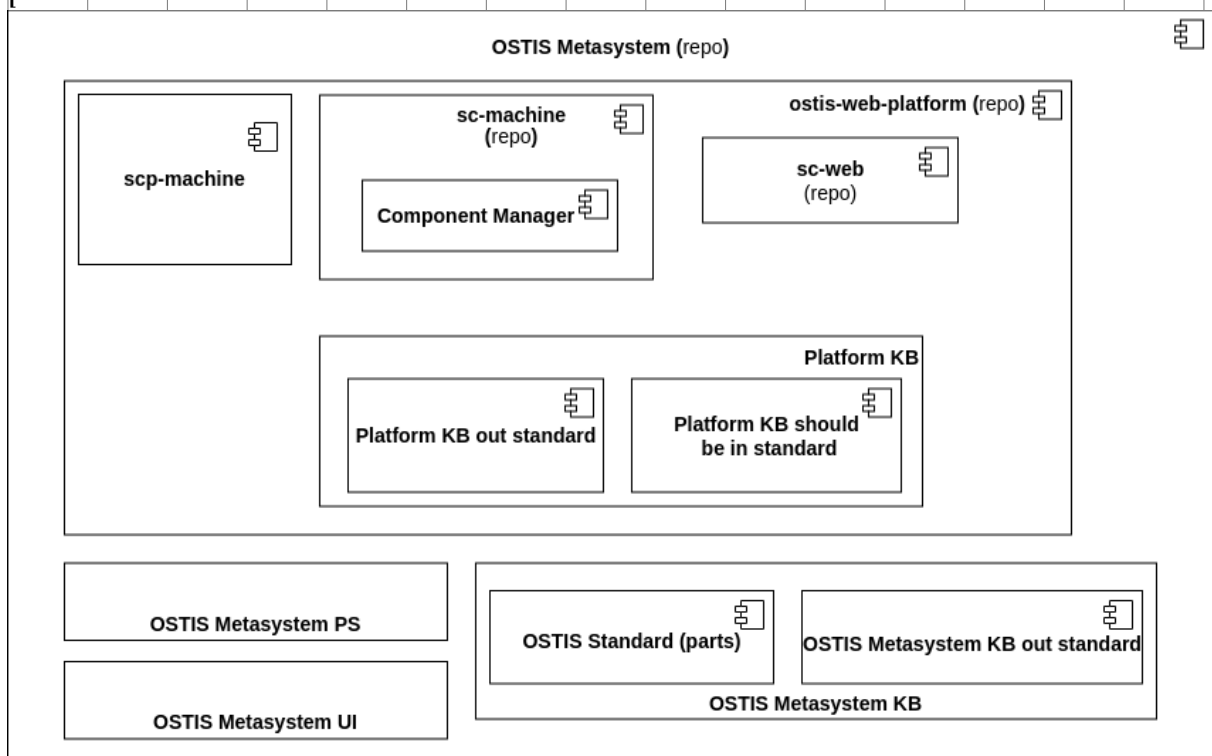


Документация Метасистемы OSTIS

Метасистема OSTIS

⇒ иллюстрация*:



Агент перевода основных и системных идентификаторов узлов из sc-памяти в текстовый файл

:= [sc-агент трансляции идентификаторов узлов из sc-памяти в текстовый файл]

⇒ задачи*:

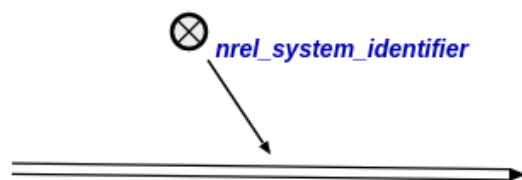
- поиск системных и основных идентификаторов узлов в sc-памяти
- проверка узлов на наличие только одного системного идентификатора и одного основного идентификатора на русском языке
- трансляция в текстовый файл является

⇒ аргументы агента*:

пустое множество

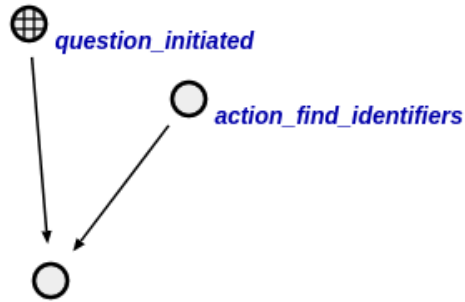
⇒ алгоритм*:

- Поиск всех узлов с помощью итератора, который ищет все конструкции вида



- Проверка каждого узла на выполнение трех условий:
 - Наличие только одного системного идентификатора.
 - Наличие только одного основного русского идентификатора.
 - Принадлежность одному из sc-типов узлов.
- Если не выполняется одно из условий, то запись данных об узле в файл не выполняется.
- Если у узла более одного системного идентификатора, то вызывается исключение.
- Если все три условия выполняются, то данные об узле записываются в файл.
- Если произошла ошибка при работе с файлом, вызывается исключение.

