

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Научная статья

УДК 004.04

DOI: 10.18101/2304-5728-2022-2-102-111

ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА ОБРАБОТКИ ДИНАМИЧЕСКИХ ОТНОШЕНИЙ

© Федорченко Людмила Николаевна

кандидат технических наук, старший научный сотрудник,
Санкт-Петербургский федеральный исследовательский центр
Российской академии наук
Россия, 199178, г. Санкт-Петербург, 14 Линия В.О., 39
Санкт-Петербургский государственный университет
Россия, 199034, г. Санкт-Петербург, Университетская наб., 7/9
lnf@iiias.spb.su

© Гейда Александр Сергеевич

доктор технических наук, ведущий научный сотрудник,
Санкт-Петербургский федеральный исследовательский центр
Российской академии наук
Россия, 199178, г. Санкт-Петербург, 14 Линия В.О., 39
geida@iiias.spb.su

Аннотация. В настоящее время внедрение новых информационных технологий (цифровых) в производство и другие отрасли народного хозяйства привело к развитию ряда современных направлений исследований, связанных с разработкой систем искусственного интеллекта, задач обработки динамических знаний, задач выработки решений в предметных областях с динамической структурой. В дальнейшем необходимо развивать программные средства поддержки таких технологий обработки динамических знаний, что заключается в создании комплекса инструментальных средств с хорошим пользовательским интерфейсом для решения различных задач принятия решений в областях с динамикой. Будем полагать, что формальное описание динамики структуры предметной области задается системой с динамическими отношениями [1], которые являются отношениями на множестве объектов, зависящими от времени.

Статья посвящена описанию прототипа программной системы, которая реализует алгебру операций над динамическими отношениями (циклограммами) и отображает их на экран в графическом виде. Графическая интерпретация отношений по сравнению с текстовой и табличной более наглядна, а использование набора операций позволяет быстрее осмысливать результаты и управлять процессом решения задачи.

Ключевые слова: динамические знания, динамические отношения, инструментальная система, алгебра динамических отношений.

Для цитирования

Федорченко Л. Н., Гейда А. С. Инструментальная система обработки динамических отношений // Вестник Бурятского государственного университета. Математика, информатика. 2022. № 2. С. 102–111.

Введение

К настоящему моменту ясно, что развитие программных средств поддержки разных видов человеческой деятельности заключается в создании комплекса инструментальных средств с хорошим пользовательским интерфейсом для моделирования данного вида деятельности.

Актуальность работы вызвана тем, что сегодня практически полностью отсутствуют развитые инструментальные средства автоматизации процесса разработки отдельных частей гибридной экспертной системы (ЭС) с динамическими знаниями, с помощью которых реализуются представление и обработка этих динамических знаний.

Эффект гибридности ЭС достигается не только за счет непосредственного включения отдельных программных средств в систему принятия решений, но и возможностью обращения из данной системы к другим приложениям или модулям, содержащим нужные средства. Поэтому в технологии ЭС отдельные средства удобно оформлять в виде отдельных блоков-модулей или приложений (например, табличное представление данных, экранное представление динамических отношений, графический редактор и т. д.). Отсюда предъявляются требования, в каком направлении должна идти работа по автоматизации процесса разработки прикладных систем с динамическими знаниями.

Опыт работы по созданию прототипа гибридной ЭС позволяет считать, что на «выходе» инструментальной системы должен быть отдельный модуль на языке высокого уровня, который практически не требует корректировки и может быть без усилий подключен к основному приложению.

Каждая ЭС имеет свой пользовательский интерфейс, это те атрибуты системы, которые определяют взаимодействие между пользователем и самой системой. В последние годы стал совершенно привычным экранный характер интерфейса в современных диалоговых системах. В зарубежных публикациях пользовательский интерфейс определен как наиболее важный элемент ЭС. Более того, интерфейс может влиять на качество принятия решения.

