

УДК 004.4'232  
DOI: 10.15827/0236-235X.140.680-688

Дата подачи статьи: 15.06.22  
2022. Т. 35. № 4. С. 680–688

## **Программа редактирования семиотических сетей для систем управления роботами**

П.С. Сорокоумов<sup>1</sup>, научный сотрудник, petr.sorokoumov@gmail.com

М.А. Ровбо<sup>1</sup>, научный сотрудник, rovbo@gmail.com

<sup>1</sup> Национальный исследовательский центр «Курчатовский институт»,  
г. Москва, 123182, Россия

Задачи обработки данных с графовой структурой имеют важное значение во многих практически значимых приложениях, в том числе при обработке текстов на естественном языке. Для голосового управления роботом понимание команд основано на сопоставлении семантических сетей, описывающих модель мира робота, и полученной команды. Наиболее простой и доступный для понимания эксперта способ описания такого сопоставления – система правил логического вывода, определяющих понятные роботу конструкции в результатах семантического анализа входного текста. Так как существующие средства и методы обработки либо избыточно сложны, либо недостаточно хорошо поддерживают отладку готовых систем правил, возникла необходимость в разработке специального решения.

Предлагаемое программное средство для работы с данными, представленными в сетевой модели, облегчает создание таких правил логического вывода для модификации модели мира. Совместное представление данных и методов их обработки реализовано в рамках семиотического подхода, в котором основными сущностями модели являются знаки, объединяющие сведения о состоянии объектов реального мира и возможных изменениях в этих состояниях. Применение в редакторе семиотического подхода к организации системы на основе правил позволяет связать данные и правила в рамках единой целостной картины мира интеллектуального агента.

Предоставление пользователю обоснования принятых решений в виде легко интерпретируемых списков примененных правил с перечнем параметров облегчает расширение, отладку и сопровождение разработки. Полученная система может оказаться полезной и в других предметных областях, в которых модификацию состояния системы удобно описывать правилами логического вывода.

**Ключевые слова:** семантическая сеть, сетевая модель данных, семиотика, визуальный редактор, интерфейс, робот.

Обработка данных с сетевой структурой является важной составной частью многих процессов обработки информации, и ее не всегда удается качественно автоматизировать. В связи с этим часто возникает необходимость в программных средствах удобного и понятного для человека представления сетей. Данная область сейчас активно развивается в связи с повсеместным внедрением средств инженерии знаний, в которых сетевые модели общеприняты и широко распространены. Несмотря на наличие большого количества разнообразных программ такого рода, многие предметно-ориентированные модели все еще недостаточно хорошо обрабатываются готовыми средствами, что обусловило необходимость разработки специализированной системы редактирования и отладки для одной из сетевых моделей – семиотического представления данных.

Семиотическая модель мира оказалась весьма удобной для некоторых задач высокоуровневого управления, например, в области

робототехники, в связи с чем возникает проблема редактирования соответствующих сетей и построения правил преобразования в них других представлений данных. В частности, в данной статье рассматривается применение разработанного средства редактирования сетевого представления для формирования правил перевода результата семантического разбора предложения в понятное робототехническому агенту представление команды.

В определенном смысле эта задача похожа на контекстное моделирование, описываемое в [1], если рассматривать результат лингвистического разбора команды как контекстную информацию нижнего уровня, из которой путем применения правил необходимо сформировать высокоуровневый контекст задачи для выполнения агентом. Как видно из этого исследования, существует множество различных средств работы с системами, обрабатывающими контекст на основе онтологий или логических правил, в основном для Интернета вещей, прило-

жений семантического веба, ассистивных систем и умного дома; каждое из них имеет свою специфику для решения задач одной из этих областей.

Авторами настоящей статьи рассматривается разработка средства анализа и редактирования правил для сетевого представления данных с учетом особенностей описанной робототехнической задачи.

Преобразование метаданных предметной области в понятия словаря другой области (или в более формальном случае построение отображения в соответствующую онтологию), а также разработка интерфейсов пользователя для использования систем, осуществляющих такое преобразование, все еще являются актуальными исследовательскими задачами [2], и поэтому распространенным подходом является разработка решения под конкретную практическую задачу. Один из способов – визуальное представление правил и отношений в виде графа, что, например, было применено при разработке визуальных средств обработки и редактирования систем, содержащих правила и RDF-описания для немонотонной логики в [3]. Способом представления, удобным для редактирования зависимостей экспертом, также выбрано представление зависимостей в визуализации в виде графа, основным инструментом анализа выступают методы обработки, такие как поиск кратчайшего пути на графе.

