

The method of transmission of SDL – specifications using high-level Petri nets

Anastasia Zabolotna

Department of Information Systems, Taras Shevchenko Kyiv National University, UKRAINE,
Kiev, 2, avenue Glushkov, Building 6,
E-mail: ZabolotnaA@gmail.com

Analysis, validation and verification of communication protocols – are the current problem of modern programming. Manual modeling often leads to mistakes, so we can face the problem of translation process verification that is difficult task for systems of real complexity. Automatic transmission of specifications into formal models for which there are effective methods of analysis and automatic verification tools, solve this problem.

Natural approach to verification is based on the use of formal models such as finite automata, Petri nets and their generalizations. Herewith, the process of analysis and verification becomes easier.

This research describes the SDL-systems with timers that allow to provide adequately a significant class of communication protocols. The method of transmission into modified colored Petri nets - hierarchical typified time networks which use the concept of time interval proposed by Merlin, priorities and special places which represent token turns, was proposed for them.

Algorithm of transmission of SDL-specifications into the network models of the system SDLE implemented by double-pass transmission.

Modeling method based on the fact that the multilevel description of a system in SDL position of each process exemplar in the general hierarchy of the system remains unchanged, allowing to transmit the system description to the structure of the network, and to model the exemplars of the process using tokens.

As a result, a network model is created which won't contain in any of its sites more than one token which models a process exemplar. Thus, if during the functioning of the system, n of different exemplars of any process can exist, then every place of its modeling network can't contain more than n tokens, moreover each token will correspond its process exemplar. This fact allows to significantly improve the effectivity of modeling, because it substantially reduces the variants surplus of linking of variables.

Переклад виконано Малиновською О. А., центр іноземних мов «Universal Talk», www.utalk.com.ua

Метод трансляції SDL – специфікацій за допомогою мереж Петрі високого рівня

Анастасія Заболотна

Кафедра інформаційних систем, Київський національний університет імені Тараса Шевченка, УКРАЇНА,
м.Київ, просп. Глушкова, 2, корпус 6,
E-mail: ZabolotnaA@gmail.com

Робота присвячена дослідженню проблеми автоматичної побудови мережеских моделей SDL-специфікацій. Мова специфікацій та опису SDL прийнята в якості міжнародного стандарту. Розглядаються SDL-системи з таймерами, засобом збереження сигналу пріоритетами, що дозволяють адекватно представити значний клас комунікаційних протоколів.

В якості моделей вибрані кольорові мережі Петрі, котрі розширюються за допомогою семантики часу та пріоритетів. В роботі описується метод трансляції SDL-систем в дану мережеску модель.

Ключові слова – мережа Петрі, SDL, мережева модель, комунікаційний протокол, екземпляр процесу.

I. Вступ

В останні роки прогрес засобів передачі та обробки інформації призвів до стрімкого проникнення телекомунікаційних технологій в усі сфери життя, завдяки чому побудова та верифікація комунікаційних протоколів стала однією з областей формальної верифікації, що найбільш активно розвиваються. Великі обсяги і високий рівень складності розроблених протоколів вимагали створення засобів автоматичної або напів-автоматичної верифікації розподілених асинхронних систем, в тому числі і комунікаційних протоколів [1,2].

Для представлення розподілених систем часто використовується мова виконуваних специфікацій SDL [1, 2], прийнята в якості стандарту ІТУ. Перевага SDL в її виразності, проте, саме вона і ускладнює аналіз та верифікацію специфікацій цих систем. Один із підходів полягає в автоматичному переводі специфікацій розподілених систем в моделі, для яких розроблені ефективні методи аналізу. В якості моделей вибрані модифіковані кольорові мережі Петрі, названі ієрархічними часовими типізованими мережами (ІЧТ-мережами) [3]. ІЧТ - мережі розширюють безпечні кольорові мережі Петрі за допомогою поняття часу (семантика Мерліна), пріоритетів, а також спеціальних місць, що зображують черги фішок.

