

С. А. АББАС, А. И. ВОДЯХО  
Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина), Санкт-Петербург

Н. А. ЖУКОВА  
Санкт-Петербургский институт информатики и автоматизации Российской Академии Наук, Санкт-Петербург

М. А. ЧЕРВОНЦЕВ  
Научно-инженерный центр СПбГЭТУ «ЛЭТИ», Санкт-Петербург

## ТИПОВЫЕ ЗАДАЧИ СБОРА ДАННЫХ В ГЕТЕРОГЕННЫХ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННО-ОРИЕНТИРОВАННЫХ СИСТЕМАХ И ВОЗМОЖНЫЕ ВАРИАНТЫ ИХ РЕШЕНИЯ С ИСПОЛЬЗОВАНИЕМ МОДЕЛЬНОГО ПОДХОДА

*Аннотация.* В докладе рассматриваются возможные подходы к построению систем сбора данных в многоуровневых распределенных системах. Делается вывод о возможности использования модельного подхода. Анализируются требования, предъявляемые к системе моделей. Рассматриваются возможные подходы к построению моделей. Предлагается система моделей, описывающих структуру и поведение наблюдаемой системы в терминах четырех графов: графа ресурсов, графа потока данных, графа потока управления и графа потока запросов, которые могут быть автоматически построены на основе информации, поступающей от наблюдаемой системы в форме лог-файлов. Предлагается виртуальная машина, реализующая данный подход.

**Введение.** Быстрое развитие техники и технологий, в частности, нанотехнологий, телекоммуникаций, программной инженерии позволяет создавать за приемлемые деньги и сроки системы принципиально более высокого уровня сложности по сравнению с существующими системами. При этом увеличивается не только сложность систем, но и повышается уровень их интеллекта за счет увеличения доли информационной составляющей, что позволяет относить эти системы к классу информационно-ориентированных систем (ИОС) (software intensive systems) [1]. В большинстве своем, современные ИОС представляют собой многоуровневые гетерогенные системы, состоящие из большого числа элементов разной физической природы, таких как механические, электромеханические, природные системы, а также людей [2]. Такие ИОС способны реализовывать сложное поведение и в перспективе способны к обучению и самообучению.

В современных ИОС решаются самые разнообразные задачи, в частности, задачи управления, сбора данных для последующего их анализа и многие другие. Эффективное решение всех этих задач требует реализации процедур сбора данных как о наблюдаемых системах (НС), так и о самой системе сбора данных (ССД). В этом контексте задача сбора данных становится достаточно актуальной и нетривиальной, поскольку наблюдатель (Н) не имеет полной информации о структуре НС и протекающих в ней бизнес-процессах (БП).

**Процесс сбора данных.** В самом общем виде процесс сбора данных (ПСД) может быть описан как  $ПСД = \langle Н, НС, ССД \rangle$ , где  $Н$  – наблюдатели,  $НС$  – наблюдаемая система,  $ССД$  – система сбора данных. В качестве наблюдателя могут выступать как люди (операторы, сервисный персонал, аналитики, менеджеры разных уровней, так и подсистемы формирования управляющих воздействий.  $Н$  имеют свои интересы в получении данных, информации, знаний о  $НС$ .  $Н$  общаются с  $ССД$  на доменно-ориентированном языке (Domain Specific Language, DSL) [3]. Обычно число DSL определяются числом групп  $Н$ . DSL может формулировать запросы в терминах сырых данных, информации или знаний в зависимости от роли  $Н$ .