### Обработка семантических сетей при голосовом управлении мобильным роботом

Среди задач, решаемых лабораторией робототехники НИЦ «Курчатовский институт», важное место занимает управление роботами с использованием высокоуровневых команд на естественном языке, в основе которых лежат сетевые модели. Для поддержки работы интеллектуального робота с естественным языком используется подход, в котором необходимо представлять языковые выражения в некотором формализованном виде. Например, в представлении знаний робота могут встретиться сведения вида: «Робот стоит около здания», «С запада от холма находится торос», «Объект впереди является деревом». Для унификации представлений система управления роботом использует в качестве атомарных формул ограниченный набор возможных отношений между объектами, каждое из которых характеризуется фиксированным названием и набором валент-

ностей. Так, для характеристики взаимного пространственного расположения двух объектов используется отношение «находится» с тремя валентностями: про какой объект идет речь (что находится), как расположены объекты (пространственное отношение) и относительно какого второго объекта указывается положение первого (где находится). Тогда можно представить перечисленные выражения в виде: находится(торос, с запада, холм), находится(робот, около, здание), является(объект, трещина).

Если каждая атомарная формула сети отражает связь между двумя объектами, то для наглядности можно изобразить такую сеть в виде графа, вершинами которого являются упомянутые в утверждениях объекты. Дуга между вершинами существует, если соответствующие им объекты используются в одном утверждении, причем все дуги подписываются, чтобы разные утверждения об одних объектах различались. На рисунке 1 выражение «Объект является домом» представлено в виде сети с двумя вершинами и одной дугой.

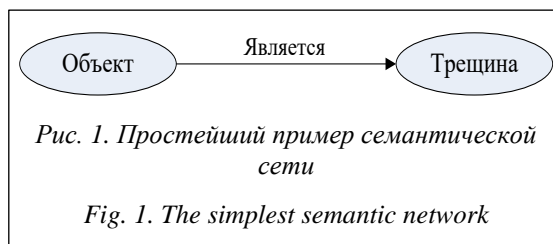
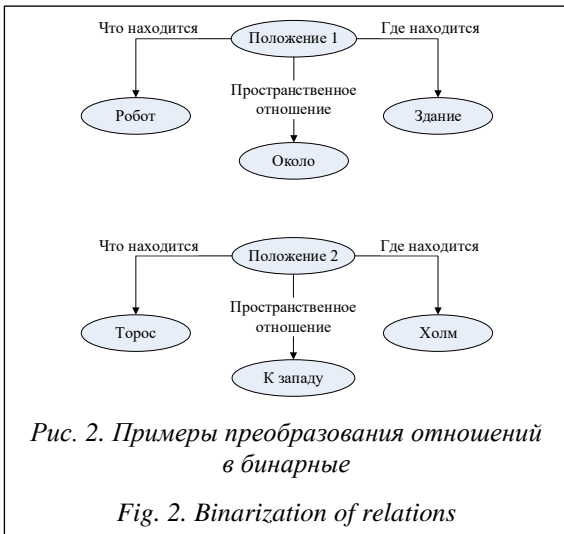


Рис. 1. Простейший пример семантической сети

Fig. 1. The simplest semantic network

В случаях, когда выражение отражает связь между тремя и более объектами, можно представить его в виде аналогичного графа, разделив на несколько отношений между двумя объектами. Для этого можно, например, добавить вершину, соответствующую самому выражению, и связать с ней упоминаемые объекты связями, соответствующими валентностям членов выражения. Примеры такого преобразования приведены на рисунке 2 для выражений «Робот стоит около здания» и «С запада от холма находится торос».

Фрагмент полной семантической сети робота представлен на рисунке 3. Как видно, сеть включает в себя и классы сущностей («Робот», «Дерево», «Строение»), и конкретных представителей классов («Дерево 1», «Сарай»). Связи между сущностями могут быть как бинарными («Является»), так и более сложными. В этом случае они реализуются с помощью введения промежуточной сущности для связи («Положение 1», «Положение 2»).