Для описания хорошего пользовательского интерфейса часто используется термин «дружественный интерфейс». Однако понятие «дружественности» точно не определено и является достаточно расплывчатым. Характеристика интерфейса, которая является «дружественной» в одной ситуации, не является «дружественной» в другой. Например, характеристика, «дружественная» для пользователя-новичка, «не дружественна» для пользователя-эксперта, и наоборот. Со временем пользователь-новичок сам становится опытным пользователем ЭС. Более того, он может быть опытным в использовании одних возможностей ЭС и новичком — в других. Следовательно, интерфейс будет «дружественным» (или продолжать

оставаться им), если он может легко адаптироваться к пользователю независимо от стилистических особенностей.

В программистской среде уже предъявляются требования к организации пользовательского интерфейса в прикладных системах. Рассматривая характеристики современных программных систем различного назначения, можно заметить, что сформировался целый ряд структур, которые обычно участвуют в организации пользовательского диалога. К таким структурам можно отнести меню, таблицу, систему окон, диаграмму и др. Количество таких структур невелико, и пользователь, начиная работать с новой системой, как правило, довольно быстро овладевает всеми возможностями диалога.

Следует подчеркнуть, что затраты на программирование интерфейса, как правило, больше (или сравнимы с) затрат(ми) на программирование функционального наполнения прикладной системы. Это относится в равной степени и к организации диалога в системе обработки динамических отношений. Поэтому одним из требований, предъявляемых к разрабатываемой инструментальной системе, является создание библиотеки процедур, реализующих диалоговые возможности в соответствии с упомянутыми традиционными стереотипами, и использование этих процедур в отторгаемом от инструмента программном модуле.

Данная статья посвящена описанию прототипа программной системы, которая реализует операции над динамическими отношениями (циклограммами) и отображает их на экран в графическом виде.

1 Основные возможности системы

Описываемая инструментальная система генерирует циклограммы бинарных динамических отношений [1].

Далее циклограммой (или экземпляром циклограммы) будем называть определенную нелинейную структуру в оперативной памяти компьютера, в которой содержится множество значений индикаторных функций бинарного динамического отношения на заданном множестве элементов $\{a_i, b_j\}$ в фиксированном интервале времени Δ .

Помимо этого циклограмма содержит дополнительную служебную информацию. Часть этой информации доступна пользователю (например, имя циклограммы или имя файла, хранящего данную циклограмму на диске), т. е. циклограмма обладает рядом атрибутов, значения которых пользователь может менять.

В качестве поддержки экранного взаимодействия система использует интерфейсные библиотеки PyQt5¹ для пользовательского интерфейса и Matplotlib² для отображения динамических отношений в виде циклограмм (набора импульсов), графиков.

¹ Документация PyQt5. URL: <https://doc.qt.io/qtforpython/> (дата обращения: 14.02.2022).

² Документация matplotlib. URL: <https://matplotlib.org/> (дата обращения: 23.02.2022).

Окном циклограммы будем называть часть экрана, отводимую для отображения значений индикаторных функций (временных импульсов) и множества элементов, содержащихся в циклограмме.

Инструментальная система обработки циклограмм предоставляет своим пользователям следующие возможности:

1. Генерирование произвольного количества циклограмм во внутреннем представлении с последующим сохранением их на диске.
2. Загрузка индикаторных функций на множестве пар элементов в две циклограммы из дисковых файлов для выполнения операции.
3. Индикаторные функции и другие доступные пользователю атрибуты циклограммы с произвольным именем.
4. В любой момент времени можно открыть окно циклограммы, отобразить ее на экране и перейти к редактированию ее атрибутов.
5. Окно циклограммы может быть любого размера и цветовой гаммы.
6. Значение глобального интервала времени, на котором определено динамическое отношение, в любой момент работы может быть изменено.
7. Библиотека операций над циклограммами обеспечивает набор теоретико-множественных и некоторых алгебраических операций, как унарных, так и бинарных.

8. Интерфейсная библиотека содержит процедуры, позволяющие проводить редактирование в окнах для двух циклограмм одновременно.