$НС$  можно представить как  $НС = \langle СТР, ФНК, РС, ПОРТ \rangle$ , где  $СТР$  – структура  $НС$ ,  $ФНК$  – реализуемая функциональность,  $РС$  – ресурсы,  $ПОРТ$  – порты доступа к  $НС$  со стороны  $ССД$ .  $СТР$  можно представить как  $СТР = \langle \{сmp\}, ПТС \rangle$ , где  $\{сmp\}$  – множество возможных структур  $НС$ , а  $ПТС$  – правила трансформации структур. Аналогично можно определить функционал, реализуемый  $НС$   $ФНК = \langle \{bn\}, ПТБП \rangle$ , где  $\{bn\}$  – множество реализуемых БП,  $ПТБП$  правила трансформации БП (здесь речь идет о статике, для представления хода реализации БП требуется использовать разметку). Реализуемую функциональность можно описать также в терминах сервисов  $ФНК = \{СЕРВ\}$ , где  $\{СЕРВ\}$  – множество реализуемых сервисов. Ресурсы описываются аналогичным образом  $РС = \langle \{pc\}, ПР \rangle$ , где  $\{pc\}$  – множество элементарных ресурсов,

а *ПР* – правила реконфигурации. Следует заметить, что ресурсы могут быть как физическими, так и виртуальным. В последнем случае они могут рассматриваться как сервисы. ПОРТ можно определить как множество пар *точка доступа – операции*  $\text{ПОРТ} = \{\text{ТД}, \text{ОП}\}$ , однако ПОРТ удобнее определить как сервис. Описанная выше модель является одноуровневой. Реально чаще всего приходится иметь дело с иерархическими системами.

В таком контексте ПСД можно определить следующим образом. Н на собственном DSL обращается к ССД с запросом о состоянии НС, о состоянии отдельных элементов, о ходе БП. получает ответ на том же DSL. Запросы могут относиться к настоящему, прошлому или будущему моменту времени.

**Предлагаемый подход к организации процесса сбора данных.** Можно выделить 2 основных альтернативных подхода к построению ССД: прямой сбор данных и сбор данных с использованием промежуточной модели. В первом случае ССД располагает минимальным знанием о НС и при поступлении запроса Н необходимая информация собирается с НС, местом хранения данных, информации и знаний о НС является сама НС. Это подход, используемый в настоящее время чаще всего, и он достаточно хорошо работает в случае, когда в качестве НС используется, например, корпоративная информационная система с достаточно стабильной структурой. Однако, для современных интеллектуальных систем с постоянно изменяющейся структурой и поведением использование прямого подхода становится проблематичным, в частности, по причине больших задержек.

Альтернативой прямому подходу является модельный подход, который предполагает наличие в ССД явной модели НС. При этом запросы Н направляются к модели НС, т. е. процесс сбора данных разбивается на 2 параллельных подпроцесса: процесс построения и поддержания модели в актуальном состоянии и процесс обработки запросов Н. Очевидно, что в крупных распределенных системах модель НС тоже является распределенной.

Функционирование модельно-ориентированной ССД может быть описано в терминах 4х асинхронно работающих виртуальных машин: процессора моделей, процессора генерации скриптов, интерпретатора скриптов и процессора обработки запросов Н и формирования представлений, которые работают с репозитарием моделей. Процессор моделей отвечает за формирование моделей, их модификацию и обработку запросов к моделям. Процессор формирования скриптов на основании запроса Н и модели формирует скрипт, который реализует процедуру сбора данных. Интерпретатор скриптов выполняют скрипты. Процессор обработки запросов и формирования представлений обрабатывает запросы Н.

Основными преимуществами модельного подхода по сравнению с прямым подходом являются: возможность уменьшить время отклика на запрос Н, отслеживать динамику структуры и поведения НС, состоящей из элементов разной физической природы, включая людей (киберфизические, социофизические системы) [2], если НС может работать с разными моделями, возможность оперативно получать данные о прошлых состояниях и в определенных пределах предсказывать поведение НС, что важно, в частности, для реализации проактивного управления и реализации таких механизмов как самодиагностика, самовосстановление и т. п.

Понятно, что прямой сбор и чисто модельный подход представляют собой крайние случаи. В крупных распределенных системах реально использование только распределенных ССД.