II. Коротка характеристика та основні властивості SDL

Графічна мова опису та специфікацій SDL є однією з найбільш відомих, що використовується для формального опису поведінки реактивних та розподілених систем. Вона прийнята в якості міжнародного стандарту (ITU-T) і входить до Рекомендації ІТУ-T серії Z.100. Спочатку вона була націлена на телекомунікаційні системи, проте тепер широко

застосовується і для опису систем керування процесами і систем реального часу.

Мова SDL включає опис як структурної, так і функціональної частини розроблюваної системи. SDL-модель розуміють як набір сполучених узагальнених абстрактних скінченних автоматів.

Архітектура SDL-моделі є багаторівневою. Найбільший загальний об'єкт (найвищий рівень) називається системою (system), все, що знаходиться поза системою, називається оточенням (зовнішнім середовищем, environment) системи та засобами SDL не описується. Система взаємодіє зі своїм оточенням, отримуючи від нього і посылаючи йому сигнали. Система сама по собі є не функціональним, а структурним описом моделі.

Коротко структуру системи можна описати таким чином. Вона складається з одного або декількох блоків (block), з'єднаних між собою і з навколишнім середовищем каналами, по яких передаються сигнали. У свою чергу, блок є іншим, більш низьким, ніж система, рівнем SDL-моделі. На даному рівні система або зовнішній блок (у разі вкладеності блоків) може розглядатися як середовище, поведінка якої спостерігається тільки через вхідні сигнали для поточного блоку. Блок також містить процеси (process), що є вже функціональними компонентами системи в тому сенсі, що саме вони визначають поведінку системи.

Процеси представляють собою нижній рівень SDL-моделі. Структурно процес завжди знаходиться в блоці, який розглядається як середовище для даного процесу. Сигнали між процесами, процесами і блоками передаються маршрутами (signalroute). Маршрути можуть бути з'єднані з каналами верхнього рівня (connect). Отже, процес - це об'єкт, який має кінцевим числом станів, переходами і чергою вхідних сигналів. Перебуваючи в деякому стані, процес витягує черговий сигнал і виконує перехід - здійснює ряд дій, зокрема, визначає свій наступний стан.

Під час роботи системи, описаної в SDL, можуть створюватися і знищуватися екземпляри процесів. Для кожного процесу може створюватися кілька примірників або жодного. Для простоти екземпляри процесів далі будемо називати процесами.

Початковий стан процесу описується конструкцією start, з якого здійснюється початковий перехід в яке-небудь стан процесу (state). Після цього процес починає функціонувати - обробляти сигнали з вхідної черги (згідно дисципліни FIFO) і здійснювати переходи в інші стани.

Кожен перехід складається з послідовності дій SDL-програми - присвоєння, установка і скидання таймера, умовний оператор, створення нового і зупинка поточного процесу, посилка сигналу і перехід в інший стан. Процес може містити оголошення своїх локальних змінних і використовувати значення локальних змінних інших процесів (export, import), а також викликати процедури, які також мають свої статки, переходи та локальні змінні. При необхідності обробити сигнал процедура використовує вхідну чергу процесу, яким вона була викликана, а посилка

сигналу процедурою інтерпретується як посилка цього сигналу цим процесом. Різні процеси можуть звертатися до одних і тих же процедур.

Процеси SDL-системи працюють асинхронно, спілкуючись між собою і середовищем за допомогою сигналів.

Запропонований підхід до верифікації SDL-специфікацій розбитий на два етапи та описаний в наступному розділі.

III. Метод трансляції SDL - специфікацій в мережі Петрі

Алгоритм перекладу SDL-специфікацій в мережеві моделі системи SDLE реалізований методом двохпрохідної трансляції [5, 6].

Мережева модель створюється за допомогою поетапного уточнення. На першому етапі будується мережа, яка розташовується на першій сторінці, відповідає основній структурі системи і містить по одному переходу для кожного блоку. Кожний канал, пов'язаний з блоком, представляється у мережі одним або двома місцями - чергами, залежно від того, був він одно- або двонаправленим. Фішки в отриманих місцях можуть приймати значення з безлічі кольорів, які визначаються сигналами, що передаються по відповідних каналах. Спочатку всі місця, породжені за описами каналів, мають нульову розмітку. З'єднання переходів та місць здійснюється дугами, напрямком яких співпадала з напрямком передачі повідомлень [4, 5].