Таким образом, основными компонентами системы являются:

- интерфейс с операционной системой, позволяющий загружать циклограммы в систему и сохранять вновь созданные на диске;
- библиотеки оконного интерфейса, предназначенные для задания окон для циклограмм;
- экранный редактор циклограмм, изменяющий атрибуты циклограммы. В частном случае редактор может создать новое динамическое отношение в пустом окне;
- библиотеки операций над множествами и динамическими отношениями.

Как уже упоминалось, в любой момент времени пользователь системы может инициализировать максимум две циклограммы. Затем он может либо выполнить операцию над этими циклограммами и получить в качестве результата новый экземпляр циклограммы, либо передать активность одной из циклограмм, а управление — процедуре редактирования атрибутов данной циклограммы, которой соответствует активное окно.

Перечислим некоторые атрибуты циклограммы, существенные для пользователя. С точки зрения языка реализации экземпляр циклограммы идентифицируется структурой данных, в которую атрибуты входят в качестве полей некоторого уровня. При инициализации экземпляра циклограммы ее атрибуты получают значения по умолчанию. Атрибут *name* типа `string[10]` сопоставляет циклограмме имя. Это имя должно присутствовать в заголовке окна циклограммы.

Атрибут *fname* типа `string[20]` также сопоставляет циклограмме имя. Это имя файла с точки зрения операционной системы, оно может включать в себя путь по дереву файлов. Например, `'\task\c1.ckl'`. Атрибут *fname* всегда должен задаваться явно.

Группа атрибутов отвечает за вид окна циклограммы. Атрибуты *x1s*, *y1s*, *x2s*, *y2s* типа `integer` задают координаты окна циклограммы на экране.

Атрибут *frame* типа `boolean` задает наличие или отсутствие рамки вокруг окна циклограммы.

Группа атрибутов, объединенных в структуру, задает цветовую гамму циклограммы. Атрибуты с именами *color.a_field*, *color.p_field*, *color.title*, *color.frame* типа `byte` задают атрибуты цветности для интервальных полей индикаторных функций (в активном/пассивном режиме), для заголовка окна, рамки окна. Эти атрибуты циклограммы в программе пользователя могут быть изменены явным образом: любому атрибуту можно присвоить новое значение путем выполнения оператора присваивания соответствующему полю структуры данных этой циклограммы.

Для изменения цветовой гаммы циклограммы можно использовать присваивание структуре полей *color* стандартного набора атрибутов цветности, описание которого включено в библиотеку оконного интерфейса.

2 Режимы работы главного меню

Управление системой осуществляется через главное меню с набором подменю и запросами в виде возникающих и исчезающих на экране текстовых окон. Слова главного меню отражают основные компоненты системы и стадии (режимы) работы с каждым экземпляром циклограммы.

FILE. Реализует интерфейс с операционной системой, который выполняется с помощью подменю со словами **LOAD**, **SAVE**, **SAVE AS**, **RENAME**, **ERASE**, **QUIT**. Данный режим осуществляет инициализацию циклограммы. Под циклограмму отводится необходимое количество оперативной памяти, инициализируются атрибуты циклограммы, происходит загрузка индикаторных функций. При этом никакого изменения изображения на экране не происходит.

WINDOW. В этом режиме создается окно циклограммы. С точки зрения интерфейсной системы оно содержит два окна:

- статическое окно для хранения постоянной для данной циклограммы информация (заголовок в начале экрана, задаваемый интервал времени $\Delta = [t_0, t_k]$ отображения циклограммы и подсказки внизу экрана);
- динамическое окно, в которое погружаются отображаемые индикаторные функции и элементы множеств из $A \times B$, формирующих отношение. Размер этого окна определяется динамически и зависит от количества строк, отводимых под элементы множества B .

По умолчанию размеры статического окна совпадают с 1/2 экрана. Но пользователь может сам изменить атрибуты окна:

- задать максимальные размеры логического окна, которые могут существенно превышать размеры экрана;

- переопределить координаты физических окон на экране;
- можно задать другие атрибуты окон: тип рамки (одинарная, двойная или никакой), цвет рамки, основной цвет окон, цвет выводимых в окна интервальных полей.