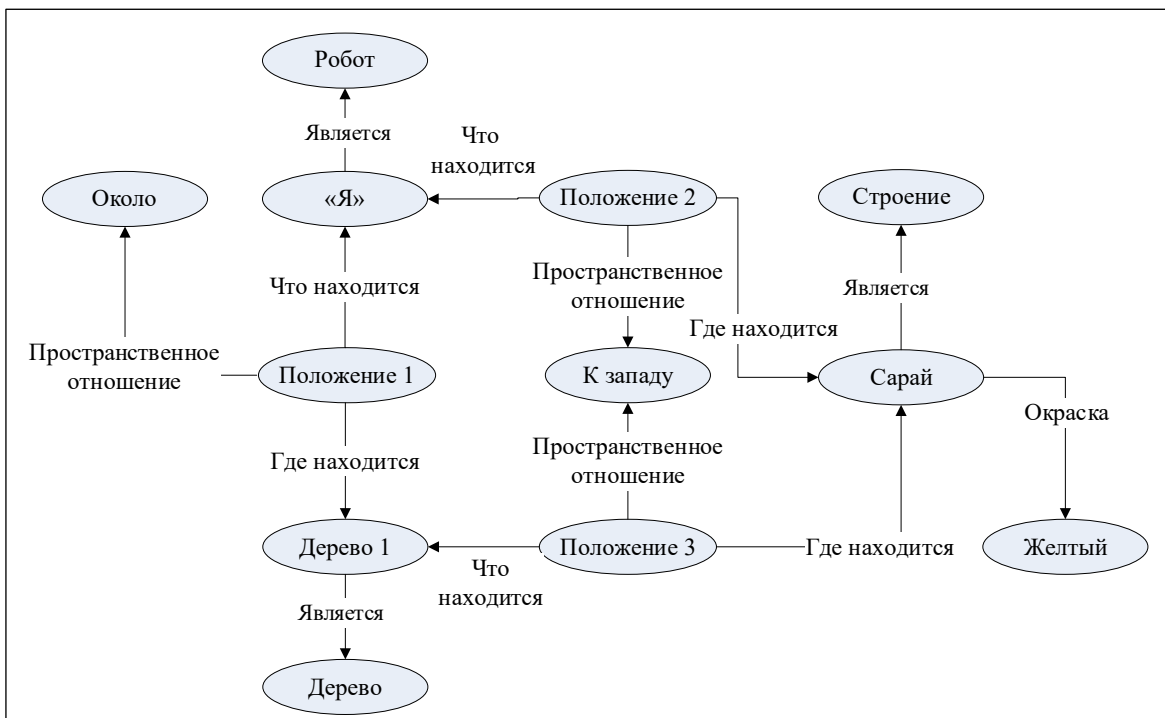
Команда оператора после распознавания также должна быть представлена в форме семантической сети, которая в дальнейшем, при обработке внутри системы управления, будет сопоставлена с семантической сетью модели мира робота. Важно отметить, что на этапе лингвистического анализа внутренние имена конкретных объектов («дерево 1», «яма 5», «строение 4») неизвестны. Чтобы сопоставление могло быть выполнено, при лингвистической обработке выявленным объектам присва-

иваются временные символические имена (X, Y, Z, ...); тогда после отправки результатов распознавания системе управления роботом она попытается сопоставить этим именам внутренние имена известных системе объектов и дополнить картину мира. Если окажется, что полученное сопоставление неоднозначно, можно будет в дальнейшем устранить эту неоднозначность запросом дополнительных признаков упомянутых объектов на основе их символических имен.

Очевидно, что подобное представление весьма гибко и подходит для самых разнообразных данных. Для описанного выше распознавания команд основной задачей является сопоставление сети с образцом-шаблоном. Целесообразно проверить, насколько хорошо поддерживается решение этой задачи существующими программными средствами.

**Основные особенности готовых средств работы с сетями**

Среди самых распространенных программ для работы с сетевыми моделями данных можно назвать Protégé [4], FluentEditor [5] и SemanticTurkey [6]. Эти и другие программы просмотра и редактирования семантических



*Рис. 3. Фрагмент семантической сети модели мира робота*

*Fig. 3. A semantic network of robot's world view*

сетей предоставляют, как правило, следующий функционал:

- непосредственно просмотр структуры сетей;
- просмотр результатов анализа сетей (статистических характеристик отдельных узлов либо клик, связности и т.п.);
- формирование запросов (например, с применением SPARQL) и просмотр их результатов в табличном виде;
- поиск данных;
- загрузка сетей из различных моделей и выгрузка их в указанном формате.