На другому етапі здійснюється трансляція блоку, яка відбувається таким же чином, що і трансляція всієї системи в цілому. Переходи, побудовані на першому етапі, замінюються підмережами, які відповідають розбиттю блоку і розташовуються на пов'язаній з цим переходом підсторінці. При трансляції блоку, що складається з підблоків і внутрішніх каналів, кожному підблоку в підмережі відповідає один перехід, кожному внутрішньому каналу - одне або два місця-черги, залежно від того, одно- чи двонаправлений канал. Трансляція блоку, що складається з процесів, відбувається аналогічно трансляції блоку будь-якого рівня ієрархії. Кожному екземпляру процесу відповідає один перехід, кожному маршруту - одне або два місця-черги, залежно від того, який маршрут - одно- або двонаправлений [4, 5].

Транслятор функціонує наступним чином. Модуль аналізатора обробляє текстовий файл, що містить SDL-специфікацію, тобто здійснює лексичну згортку та синтаксичний розбір і будує внутрішнє представлення специфікації. У разі відсутності помилок запускається модуль генерації мережевої моделі. Спочатку будується внутрішнє представлення ІЧТ-мережі, а потім здійснюється її візуалізація в системі SDLE.

Ієрархічна мережева модель (ІЧТ-мережа) - це композиція безлічі неієрархічних мереж, званих сторінками. Сторінки можуть містити вершини спеціального типу, які називаються модулями і

з'єднуються з місцями на сторінці за тим же принципом, що і переходи.

Модуль представляє підмережа, що розташовується на окремій сторінці, яка в свою чергу може містити модулі. Така сторінка називається підсторінкою сторінки, на якій розташовується модуль. Підсторінка містить копії всіх місць, з якими пов'язаний модуль. Місце-копія може бути вхідним місцем для деякого переходу або модуля на підсторінці тоді і тільки тоді, коли його прототип є вхідним місцем для модуля, що представляє підсторінку. Аналогічно, тільки копія вихідного місця-прототипу може бути вихідним місцем деякого переходу або модуля на підсторінці.

Поведінка ієрархічної мережі визначається поведінково еквівалентної їй неієрархічною мережею, яка отримана в результаті заміщення всіх модулів сторінками, які вони представляють. При цьому кожен модуль разом зі своїми дугами видаляється зі сторінки, а на його місце заноситься підмережа, що розташовувалася на підсторінці. З'єднання мереж відбувається за місцями: кожне місце-прототип склеюється з усіма своїми копіями. Побудова внутрішнього представлення мережі проводиться за кроками, що відповідають етапам в описі алгоритму трансляції. В першу чергу створюється кореневий рівень ієрархічної мережі, що складається з модулів системи SDLE, що представляють блоки SDL-специфікації. Також на цьому етапі створюються місця-черги, які моделюють канали. Ці модулі та місця з'єднуються дугами згідно з описом SDL-системи. Наступні чотири кроки генерації виконуються послідовно для кожного блоку.

На другому проході алгоритм генерує мережу, яка реалізує блок SDL-специфікації. Ця мережа, у свою чергу, містить модулі, що відповідають процесам, і місцям-чергам, що моделюють маршрути сигналів. Ці модулі та місця з'єднуються дугами згідно з описом блоку.

Далі генерується мережа, що реалізує процес SDL-специфікації. Вона містить модулі, що відповідають SDL-переходам процесу, місцям, що моделюють змінні та лічильники, і деяким службовим місцям. На цьому етапі побудови мережі дуги не створюються, а добудовуються на наступному кроці, оскільки не можна заздалегідь вказати, на яких переходах використовується змінна або таймер.