TIME. Режим задания глобального интервала времени $\Delta = [t_0, t_k]$. В настоящее время в виде день, час, минуты, секунды.

SET. В этом режиме пользователь может задать пару множеств A и B для бинарного динамического отношения и при необходимости отредактировать их (удалить некоторые элементы или добавить новые).

В данной версии A и B загружаются из файла, но существует возможность задать их с экрана.

EDIT. После того как циклограмма, заданная отношением $SKL R(A, B, [t_0, t_k])$, инициализирована, она может быть открыта. При открытии происходит отображение окна циклограммы на экран и управление передается процедуре, которая редактирует интервальные поля и элементы из $A \times B$. Пользователь имеет возможность посмотреть на экране загруженную циклограмму в качестве первого или второго операнда для дальнейшего выполнения операций. Он может «пролистать» на экране циклограммы для каждого $a_i \in A$ и на выбранных циклограммах $SKL R(a_i, B, [t_0, t_k])$ выполнить редактирование отображаемых интервалов (удалить, вставить, переместить по горизонтали, изменить значение начала или конца интервала). При выходе из этой процедуры окно циклограммы исчезает с восстановлением прежнего изображения. Далее возможно повторное открытие циклограммы в качестве второго операнда.

UNITARY. Режим выполнения унарных операций. При переходе в этот режим пользователь видит на экране список имен реализованных операций. С помощью клавиш-стрелок выбирается нужное имя и с активной циклограммой выполняется операция.

BINARY. Режим выполнения бинарных операций. Так же, как в случае унарных операций, подменю содержит список имен бинарных операций. В качестве второго операнда берется активная циклограмма, а первым операндом является та, что была загружена на предыдущем шаге в режиме **FILE**.

Файлы, сгенерированные в результате работы в режимах операций, получают имя $\langle \text{предикат} \rangle \text{старое имя} .skl$, где $\langle \text{предикат} \rangle$ — это имя выполняемой операции (например, NOT, Un, Subtr и др.), а $\langle \text{старое имя} \rangle$ — это первоначальное имя загруженного файла. Затем основное имя усекается до нужной длины и добавляется расширение $.skl$. Пользователь может также задать новое имя созданной циклограммы. В режиме **EDIT** редактирование проводится на копии файла, поэтому пользователю предоставлена возможность либо сохранить копию с текущим именем (**SAVE**), либо с новым именем (**SAVE AS**).

3 Формирование циклограммы

Формирование циклограммы начинается после ее инициирования в режиме **FILE**, т.е. после сопоставления ей имени некоторого файла.

Обозначим

$$Arg(CKL R) = \{ \langle a_i, b_j \rangle | \langle a_i, b_j \rangle \in A \times B \},$$

$$Val(CKL R) = \{ \Delta_k^{i,j} | \Delta_k^{i,j} \subset \Delta \} \text{ для } CKL R(A, B, \Delta).$$

Определим для каждого $a_i \in A$

$$Val(\langle a_i, b_j \rangle) = \{ \Delta_{i,j} | \langle a_i, b_j \rangle \in R \text{ для } \forall j \}.$$

Тогда $Val(CKL R(a, B, \Delta)) = \bigcup_j Val(\langle a, b_j \rangle)$, так что

$$\langle a, b_j \rangle \in Arg(CKL R(a, B, \Delta)).$$

Отсюда $Val(CKL R) = \bigcup_i \bigcup_j Val(\langle a_i, b_j \rangle)$,

где $\langle a, b_j \rangle \in Arg(CKL R(A, B, \Delta))$.

На экране временной интервал Δ_i изображается в виде закрашенного поля, длина которого зависит от величины Δ_i . Таким образом, множество $Val(\langle a, b_j \rangle)$ можно отобразить по горизонтали в виде группы полей, считая горизонталь временной осью. Если каждой паре $\langle a, b_j \rangle$ поставить в соответствие горизонталь на экране, то циклограмма $CKL R(a, B, \Delta)$ изображается матрицей окрашенных полей, каждый j -тый горизонтальный ряд которой соответствует множеству $Val(\langle a, b_j \rangle)$.