- ⇒ ответ агента*:
 В результате агент создает текстовый файл, в котором в виде словаря формируются структуры. Роль ключа играет основной русский идентификатор, роль значения – пара, в которой на первом месте стоит системный идентификатор, а на втором – sc-тип узла.
- ⇒ пример*:
 {“main_ru_identifier”, {“system_identifier”, “sc_type”} }
- ⇒ пример входной конструкции*:
 [



- ⇒ пример выходной конструкции*:
 [

```

2628     {"10'", {"rrel_10", "sc_node_role_relation"} },
2629     {"9'", {"rrel_9", "sc_node_role_relation"} },
2630     {"8'", {"rrel_8", "sc_node_role_relation"} },
2631     {"7'", {"rrel_7", "sc_node_role_relation"} },
2632     {"6'", {"rrel_6", "sc_node_role_relation"} },
2633     {"5'", {"rrel_5", "sc_node_role_relation"} },
2634     {"4'", {"rrel_4", "sc_node_role_relation"} },
2635     {"3'", {"rrel_3", "sc_node_role_relation"} },
2636     {"2'", {"rrel_2", "sc_node_role_relation"} },
2637     {"1'", {"rrel_1", "sc_node_role_relation"} },
  
```

Агент поиска ответа на сообщение

:= [sc-агент поиска ответа на сообщение в sc-памяти]

⇒ задачи*:

- поиск ответного действия для класса сообщения
- вызов найденного агента
- запись результата вызова найденного агента как ответа на сообщение

⇒ аргументы агента*:

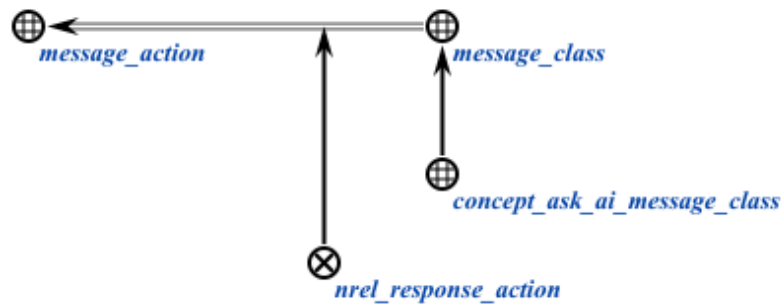
- сообщение

⇒ алгоритм*:

- [Поиск класса сообщения]
- [Проверка принадлежности класса сообщения на принадлежность классу *concept_ask_ai_message_class*]
- [Если условие не выполняется, происходит завершение работы агента]
- [Поиск соответствующего классу сообщения класса действий с помощью следующей конструкции]

⇒ пример*:

[



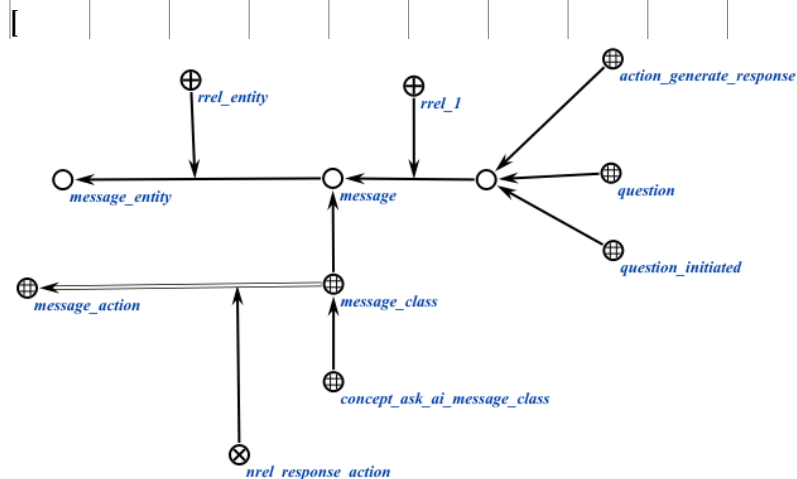
- [Если такой конструкции нет, то происходит завершение работы агента]
- [Вызов агента, соответствующего классу действий с параметрами, полученными из сообщения]
- [Ожидание завершения работы вызванного агента]
- [Если агент завершил работу успешно, то его ответ прикрепляется к сообщению отношением *nrel_response*]
- [Если агент завершил работу неуспешно, то происходит завершение работы текущего агента]