**Представление моделей НС.** Это один из ключевых вопросов, поскольку к моделям НС предъявляются жесткие требования, основными из которых являются следующие: модель должна адекватно требуемым уровням детализации описывать текущее, прошлое и будущее состояния НС в терминах ее структуры и поведения, должна иметься возможность автоматического построения и коррекции модели по поступающей в форме логов информации о событиях в НС, модель должна обеспечивать меньшее время отклика на запросы пользователей по сравнению с прямым сбором данных. Наиболее жестким следует считать требование автоматического построения структурных моделей и моделей поведения. При анализе кандидатов на использование в качестве модели НС рассматривались следующие известные подходы: акторный подход [4], G-графы [5] и модель, основанная на стратегиях управления [6]. Была выбрана последняя модель. Применительно к ССД эта модель может быть интерпретирована следующим образом:  $\text{МНС} = \langle \text{РГ}, \text{АУРГ}, \text{ГПД}, \text{ГПУ}, \text{ГПЗ}, \text{АРБП} \rangle$ , где *МНС* – модель НС, *РГ* – ресурсный граф, *АУРГ* – автомат управления ресурсным графом, *ГПД* – граф потока данных, *ГПУ* – граф потока управления, *ГПЗ* – граф потока запросов, *АРБП* – автомат реструктуриза-

ции БП. РГ описывает текущую структуру, АУРГ определяет возможности реструктуризации. БП описываются в терминах графа ГПД, отражающего зависимости по данным, ГПУ описывает порядок выполнения операторов, а ГПЗ описывает решаемую задачу в терминах запросов на выполнение операторов. В [7] рассматриваются вопросы автоматического построения многоуровневых структурных моделей НС. В рамках концепции Process Mining [8] решается задача автоматического построения ГПУ и частично ГПД.

Данная модель может быть реализована на разных платформах, в частности, в виде объектно-ориентированного представления на языке высокого уровня, на онтологической платформе, с использованием графов знаний. Анализ показывает, что наибольшее быстродействие можно получить при реализации МНС на языках высокого уровня, наиболее простые решения можно получить при реализации МНС в форме онтологий.

**Проблемы реализации модельного подхода.** На концептуальном уровне можно выделить 3 альтернативных подхода к построению ССД. В первом случае ССД строится как элемент системы управления, при этом число  $N$  минимально, основным  $N$  является подсистема формирования управляющих воздействий, а основные требования связаны с минимальным временем отклика. При этом сама ССД может быть встроена, например, в оборудование. Во втором случае ССД строится как обычная информационно-управляющая система, основным назначением которой является сбор данных и представление их в разных форматах разным группам  $N$  в соответствии с их ролями. В этом случае основная нагрузка обычно падает на подсистему обработки запросов  $N$  и формирование представлений. В третьем случае ССД строится как контекстно-ориентированная система [9], в которой имеется отдельная модель, описывающая текущий контекст.

Наибольший практический интерес с точки зрения архитектуры представляют сервисно-ориентированные архитектуры и, в первую очередь, микросервисы. Если ССД не ориентированы на работу в реальном времени, то особых проблем не возникает. Однако, если НС работает даже в режиме нежесткого реального времени, то возникают проблемы с быстродействием.

В самом общем виде сервисное представление  $СОП = \langle \{срв\}, \{скр\} \rangle$ , где  $\{срв\}$  – множество сервисов, а  $\{скр\}$  – множество скриптов, выполняющих операции над сервисами. Можно выделить две основные группы сервисов: инфраструктурные сервисы (сервисы нижнего уровня) и сервисы верхнего уровня. Скрипты представляют собой программы на DSL или языке общего назначения, которые могут строиться как в статике, так и в динамике.

Одна из потенциально возможных областей применения СДД, использующих модельный подход – это КФС [2], построенные на базе туманных платформ [10], использующих трехуровневую структуру. На нижнем уровне находятся сети датчиков, на среднем – контроллеры, а на верхнем – облачные сервера. На нижнем уровне размещаются инфраструктурные сервисы, реализующие предварительную обработку информации. Основная обработка реализуется на среднем уровне, где располагаются основные сервисы. На верхнем уровне располагаются модели верхнего уровня и сервисы, требующие наличие мощных вычислителей (аналитика, синтез моделей).