Формирование экранной циклограммы начинается с определения атрибутов статического и динамического окон. Ширина физического окна на экране $CKL R(a, B, \Delta)$ определения временного масштаба – количество секунд, приходящихся на единицу закрашиваемого поля в зависимости от интервала Δ . Вертикальные параметры окна определяются размером множества $\{ \langle a, b_j \rangle \}$, и в процессе последовательного отображения значений $Val(CKL R(a, B, \Delta))$ могут превычисляться.

Со статическим окном связывается информация, постоянная для циклограммы $CKL R(a, B, \Delta)$: заголовок, подсказка, значение глобального интервала Δ и текстовое поле со значением "a".

После занесения информации в статическое окно активным становится окно динамическое, в которое загружается информация об интервалах для всех $\langle a, b_j \rangle$ и список текстовых полей с именами из

$$\{ b_j | \langle a, b_j \rangle \in Arg(CKL) \}.$$

Динамическое окно циклограммы может быть больше экранного, поэтому с помощью клавиш-стрелок можно управлять окном, высвечивая нужную часть информации.

В целом формирование экранной циклограммы $CKL R(a, B, \Delta)$ состоит из двух этапов:

- инициирование циклограммы, т. е. открытие окон, вычисление масштаба, занесение координатных осей, заголовка, подсказок, отображение аргумента циклограммы в динамическое окно;

- выполнение процедуры *MakeCycle*, отображающей множество $Val(CKL R(a, b, \Delta))$ в динамическое окно.

Далее в цикле, выбирая элементы из A , повторяем процесс формирования экранной циклограммы для каждого элемента из A .

Из вышесказанного следует, что $CKL R(a, B, \Delta)$ отображается на экран в виде последовательности окон с информацией о $CKL R(a_i, B, \Delta)$. Эти окна можно «листать» с помощью клавиш PgUp, PgDn.

4 Экранное редактирование циклограмм

Редактирование циклограмм сводится к редактированию интервальных полей. Выбор поля осуществляется стрелками (влево и вправо), при этом выбранное поле окрашивается другим (активным) цветом. Редактирование выполняется на копии списка $Val(CKL R(a_i, B, \Delta))$, который создается в процессе работы процедуры *MakeCycle*.

В начальный момент процедурой *choose-cycle-field* выбирается последний элемент списка, делается протяжка динамического окна, и поле, окрашенное активным цветом, появляется на экране. При выходе из процедуры *Cycle* этот рабочий список полей копируется в основной массив записей.

С помощью функциональных клавиш с выбранным полем можно проводить следующие операции:

- **F2**: высветить в дополнительном окне значение интервала $[t_0^{(ij)}, t_k^{(ij)}]$ в формате (день, час, мин, с), если поле принадлежит $Val(<a_i, b_j>)$.

- **F5**: изменить длину поля (коррекция значения $t_k^{(ij)}$).

- **P6**: изменить позицию поля (коррекция значений $t_0^{(ij)}$ и $t_k^{(ij)}$).

- **DEL**: удалить интервал.

- **ESC**: выход из режима редактирования интервалов.

Кроме выборки поля по курсору и функциональной клавише **F3** можно вставить в список новое интервальное поле.

5 Процедуры операций

К настоящему времени полный набор операций библиотека не содержит, реализована лишь часть из них. В основном это операции, которые понадобились при создании программ обработки динамических знаний в конкретных приложениях, с которыми приходилось работать автору. Перечислим процедуры операций, которые уже имеются в библиотеке:

- *Cycle_transform* — изменение глобального интервала времени;

- *Cycle_project B* — сужение циклограммы на подмножество в B ;

- *Cycle_inv* — представление циклограммы $CKL R(A, B, \Delta)$ в псевдоинверсной форме для листания циклограммы по элементам множества A .