}

⇒ ответ*:

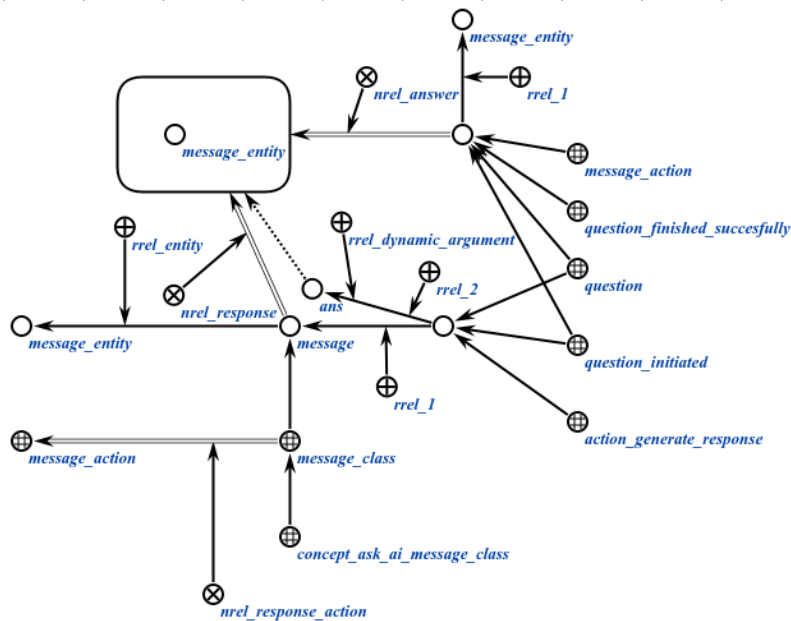
[В результате агент не выдаёт никакого ответа, однако прикрепляет полученный им ответ другого агента к сообщению]

⇒ пример входной конструкции*:



⇒ пример выходной конструкции*:

[

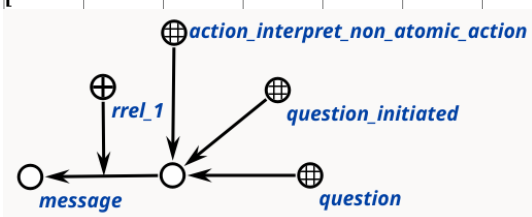


Агент ответа на сообщение

```

:= [Данный агент генерирует ответное сообщение, связанное с исходным сообщением с помощью
    отношения nrel_reply]
⇒ класс действия*:
[action_reply_to_message]
⇒ задачи*:
{
• найти ответ на сообщение
}
⇒ аргументы агента*:
{
• сообщение
⇒ пояснение*:
[linkAddr - sc-ссылка с текстом сообщения пользователя]
}
⇒ алгоритм*:
{
• [Генерация узла сообщения в базе знаний, идентификация полученного текстового файла
  как текст этого сообщения]
• [Генерация необходимой конструкции для вызова агента интерпретации неатомарного
  действия. Пример этой конструкции показан ниже]
}

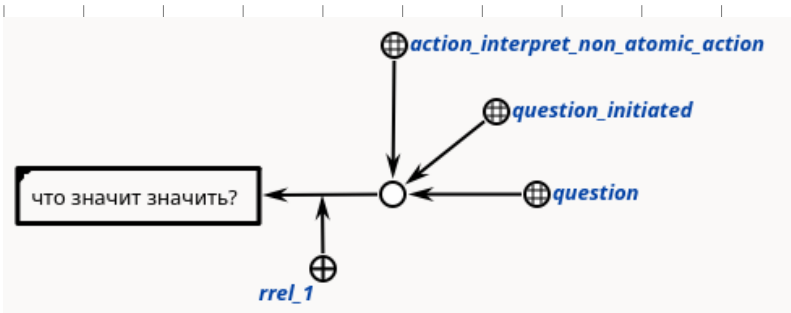
```



