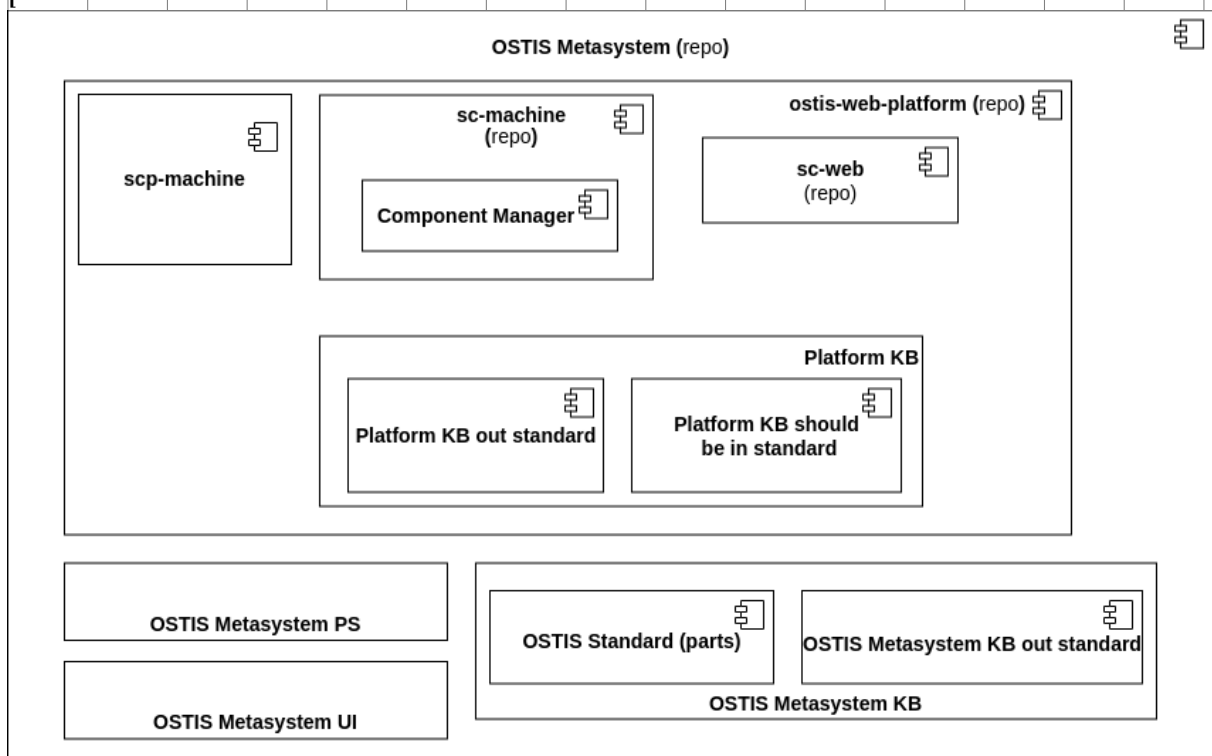


Документация Метасистемы OSTIS

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*:

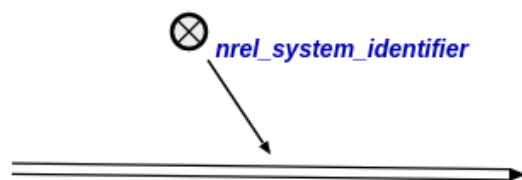
- поиск системных и основных идентификаторов узлов в *sc*-памяти
- проверка узлов на наличие только одного системного идентификатора и одного основного идентификатора на русском языке
- трансляция в текстовый файл является

⇒ аргументы агента*:

пустое множество

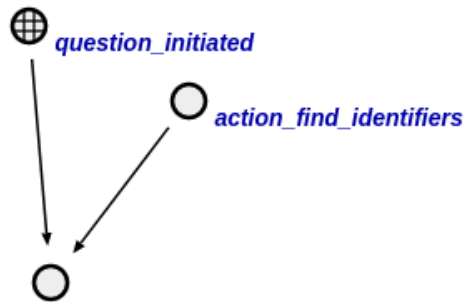
⇒ алгоритм*:

- 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*:
`[`



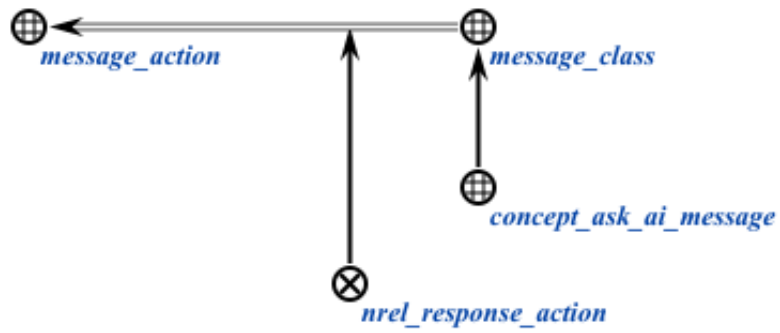
- ⇒ пример выходной конструкции*:
`[`

```

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"} },
2638     {"0'", {"rrel_0", "sc_node_role_relation"} }
  
```

Агент поиска ответа на сообщение

- := [sc-агент поиска ответа на сообщение в sc-памяти]
- ⇒ задачи*:
`<`
 - поиск агента, вызов которого является ответным действием для класса сообщения
 - вызов найденного агента
 - запись результата вызова найденного агента как ответа на сообщение`>`
- ⇒ аргументы агента*:
`<`
 - сообщение`>`
- ⇒ примечание*:
 [сообщение, для которого нужно найти ответ]
- ⇒ алгоритм*:
`<`
 - [Поиск класса сообщения]
 - [Проверка принадлежности класса сообщения на принадлежность классу *concept_ask_ai_message*]
 - [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]

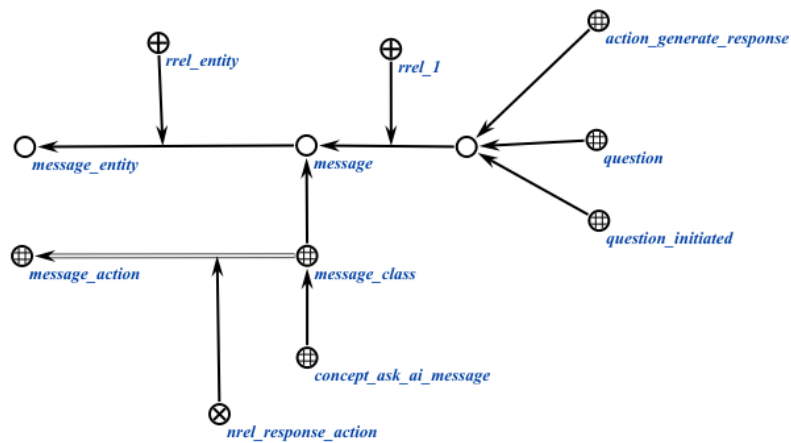
]

⇒ ответ*:

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

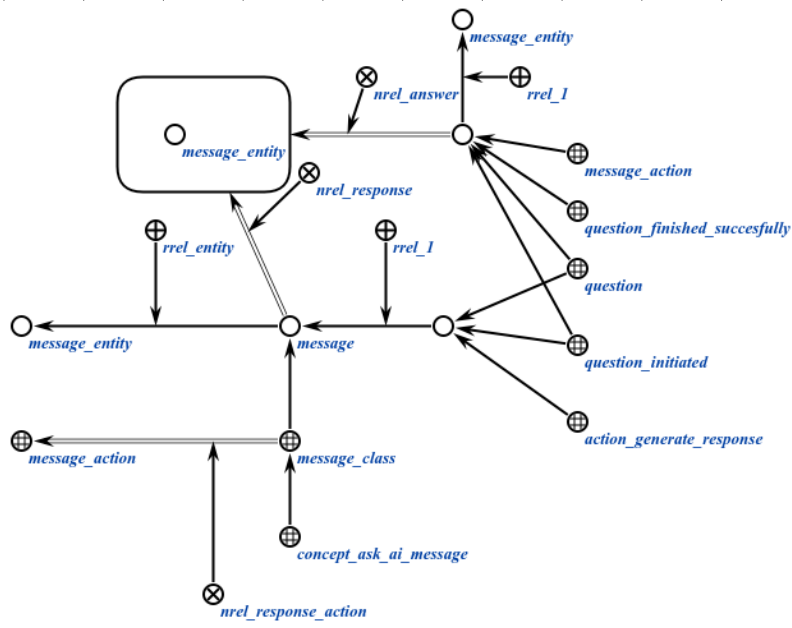
⇒ пример входной конструкции*:

[



⇒ пример выходной конструкции*:

[



/* Section ***** */

Библиографический раздел Метасистемы OSTIS

⋃=
{
}