- Cycle_inverse — транспонирование циклограммы;
 - Cycle_not — инвертирование циклограммы;
 - Cycle_un — агрегирование циклограммы по второму аргументу;
 - Cycle_joint — совмещение на одном экране двух циклограмм $SKL R_1(A, B, \Delta)$ и $SKL R_2(B, A, \Delta)$ с представлением второй в транспонированной форме (эта операция не описана, она нужна как средство ускорения процесса восприятия и осмысливания результатов пользователем);
 - Cycle_subtract — теоретико-множественное вычитание (разность) циклограмм;
 - Cycle_intersect — теоретико-множественное пересечение циклограмм;
 - Cycle_BinUn — теоретико-множественное объединение циклограмм.
- Библиотека процедур для выполнения операций с циклограммами по-полняется в процессе решения прикладных задач.

Заключение

К настоящему времени система генерации программ обработки динамических знаний ещё не достигла того уровня, начиная с которого её можно в полной мере считать инструментарием, ориентированным на конечного пользователя. В частности, в системе отсутствует входной инструментальный язык необходимого уровня, ещё недостаточно обеспечения системы сервисными средствами. В планы авторов входит доведение системы до коммерческого уровня. Однако уже сейчас она может использоваться подготовленным программистом для генерации программ обработки динамических знаний, в результате чего создаются программы на языке Python, которые в дальнейшем можно использовать независимо от системы в других прикладных программах. Такой опыт имеется, и он показал возможность существенного ускорения процессов разработки приложений по обработке динамических знаний и повышения их алгоритмической эффективности благодаря использованию описанного здесь инструмента.

Литература

1. Динамические отношения в задачах обработки знаний / А. С. Гейда, Л. Н. Федорченко, И. В. Афанасьева, Д. С. Хасанов // Вестник Бурятского государственного университета. Математика, информатика. 2021. № 3. С. 39–61. DOI: 10.18101/2304-5728-2021-3-39-61. Текст: непосредственный.

Статья поступила в редакцию 20.05.2022; одобрена после рецензирования 15.06.2022; принята к публикации 08.09.2022.

TOOL FOR DYNAMIC RELATIONS PROCESSING

Ludmila N. Fedorchenko

PhD, Senior Researcher,

St. Petersburg Federal Research Center of the Russian Academy of Sciences

39 14th Line of V.O., St. Petersburg 199178, Russia.

St. Petersburg State University,

7/9 Universitetskaya Emb., St. Petersburg 199034, Russia

Inf@iiias.spb.su

Alexander S. Geida

PhD, Senior Researcher,

St. Petersburg Federal Research Center of the Russian Academy of Sciences

39 14th Line of V.O., St. Petersburg 199178, Russia

geida@iiias.spb.su

Abstract. At present, the introduction of new information technologies (digital) into production and other sectors of the national economy has led to the development of a number of modern areas of research related to the development of artificial intelligence systems, tasks of processing dynamic knowledge, tasks of making decisions in subject areas with dynamic structure. There is a need to develop software tools to support such dynamic knowledge processing technologies, which consists in creating a set of tools with a good user interface for solving various decision-making problems in areas with dynamics. We will assume that the formal description of the dynamics of the structure of the subject area is given by a system with dynamic relations, which are general relations on a set of objects that depend on time.

The article is devoted to the description of the prototype of the software system, which implements the algebra of operations on dynamic relations (cyclograms) and displays them on the screen in a graphical form. The graphical interpretation of relations is more visual and accessible for review than the textual and tabular ones, and the use of a set of operations on relations allows you to quickly comprehend the results and manage the process of solving the problem..

Keywords: dynamic knowledge, dynamic relations, tools, algebra of dynamic relations.

For citation

Fedorchenko L. N., Geida A. S. Tool for Dynamic Relations Processing // Bulletin of Buryat State University. Mathematics, Informatics. 2022. N. 2. P. 102–111.

The article was submitted 20.05.2022; approved after reviewing 15.06.2022; accepted for publication 08.09.2022.