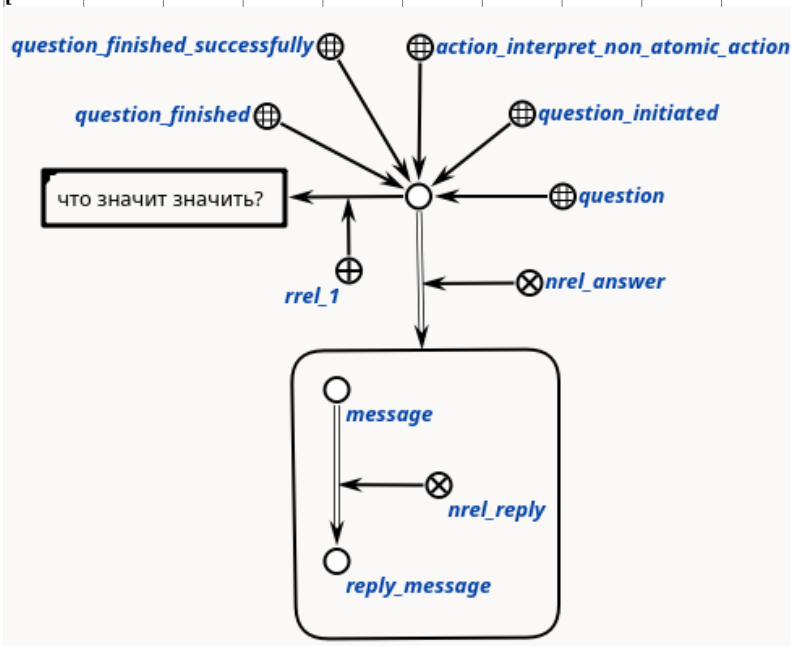
- [Ожидание завершения работы агента интерпретации и выполнение поиска ответного сообщения, сгенерированное во время работы агента интерпретации]
- [Добавление ответного сообщения к ответной структуре]

⇒ пример входной конструкции*:

[



⇒ пример выходной конструкции*:



⇒ язык реализации агента*:

[c++]

⇒ возможные результаты*:

⟨• SC_RESULT_OK

⇒ пояснение*:

[ответное сообщение сгенерировано]

• SC_RESULT_ERROR

⇒ пояснение*:

[внутренняя ошибка]

⟩

Агент классификации сообщения

:= [sc-агент классификации сообщения в sc-памяти]

⇒ класс действия*:

[action_message_topic_classification]

⇒ используемые библиотеки*:

⟨• Wit.ai

⇒ пояснение*:

[для классификации сообщений и получения сущностей]

⟩

⇒ комментарий*:

- ⟨• входное сообщение должно содержать текстовый файл с текстом на русском языке
- выделяемая сущность должна быть формализована в базе знаний

⟩

⇒ задачи*:

⇒ *определить класс сообщения*

⇒

аргументы агента:*

⇒ *сообщение*

пояснение:*

[MessageAddr - элемент класса concept_message]

⇒

алгоритм:*

⇒ [Получение текста сообщения]

• [Получение темы сообщения]

• [Получение признаков сообщения]

• [Получение сущности сообщения]

• [Если что-то не получилось получить, то происходит завершение работы агента]

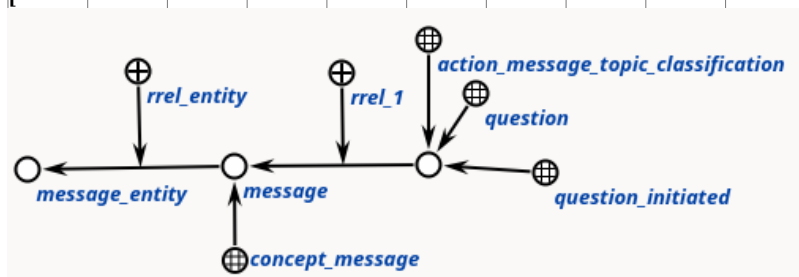
• [Если агент завершил работу успешно, то его ответ прикрепляется к сообщению отношением *nrel_entity*]

• [Если агент завершил работу неуспешно, то происходит завершение работы текущего агента]

⇒

пример входной конструкции:*

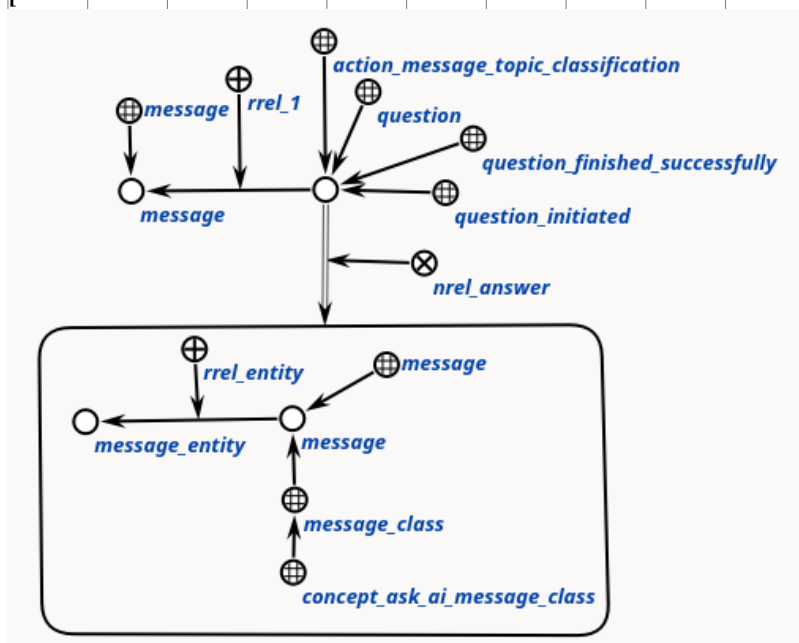
[



]