Обычно этот функционал реализуется через графический интерфейс пользователя (см. <http://www.swsys.ru/uploaded/image/2022-4/2022-4-dop/6-.jpg>), хотя используются и консольные интерфейсы.

Во многих случаях указанные задачи представления решаются ими успешно и позволяют достигать практически важных целей, но и недостатки готовых универсальных программ весьма существенны:

- слабо поддерживается отладка, то есть возможность удобно проследить историю тех или иных изменений в базе знаний, их происхождение;
- отсутствуют специфические средства для задач, отличающихся от предельно обобщенных и универсально применимых, что вызывает необходимость разработки специализированных плагинов или сторонних дополнений.

Другие довольно развитые и мощные средства анализа сетевых и графовых данных, такие как Apache Jena или AllegroGraph, поддерживают множество разнотипных моделей данных и средств описания онтологий, в том числе и распространенный стандарт OWL. Наличие встроенных средств логического вывода в принципе позволяет решать задачи рассматриваемого класса, но предлагаемые ими средства требуют избыточно сложных описаний предметной области, мешающих эксперту при их анализе. Перечисленные выше недостатки готовых средств затрудняли тестирование, диагностику и отладку, поэтому было принято решение о разработке специальной программы, имеющей достаточно развитые средства поддержки используемого класса сетевых моделей. Облегчают диагностику целевой системы также специфические особенности примененной семиотической модели представления знаний, описанные далее.

## Специфика семиотики в управлении роботами

Семиотическая система представления знаний построена с использованием определения знака по Г.С. Осипову [7] с некоторыми модификациями [8]. Минимальной единицей знаний в данной модели является отношение, то есть тройка из субъекта, предиката и объекта, описывающая некоторое истинное выражаемое предикатом соответствие между субъектом и объектом. Знак представляет собой комбинацию из четырех компонентов – имени, образа, значения и личностного смысла, причем имя знака используется только в качестве идентификатора. Образ представляет собой набор отношений, выражающих все имеющиеся истинные суждения о знаке-субъекте. Компонентами значения и личностного смысла являются правила вывода в формате MYCIN, то есть четверки ( $C^+$ ,  $C^-$ ,  $A$ ,  $D$ ), где  $C^+$  – список отношений, которые должны быть в образах каких-либо знаков системы для срабатывания правила;  $C^-$  – список отношений, которых не должно быть ни в каких образах для срабатывания правила;  $A$  – список отношений, добавляемых в образы знаков после срабатывания правила;  $D$  – список отношений, удаляемых из образов знаков после срабатывания правила. Значение и личностный смысл различаются временем срабатывания: правила значения могут запускаться в произвольное время, как только условия  $C^+$  и  $C^-$  стали истинными, а правила личностного смысла – только при явном принятии решения об их запуске специальной подсистемой. Например, знак самого робота может включать в себя образ, значение и личностный смысл.

Образ – список отношений:  $type(me, robot)$ ,  $type(me, object)$ ,  $task(me, no)$ ,  $coord(me, "(1,3)")$ .

Значение – список правил, которые должны быть выполнены сразу при удовлетворении их условий. Например, правило принудительной остановки действия может быть описано как ЕСЛИ  $urgent\_finish(me, A)$  И  $task(me, A)$  И НЕ  $memorize(me, A)$  ТО ДОБАВИТЬ  $task(me, no)$ , УБРАТЬ  $task(me, A)$ .

Данная запись дословно означает: «если мне (то есть роботу) нужно срочно завершить задачу  $A$ , причем я ее сейчас выполняю, и сохранять ее для перезапуска не требуется, то нужно добавить в мой образ сведения о том, что сейчас я не выполняю никакую задачу, и убрать из моего образа сведения о том, что я выполняю задачу  $A$ ».

Личностный смысл – список правил, решение о запуске которых должно приниматься в явном виде специальной подсистемой. Например, правило запуска процедуры поиска может выглядеть как ЕСЛИ  $type(X, action)$  И  $actor(X, me)$  И  $action\_type(X, find)$  И НЕ  $!completed(me, X)$ , И  $task(me, Y)$ , И  $X!=Y$ , И  $Y!=no$  ТО ДОБАВИТЬ  $task(me, X)$  И  $interrupts(me, Y)$ , УБРАТЬ  $task(me, Y)$ .

Дословно это правило означает: «если в модели мира существует действие  $X$ , которое должен выполнить я и которое сводится к запуску процедуры поиска, при этом я еще не завершил это действие  $X$ , а выполняю какое-то другое действие  $Y$ , то нужно начать действие  $X$ , прервать работу над  $Y$  и стереть сведения о том, что я выполняю  $Y$ ».