**Заключение.** Предлагаемый подход является достаточно общим, и в большинстве случаев нет необходимости использовать его в полном объеме. Основной сферой применения следует считать сложные распределенные КФС с постоянно изменяющейся структурой и динамическими бизнес-процессами. Несомненным достоинством предлагаемого подхода является возможность строить НС нового уровня сложности и возможность реализовывать процедуры сбора данных в КФС с высоким уровнем интеллекта. Основными проблемами, связанными с практическим применением предлагаемого модельного подхода являются достаточно высокая сложность и относительно большое время отклика.

#### ЛИТЕРАТУРА

1. International Standard ISO/IEC/IEEE 42010 Systems and software engineering — Architecture description [Электронный ресурс]. — Электрон. дан. — Режим доступа: <http://www.iso.org>
2. Sanfelice R.G. Analysis and Design of Cyber-Physical Systems. A Hybrid Control Systems Approach // Cyber-Physical Systems: From Theory to Practice / D. Rawat, J. Rodrigues, I. Stojmenovic. CRC Press, 2016. ISBN 978-1-4822-6333-6.
3. Fowler M. Domain-Specific Languages: Addison-Wesley. Upper-Saddle River. NJ 2014. 583 p.
4. Agha G.A. Actors: A Model of Concurrent Computation in Distributed Systems: MIT. MA 1986, 142 p.
5. **Охтилев М.Ю., Соколов Б.В., Юсупов Р.М.** Интеллектуальные технологии мониторинга и управления структурной динамикой сложных технических объектов. М.: Наука, 2006. 410 с.

6. Функционально-ориентированные процессоры / Водяхо А.И., Смоллов В.Б., Плюсин В.У., Пузанков Д.В.; /под ред. В.Б. Смоллова. Л.: Машиностроение. Ленингр. отд. 1988. 224 с.
7. Distributed Technical Object Model Synthesis Based on Monitoring Data / V.Yu. Osipov, A.I. Vodyaho, N.A. Zhukova, M. Tianxing, S. Lebedev // International Journal of Knowledge and Systems Science (IJKSS). 2019. Vol. 10, Iss. 3. Art. 3. P. 27–43.
8. Van der Aalst W. Process Mining Data Science in Action. Second Edition. Heidelberg. Springer, 2016. 468 p
9. Loke S. Context-Aware Pervasive Systems. Architectures for a New Breed of Applications. Taylor & Francis Group, Boca Raton, FL, 2007. 220 p.
10. IEEE Standard Association. FOG – Fog Computing and Networking Architecture Framework, [Online] <http://standards.ieee.org/develop/wg/FOG.html> (дата обращения 17.11.2019).

S.A. Abbas, A.I. Vodyaho, (St. Petersburg Electrotechnical University «LETI», St. Petersburg);  
N.A. Zhukova, (St. Petersburg Institute for Informatics and Automation of the Russian Academy of  
Science, St. Petersburg); M.A. Chervontsev, (Research and Engineering Center of Saint Petersburg  
Electro-Technical University, St. Petersburg)

### **Typical Data Collection Problems in Heterogeneous Distributed Software Intensive Systems and Possible Solutions Based on Model Approach**

The report discusses possible approaches to building data collection systems in multi-level distributed systems. The conclusion is made about the possibility of using a model approach. The requirements for the system of model are analyzed, possible approaches to building models are considered. The model which describes the structure and the behavior of the observed system is proposed. The model describes the observed system in terms of 4 graphs: a resource graph, a data flow graph, a control flow graph, and a query flow graph, which can be automatically constructed on the base of information received from the observed system in the form of log files. A virtual machine that implements this approach is proposed.