⇒ *пример выходной конструкции*:*

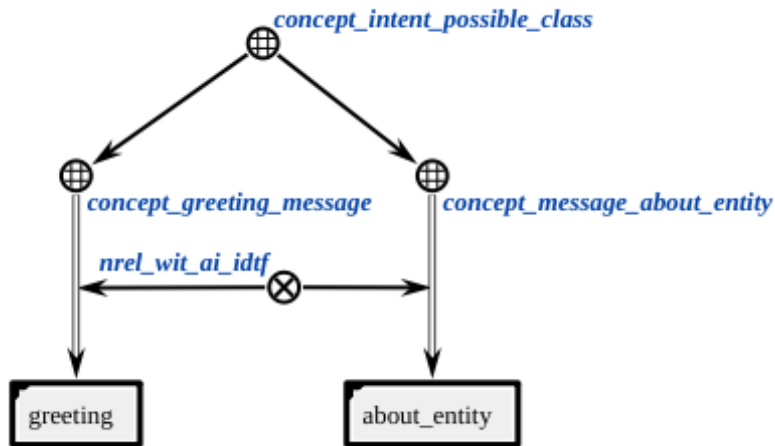
[



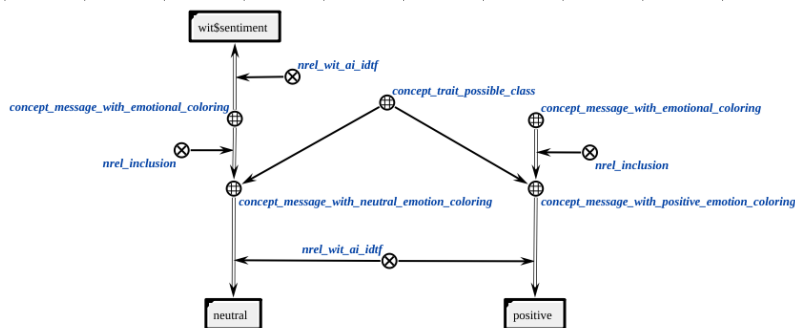
]

⇒ *Пример структуры, необходимой для классификации сообщения по теме*:*

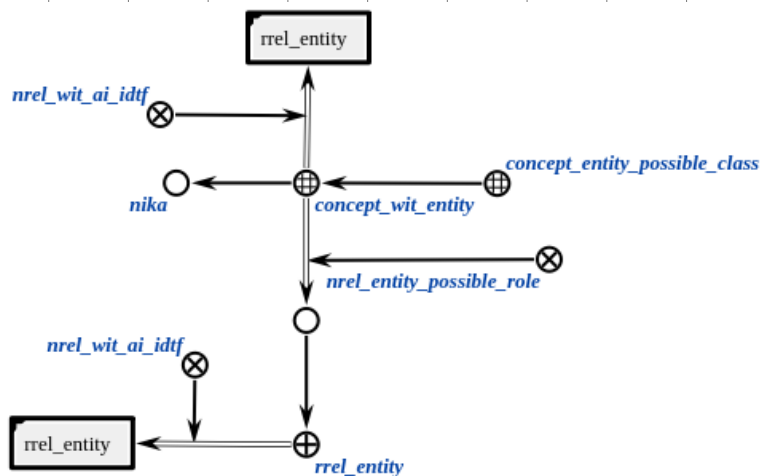
[



⇒ Пример структуры, необходимой для классификации сообщения по признакам*:



⇒ Пример структуры, необходимой для получения сущностей сообщения*:



⇒ язык реализации агента*:

[c++]

⇒ возможные результаты*:

• SC_RESULT_OK

⇒ пояснение*:

[сообщение успешно классифицировано (или произошла пустая классификация), или действие не принадлежит action_message_topic_classification]

• SC_RESULT_ERROR

⇒ пояснение*:

[внутренняя ошибка]

)

/* Section ***** */

Библиографический раздел Метасистемы OSTIS

⋃=
{
}