Отличие такого правила-личностного смысла от описанного выше правила-значения в том, что значение должно срабатывать всегда и безусловно, а решение о запуске правила-личностного смысла агент принимает сам в соответствии с некоторой процедурой, например, один раз на каждом шаге моделирования. Это сближает созданную систему с некоторыми когнитивными архитектурами, правила в которых работают по схожему принципу, например, с Soar [9].

В отличие от распространенной модели правил SWRL данный вариант описания поддерживает явное удаление информации из системы, что важно для практической реализации поставленных задач.

Работа семиотической системы управления роботом основана на постоянном выполнении шагов следующего цикла обработки:

- изменение образов знаков по данным внешних систем (сенсоров, источников команд, исполнительных процедур робота);
- выполнение всех допустимых правил-значений;
- поиск всех доступных для запуска правил-личностных смыслов;
- исполнение одного из найденных правил, выбранного специализированной системой принятия решений;
- выполнение всех допустимых правил-значений.

Разрабатываемая программа должна облегчить отладку для каждой из стадий цикла.

### Архитектура программы

Программа разработана с использованием классической архитектуры MVC. В состав мо-

дели входят компонент хранения, загрузчик данных формата RDF (посредством библиотеки *rdflib*) и семиотический обработчик, способный менять состояние модели и выдавать при этом список сработавших правил с примененными подстановками.

Интерфейс полученного приложения, реализованного на языке программирования Python, показан на рисунке 4. Как видим, загруженная семиотическая модель (сверху слева) обработала входные данные (слева снизу показаны в текстовом виде, в центре – в графическом) путем выработки новых отношений правилами (показаны справа). Из графа извлечена цепочка зависимостей, пригодная для формирования нового правила для набора (слева сверху).

Семиотические сети в разработанной системе могут быть представлены как в текстовом, так и в графическом виде. Текстовое представление знаков выполняется в специально разработанном формате, в котором для каждого знака указываются его компоненты (образ  $p$ , значение  $m$ , личностный смысл  $a$ ). При этом обработка картины мира и входных данных, способных поменять эту картину, выполняется отдельно. Для облегчения восприятия и картину мира, и входные данные можно представить в графическом виде, как показано на рисунке 4 в центре.

Для облегчения построения новых правил в программу добавлена возможность автоматического построения цепей отношений от одной выделенной сущности к другой. Обход графа при построении набора выполняется по алгоритму Дейкстры. Выделенные сущности отмечаются цветом.

Для контроля результатов используется панель вывода данных о сработавших при семиотическом анализе правилах вместе с использованными подстановками (справа снизу).

### Применение программы

Программа используется для отладки систем правил. После загрузки модели мира из файла, содержащего образы, значения и личностные смыслы знаков в виде отношений и правил, программа может обработать входные данные как семиотическую сеть, задаваемую в формате RDF. После окончания обработки отображаются появившиеся в картине мира отношения, а также список сработавших правил с указанием на то, какие именно отношения были ими добавлены и убраны из модели мира.