На наступному кроці трансляції SDL-переходів створюється підмережа, в якій реалізується логіка SDL-переходу. В процесі побудови цієї мережі створюються дуги для мережі третього рівня.

На завершальному етапі будуються модулі, що реалізують процедури і дії з таймерами, якщо такі є.

При побудові мережі в системі SDLE використовуються засоби створення ієрархічних мережеских моделей, що надаються цією системою. Після того як мережа створена, здійснюється розміщення її елементів на площині, тобто кожному елементу приписуються координати на відповідних сторінках системи SDLE. При цьому фрагменти мережі, які реалізують дії SDL-переходів, розміщуються типовим чином.

IV. Моделювання системи та блоку

Кожному каналу в мережі буде відповідати місце, тип якого є list. Будь-яка фішка в такому місці буде представлятися списком, кожен елемент списку - записом, значення першого поля запису є особистий ідентифікатор екземпляра процесу-одержувача, другого - особистий ідентифікатор екземпляра процесу-відправника, інші поля будуть відповідати сигналу в черзі до екземпляру процесу-одержувача. Оскільки в КМП черга до екземпляра процесу представляється однією фішкою, то місце, що моделює порт процесу, буде вхідним і вихідним для кожного переходу в мережі, що поміщає сигнал в чергу чи вилучає сигнал з черги [3].

Для моделювання особистих ідентифікаторів динамічно створюваних екземплярів процесів, на першому етапі має бути створено службове місце, назовемо його Place_id, яке буде містити одну фішку. Значення цієї фішки є ціле число - особистий ідентифікатор процесу, створюваного динамічно під час функціонування мережі.

Початкова розмітка цього місця - фішка значення $n + 1$, де n - кількість екземплярів процесів, створених в момент ініціалізації системи. Це місце буде вхідним і вихідним місцем для всіх переходів мережі (N-переходів), які моделюють блоки. Блоки містять як процеси, що виконують запит на створення іншого процесу, так і породжувані процеси.

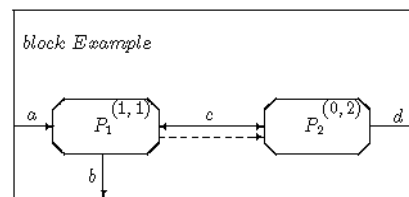


Рис.1 Опис блоку Example.

На наступних етапах здійснюється побудова мереж для блоків, також відповідно до алгоритму для SDL-специфікацій без динамічних конструкцій. Переходи, побудовані на попередніх етапах, замінюються підмережами, які будуть відповідати розподіленню блоків і розташовуватися на окремих підсторінках. Службове місце Place_id буде вхідним і вихідним місцем для всіх N-переходів, моделюючих підблоки, які містять процеси, виконують запит на створення іншого процесу, і породжують процеси. На етапі трансляції блоку, що складається з процесів, місце Place_id також буде вхідним і вихідним місцем для всіх N-переходів, що моделюють процеси, виконують запит на створення іншого процесу, і породжувані процеси.

Крім того, на цьому етапі трансляції для кожного N-переходу, який моделює процес, що містить оператор create, буде створено службове місце, назовемо його Create_place, що є вихідним місцем для цього N-переходу і вхідним для N-переходу, що моделює породжуваний процес, вказаний в операторі create. Фішки в місці Create_place можуть приймати значення з безлічі кольорів Create_place_id, який визначається наступним чином:

$\text{Create_place_id} = \text{product Int} \cdot \text{Int}$.

Значення першого поля фішки в місці Create_place є особистий ідентифікатор створюваного оператором create примірника процесу, значення другого поля - особистий ідентифікатор "екземпляра-батька".

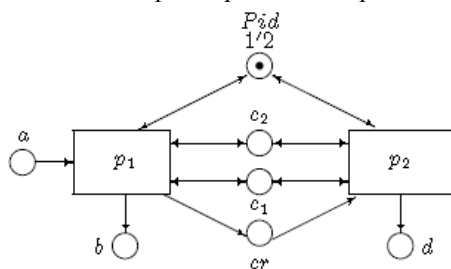


Рис.2 Мережа для блоку Example.

