

Agricultural Marketing Service [Electronic resource] / A. Sparger, N. Marathon. – Available at: <http://dx.doi.org/10.9752/TS049.06-2015>.

Каргін А. О., д.т.н., професор,
Іванюк О. І., аспірант (УкрДУЗТ)

ЗАСТОСУВАННЯ ЧАСОВОЇ ЛОГІКИ В АЛГОРИТМАХ ДОСЛІДЖЕННЯ ОТОЧЕННЯ МОБІЛЬНОГО РОБОТА

Мобільні роботи – клас автономних роботів, здатних переміщуватись у просторі на основі методів орієнтування (SLAM) або по наперед заданим маршрутам та виконувати збір сенсорної інформації про оточення з метою моніторингу ситуації. Алгоритми обробки первинної сенсорної інформації для реєстрації та ідентифікації подій базуються на різних моделях. У якості такої моделі, в роботі пропонується використання часової логіки.

Часова (темпоральна) логіка – розширення математичної логіки, що покликане описувати послідовності подій та їх взаємозв'язок на часових шкалах у формалізованому вигляді. Є одним з представників псевдофізичних та модальних логік. Фундамент сучасної часової логіки був закладений Артуром Пріором у 1950-1960-х роках [1], а подальший розвиток вона отримала завдяки працям Аміра Пнуелі, Ханса Кампа [2] та інших спеціалістів у галузі логіки.

Центральними поняттями часової логіки є: подія $e_i \in E$ (E – множина подій); момент часу $t_j \in T$ (T – множина моментів часу); операція τ – співставлення події e_i моменту часу t_j ; операція ρ – відстань (у часі) між різними подіями e_i та e_k ; операції відношення між різними подіями e_i та e_k .

Множина T є упорядкованою та утворює часову шкалу, на якій моменти часу t_j розташовані послідовно.

Якщо події e_i можна поставити у відповідність лише один момент часу t_j ($(\tau(e_i, t_j), t_j \in T' \subseteq T, |T'|=1)$), то така подія називається точковою; якщо моментів часу більше одного ($(\tau(e_i, t_j), t_j \in T' \subseteq T, |T'|>1)$) та вони розташовані на часовій шкалі послідовно, то така подія називається інтервальною, а якщо моменти часу розташовані непослідовно – ланцюговою. Ланцюгову

подію можна вважати комбінацією декількох точкових та/або інтервальних подій.

Інтервальні події можна визначати через дві точкові події (маркери): μ_i^n – початок події e_i та μ_i^k – завершення події e_i .

Розрізняють метричні та топологічні часові шкали.

Метричними називають ті шкали, між моментами часу яких можна визначити відстань у числових одиницях (наприклад, відстань між моментами часу «травень» та «серпень» складає 3 місяці). Розрізняють абсолютні та відносні метричні шкали. Абсолютні шкали мають умовну, але загальноприйнятну точку відліку, відносно якої визначається відстань між моментами часу. У відносних шкалах відстань між моментами часу визначається відносно одна одній.

Топологічні шкали є такими, що їх моментам часу не можна дати ім'я та ввести операцію відстані між ними. Топологічні шкали дозволяють лише відновити послідовність подій у часі відносно одна одній. Окремим випадком топологічних часових шкал є нечіткі шкали, моменти часу яких представляють у вигляді лінгвістичних змінних. Операція τ не реалізується для топологічних шкал, замість неї вводиться операція ϑ , що відповідає нечіткому відношенню моделювання.

Окрім стандартних логічних операцій (інверсія, кон'юнкція, диз'юнкція, імплікація) темпоральна логіка використовує спеціальні часові операції, що дозволяють задавати відношення між різними подіями. В роботах [1] Артура Пріора запропоновано чотири часових оператори: P : «Був такий випадок, що...», F : «Буде такий випадок, що...», G : «Завжди буде так, що...», H : «Завжди було так, що...». Ханс Камп у своїй дисертації [2] увів оператори S : «Since» (з того часу), U : «Until» (до того часу). В [3] професором Поспеловим представлено наступний набір часових операторів: r_1 : «одночасно», r_2 : «бути раніше», r_3 : «бути пізніше», r_4 : «починатися одночасно», r_5 : «закінчуватись одночасно», r_6 : «безпосередньо примикати зліва», r_7 : «перетинатись у часі», r_8 : «співпадати у часі», r_9 : «бути всередині (у часі)». Автор зазначає, що даний набір не є мінімальним, тобто деякі операції можуть бути виражені одна через одну за допомогою стандартних логічних операцій.