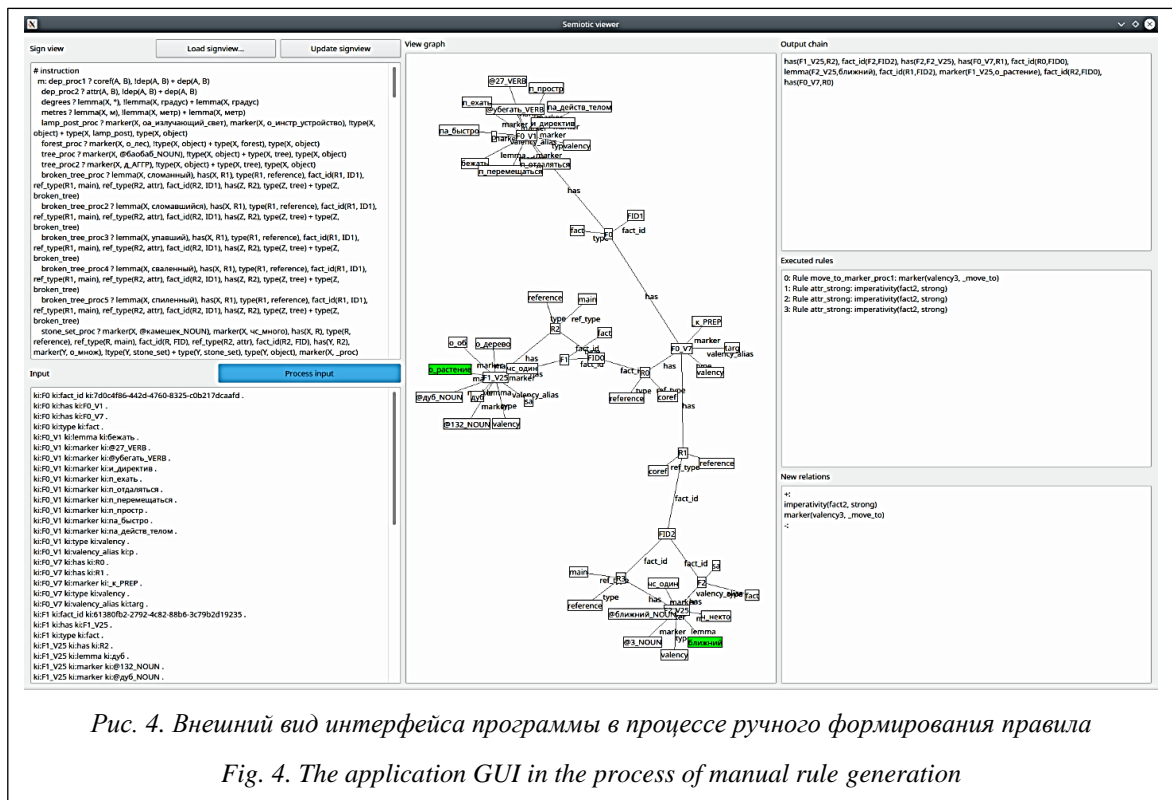


Рис. 4. Внешний вид интерфейса программы в процессе ручного формирования правила  
 Fig. 4. The application GUI in the process of manual rule generation

Таким образом, программа позволяет обосновать вывод о наличии ошибки в конкретном правиле или отследить, какие именно входные данные вызвали неверный вывод.

Для облегчения редактирования системы правил используется специально созданная процедура получения цепочек отношений, связывающих две или более сущности во входных данных. Далее эти цепочки можно использовать для записи нового правила, позволяющего перенаправить обработку и предсказуемым образом изменить результат.

В качестве примера использования системы рассмотрим задачу формирования команд для робота по результатам их семантического анализа. Используемый компонент лингвистической обработки превращает выражение естественного языка в набор отношений, где субъектами и объектами служат части фразы и их признаки, а предикатами – распознанные смысловые связи между ними. Например, лингвистический анализатор разбирает фразу «иди к человеку» следующим образом:

```
F0 fact_id 8ba4c55e-d6e3-4665-aaf7-a2ae2864310d.
F0 has F0_V1.
F0 has F0_V7.
F0 type fact.
F0_V1 lemma идти.
F0_V1 marker @212_VERB.
```

```
F0_V1 marker @идти VERB.
F0_V1 marker и директив.
F0_V1 marker п ехать.
F0_V1 marker п перемещаться.
F0_V1 marker па действ_телом.
F0_V1 type valency.
F0_V1 valency_alias p.
F0_V7 lemma человек.
F0_V7 marker @29_NOUN.
F0_V7 marker @женщина_NOUN.
F0_V7 marker _к_PREP.
F0_V7 marker ч некто.
F0_V7 marker ч признак.
F0_V7 marker чс один.
F0_V7 type valency.
F0_V7 valency_alias tar.
```

Признаки сущностей обозначены здесь предикатом marker, базовые формы слов – предикатом lemma, роли сущностей во фразе (валентности) – предикатом valency\_alias. Очевидно, что напрямую отдавать команды роботу в таком виде нежелательно, так как определить их смысл будет крайне затруднительно. Поэтому необходимо преобразовывать подобные конструкции в лишенные избыточности команды, причем предсказуемо и интерпретируемо. Это преобразование реализовано посредством специально созданной системы правил, которая находит в разборе шаблонные конструкции и добавляет к отношениям соот-

ветствующие им известные роботу виды предикатов. Например, правило, распознающее в этом описании человека, выглядит следующим образом: ЕСЛИ *marker(X, @женщина\_NOUN)* И НЕ *lemma(X, ты)* И НЕ *lemma(X, себя)* ТО ДОБАВИТЬ *type(X, human)* И *type(X, object)*.