Для наведеного на рис. 1 опису блоку Example відповідна мережа показана на рис. 2, службове місце Create_place є вихідним для переходу p_1 і вхідним для p_2 , моделюючими процеси P_1 і P_2 відповідно. Двонаправлений канал c , що з'єднує блоки P_1 і P_2 , в мережі представляється місцями c_1 і c_2 . Місце Place_id буде вхідним і вихідним місцем для переходів p_1 і p_2 .

V. Система SDLE

Система SDLE являє собою інтегрований програмний комплекс для проектування, аналізу та симуляції моделей ПЧТ-мереж, що включає:

- транслятор з мови SDL;
- симулятор - блок імітаційного моделювання.

Цикл роботи користувача в системі SDLE виглядає наступним чином. Будується вихідна мережева модель досліджуваної системи, або ця мережева модель виходить яким-небудь іншим способом, наприклад, як результат трансляції з мови здійснених специфікацій SDL. Потім проводиться симуляція моделі в автоматичному режимі. Система дозволяє візуально контролювати хід симуляції, відстежувати аналіз структурних властивостей поетапно. За результатами симуляції вихідна модель уточнюється, і цикл розробки повторюється до отримання задовільних результатів.

Симулятор дозволяє простежити за процесом функціонування мережі при заданій початковій розмітці. При цьому часто вдається виявити семантичні помилки моделі, які проявляються у неочікуваній поведінці мережі.

Висновок

У роботі описана процедура трансляції SDL-специфікацій з динамічними конструкціями в кольорові мережі Йенсена, збагачені пріоритетами.

Розширення цих мереж пріоритетами дозволяє розвинути засоби симуляції та аналізу.

Спосіб моделювання заснований на тому, що в багаторівневому описі системи в SDL позиція кожного примірника процесу в загальній ієрархії системи залишається незмінною, що дозволяє опис системи транслувати в структуру мережі, а екземпляри процесу моделювати за допомогою фішок.

У результаті роботи алгоритму створюється така мережева модель, в якій в кожному місці буде містити не більше однієї фішки, що моделює деякий екземпляр процесу. Таким чином, якщо під час функціонування системи може існувати n різних екземплярів будь-якого процесу, то в кожному місці моделюючої його мережі може містити не більше n фішок, причому кожна фішка буде відповідати своєму екземпляру процесу. Це факт дозволяє істотно підвищити ефективність моделювання, тому що істотно зменшує перебір варіантів зв'язування змінних.

Література

- [1] Specification and Description Language (SDL). Recommendation Z.100. — ITU-T, 2000.
- [2] Jensen K., Christensen S., Wells L. Colored Petri Nets and CPN Tools for Modeling and Validation of Concurrent Systems // Intern. J. on Software Tools for Technology Transfer/ 2007/ Vol. 9. P. 213-254.
- [3] Jensen K. Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer - Verlag, 1997. Vol.1-3
- [4] Заболотна А.С. Трансляція SDL – специфікацій в модифіковані мережі Петрі. І Міжнародна наукова конференція студентів, аспірантів та молодих вчених «Теоретичні та прикладні аспекти кібернетики» (ТААС'2011). Київський національний університет імені Тараса Шевченка. Факультет кібернетики. Київ, 21-25 лютого 2011, ст. 85-87.
- [5] Заболотна А.С. Моделювання динамічних конструкцій SDL – специфікацій за допомогою мереж Петрі. Міжнародна наукова дисциплінарна конференція студентів, аспірантів та молодих вчених «Шевченківська весна 2011». Київський національний університет імені Тараса Шевченка. Факультет кібернетики. Київ, 21-25 березня 2011
- [6] Заболотна А.С. Моделювання та верифікація комунікаційних протоколів, представлених на мові SDL, за допомогою мереж Петрі високого рівня. XVII International Conference "Problems of decision making under uncertainties" – PDMU – 2011. Східниця, Україна, 23-27 травня 2011.