В роботі наводиться приклад використання часової логіки для побудови алгоритму моніторингу оточення за допомогою мобільного робота з метою виявлення пожежонебезпечних ситуацій. Робот отримує інформації від трьох сенсорів: температури (t , °C), загазованості (n , мг/м³) та освітленості (E_v , лк). Первинна сенсорна інформація представляється у якості подій нульового рівня (t_i, g_i, l_i), що відповідають приналежності показника певному інтервалу:

$$t \in [0, 10) \rightarrow t_1;$$

$$t \in [10, 20) \rightarrow t_2;$$

$$t \in [20, 30) \rightarrow t_3;$$

$$t \in [30, 40) \rightarrow t_4;$$

$$t \in [40, 50) \rightarrow t_5;$$

$$n \in [0, 100) \rightarrow g_1;$$

$$n \in [100, 220) \rightarrow g_2;$$

$$n \in [220, 600) \rightarrow g_3;$$

$$E_v \in [5000, 10000) \rightarrow l_1;$$

$$E_v \in [10000, 25000) \rightarrow l_2;$$

$$E_v \in [25000, 50000) \rightarrow l_3.$$

На основі подій нульового рівня робиться висновок про наявність/відсутність подій першого

рівня: T – підвищення температури, G – підвищення рівня загазованості, L – підвищення рівня освітленості:

$$t_1 r_2 t_2 r_2 t_3 r_2 t_4 r_2 t_5 \rightarrow T;$$

$$g_1 r_2 g_2 r_2 g_3 \rightarrow G;$$

$$l_1 r_2 l_2 r_2 l_3 \rightarrow L.$$

На основі подій першого рівня робиться висновок про наявність/відсутність події другого рівня: F – наявність загоряння:

$$(T r_1 G r_1 L) \vee (G r_3 (T r_1 L)) \vee (L r_2 (T r_1 G)) \rightarrow F.$$

Список використаних джерел

1. Artur Prior. Time and Modality. Oxford: Oxford University Press. 1957.
2. Hans Kamp. Tense Logic and the Theory of Linear Order. Dissertation. 1968.
3. Поспелов Д.А. Ситуационное управление: теория и практика. – М.: Наука. – Гл. ред. физ.-мат. лит., 1986. – 288 с.

Косолапов А. А., д.т.н.,

Павленко И. А.,

Фролов Н. И. (ДНУЖТ ім. акад. В. Лазаряна)

МОБИЛЬНАЯ ИНТЕРАКТИВНАЯ СИСТЕМА ДЛЯ ОБУЧЕНИЯ И ОПЕРАТИВНОГО ТЕСТИРОВАНИЯ

Эффективность внедрения систем дистанционного обучения в университетах Украины вызывает много сомнений у опытных преподавателей, за плечами которых десятки лет преподавания научно-технических дисциплин и тысячи выпускников, высококвалифицированных специалистов, которые успешно работают на современных предприятиях, в научно-исследовательских институтах, в лабораториях и университетах по всему миру. Действительно, сейчас в ИНТЕРНЕТ есть учебно-методические материалы практически по всем отраслям знаний и дисциплинам и, кажется, находи, бери и учишь виртуально, а университеты не нужны. Скоро выпускникам школ знания по определенной профессии будут считываться и записываться в их память с заранее сформированных специалистами «лент» (см. повесть А. Азимова «Профессия», 1957 г.). Но почему

тогда необходимо учиться 4,5 года, когда в сети всё есть? Почему в лесных дебрях Амазонки и «гималаях» образовательных ресурсов необходим проводник к конечной цели «путешествия»? Потому что основная задача высшего образования – **научить учиться**, в интерактивном общении с преподавателем и в социуме группы обучаемых, научить мыслить креативно, а не компилировать известные решения, важно научить видеть синергетику знаний. Не отрицая преимуществ использования новых информационных технологий в учебном процессе, мы приступили к разработке мобильной интеллектуальной интерактивной системы для обучения студентов с оперативным тестированием знаний.

Техническая структура системы включает компьютер преподавателя (ноутбук, далее НБ) на ОС Windows и личные смартфоны обучаемых под управлением ОС Android. Система организована на клиент-серверной архитектуре с облачным хранилищем данных на основе мощного сервиса Firebase. Это сервис, предоставляющий API для хранения и синхронизации данных. База данных позволяет работать с данными, которые хранятся как JSON, и синхронизируются в реальном времени. JSON (англ. JavaScript Object Notation) – текстовый формат обмена данными, основанный на JavaScript. Взаимодействие с сервисом построено на основе архитектурного стиля REST (от англ. Representational State Transfer – «передача состояния представления») – архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании **распределённой гипермедиа-системы**. В определённых случаях в системах, основанных на данных, это приводит к повышению производительности и упрощению архитектуры. В широком смысле компоненты в REST взаимодействуют наподобие взаимодействия клиентов и серверов в WWW.

Основные функции системы: 1 - подготовка