Разработанная программа позволяет получать сведения о примененных правилах в удобном и доступном для интерпретации виде. Список правил, использованных в разборе указанной фразы, имеет вид

1. *human\_proc: X=F0\_V7*
2. *move\_to\_marker\_proc1: X=F0\_V1*
3. *move\_to\_proc\_1: F=F0, X=F0\_V1, Y=F0\_V7*

Так как все правила в программе доступны для просмотра, по такому списку можно восстановить ход разбора и при необходимости скорректировать правила.

Если какая-то фраза не разбирается, программа позволяет быстро получить новое правило и проверить его работу. Например, для разбора команды «уменьши расстояние между тобой и человеком», которая должна интерпретироваться так же, как команда «поезжай к человеку», потребовалось создавать новое правило со сложным условием: *marker(Y, п\_уменьшать), type(Y, valency), marker(Z, о\_расстояние), type(Z, valency), has(X, Y), has(X, Z), type(X, fact), valency\_alias(G, sa), type(G, object), !type(G, robot), has(G, R1), ref\_type(R1, main), fact\_id(R1, FID1), ref\_type(R2, coref), fact\_id(R2, FID1), has(IMF, R2), marker(IMF, \_между\_PREP), type(Subj, robot), has(Subj, R3), ref\_type(R3, main), fact\_id(R3, FID2), ref\_type(R4, coref), fact\_id(R4, FID2), has(IMF, R4)*.

Построение данного условия в программе автоматизировано: пользователь должен выделить вершины семантической сети, а цепь связей между ними будет найдена автоматически. Применение созданной программы позволило значительно снизить затраты времени на пополнение системы команд робота, особенно при разборе синонимичных конструкций наподобие «поезжай к человеку», «надо ехать к человеку», «быстро, к человеку!», «поедем к человеку» и т.п. Тестирование правильности получаемых разборов, позволяющее не нарушать работу ранее введенных правил при пополнении системы, выполняется отдельной программой.

### Заключение

Была разработана программа для редактирования семиотической модели данных на основе логического вывода, способная функционировать как в среде Windows, так и в среде Linux. Во втором случае возможно ее подключение к реальной или моделируемой робототехнической системе, работающей под управлением ROS. На примере обработки команд на естественном языке показано, что программа позволяет при интенсивном дополнении набора правил обработки сократить его, разделить на модули и ускорить вывод, что повышает эффективность работы эксперта над конкретными прикладными задачами. Перспективным представляется использование системы в прочих задачах обработки сетевых данных с помощью системы правил логического вывода.

*Работа выполнена при поддержке НИИЦ «Курчатовский институт» (приказ № 2754 от 28.10.2021).*

### Литература

1. Pradeep P., Krishnamoorthy S. The MOM of context-aware systems: A survey. *Computer Communications*, 2019, vol. 137, pp. 44–69. DOI: 10.1016/j.comcom.2019.02.002.
2. Munir K., Anjum M.S. The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics*, 2018, vol. 14, no. 2, pp. 116–126. DOI: 10.1016/j.aci.2017.07.003.
3. Kontopoulos E., Bassiliades N., Antoniou G. Deploying defeasible logic rule bases for the semantic web. *Data & Knowledge Engineering*, 2008, vol. 66, no. 1, pp. 116–146. DOI: 10.1016/j.datak.2008.02.005.
4. Musen M.A. The protégé project. *AI Matters*, 2015, vol. 1, no. 4, pp. 4–12. DOI: 10.1145/2757001.2757003.
5. Cognitum Software House. *FluentEditor*. URL: <https://cognitum.eu/semantics/fluenteditor/> (дата обращения: 12.06.2022).
6. Paziienza M.T., Scarpato N., Stellato A., Turbati A. Semantic Turkey: A browser-integrated environment for knowledge acquisition and management. *Semantic Web*, 2012, vol. 3, no. 3, pp. 279–292. DOI: 10.3233/sw-2011-0033.

7. Панов А.И., Осипов Г.С., Чудова Н.В., Кузнецова Ю.М. Знаковая картина мира субъекта поведения. М.: Физматлит, 2018. 264 с.

8. Sorokoumov P.S., Rovbo M.A., Moscovsky A.D., Malyshev A.A. Robotic wheelchair control system for multimodal interfaces based on a symbolic model of the world. In: *Studies in Systems, Decision and Control*, 2021, pp. 163–183. DOI: 10.1007/978-3-030-68172-2\_14.

9. Laird J.E., Rosenbloom P. *The Evolution of the Soar Cognitive Architecture*. 1994, 45 p.

Software & Systems  
DOI: 10.15827/0236-235X.140.680-688

Received 15.06.22  
2022, vol. 35, no. 4, pp. 680–688

### Semiotic network editing software for robot control systems

**P.S. Sorokoumov**<sup>1</sup>, *Research Associate, petr.sorokoumov@gmail.com*

**M.A. Rovbo**<sup>1</sup>, *Research Associate, rovboma@gmail.com*

<sup>1</sup> *National Research Centre “Kurchatov Institute”, Moscow, 123182, Russian Federation*

**Abstract.** The problem of processing data structured as a graph is important for applications in many domains, including natural language text processing. For voice control of a robot, command interpretation is based on matching between semantic networks that describe a robot’s world model and a received instruction. The simplest and most understandable way for an expert to describe such matching process is a system of inference rules that determine constructs recognizable for a robot in the results of the semantic analysis of text. Since the existing processing tools and methods are either overly complex or do not support the debugging of complex rule systems well enough, there is a need in a special solution.

The proposed software tool for network data processing facilitates the development of such inference rules for modifying a world model. The joint representation of data and their processing methods is implemented within the framework of a semiotic approach, which has signs that combine information about the state of entities in the real world and possible changes in this state as the main entities of the model. Application of the semiotic approach to organizing a rule-based system allows linking data and rules within a single world model of an intelligent agent.

Providing a user with a rationale for the decisions made in the form of easily interpretable lists of applied parameterized rules facilitates extension, debugging and maintenance of the system. The developed software can also be useful in other domains where it is convenient to describe the system state modification by logical inference.

**Keywords:** semantic network, network data model, semiotics, visual editor, interface, robot.

**Acknowledgements.** *The project is supported by NRC “Kurchatov institute” (order no. 2754 dated 28.10.2021).*

### References

1. Pradeep P., Krishnamoorthy S. The MOM of context-aware systems: A survey. *Computer Communications*, 2019, vol. 137, pp. 44–69. DOI: 10.1016/j.comcom.2019.02.002.

2. Munir K., Anjum M.S. The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics*, 2018, vol. 14, no. 2, pp. 116–126. DOI: 10.1016/j.aci.2017.07.003.

3. Kontopoulos E., Bassiliades N., Antoniou G. Deploying defeasible logic rule bases for the semantic web. *Data & Knowledge Engineering*, 2008, vol. 66, no. 1, pp. 116–146. DOI: 10.1016/j.datak.2008.02.005.

4. Musen M.A. The protégé project. *AI Matters*, 2015, vol. 1, no. 4, pp. 4–12. DOI: 10.1145/2757001.2757003.

5. *Cognitum Software House. FluentEditor*. Available at: <https://cognitum.eu/semantics/fluenteditor/> (accessed June 12, 2022).

6. Paziienza M.T., Scarpato N., Stellato A., Turbati A. Semantic Turkey: A browser-integrated environment for knowledge acquisition and management. *Semantic Web*, 2012, vol. 3, no. 3, pp. 279–292. DOI: 10.3233/sw-2011-0033.



7. Panov A.I., Osipov G.S., Chudova N.V., Kuznetsova Yu.M. *Semiotic World View of Behavior Subject*. Moscow, 2018. 264 p. (in Russ.).

8. Sorokoumov P.S., Rovbo M.A., Moscovsky A.D., Malyshev A.A. Robotic wheelchair control system for multimodal interfaces based on a symbolic model of the world. In: *Studies in Systems, Decision and Control*, 2021, pp. 163–183. DOI: 10.1007/978-3-030-68172-2\_14.

9. Laird J.E., Rosenbloom P. *The Evolution of the Soar Cognitive Architecture*. 1994, 45 p.

#### Для цитирования

Сорокоумов П.С., Ровбо М.А. Программа редактирования семиотических сетей для систем управления роботами // Программные продукты и системы. 2022. Т. 35. № 4. С. 680–688. DOI: 10.15827/0236-235X.140.680-688.

#### For citation

Sorokoumov P.S., Rovbo M.A. Semiotic network editing software for robot control systems. *Software & Systems*, 2022, vol. 35, no. 4, pp. 680–688 (in Russ.). DOI: 10.15827/0236-235X.140.680-688.