ДИНАМИЧЕСКАЯ ВЕРИФИКАЦИЯ БАЗ ЗНАНИЙ

П.В. Титенков
Кафедра интеллектуальных информационных технологий
Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: pavel.titenkov@gmail.com

В работе рассматривается роль процесса верификации в рамках жизненного цикла интеллектуальных систем, методы и принципы динамической верификации и валидации баз знаний интеллектуальных систем, их особенности, преимущества динамической верификации относительно статической, специфика ее применения, возможные проблемы внедрения данного механизма в процесс работы базы знаний интеллектуальной системы.

Введение

Одним из основных показателей любого хранилища данных является показатель качества, который демонстрирует насколько корректна и актуальна информация, хранимая в базе. Объемы хранилищ данных неуклонно растут и вместе с ними растут и риски появления ошибок, дублирования, противоречий и прочих неточностей в хранимой информации. Всвязи с этим сегодня наблюдается огромный интерес к вопросам верификации и валидации информации.

I. Роль верификации баз знаний

Верификация представляет собой процесс исследования и обоснования того, что данные корректны и соответствуют установленным правилам, изложенным в некоторой спецификации. Существует ряд приложений, в которых ошибки не критичны. Они приводят к легким моральным травмам, к возможным сбоям, требующим восстановление системы. Такие ошибки обычно очевидны и их можно быстро исправить. Однако с ростом систем, с увеличением количества пользователей, с увеличением количества предоставляемых функций, цена каждой такой ошибки вырастает в десятки раз [1]. Эти ошибки теряют свою очевидность, становятся критическими для системы и заставляют разработчиков переделывать целые модули для их устранения. Таким образом, цена ошибки и роль верификации растут вместе с системой и напрямую зависят от масштабов ее использования.

Верификация наиболее важна в системах с повышенными требованиями к надежности (Safety-critical systems). В таких системах ошибки приводят к гибели людей, крупным финансовым потерям, ущербу окружающей среде и так далее [2]. Например, недостаточно протестированный модуль запуска ракеты Ariane-5 привел к тому, что в июне 1996 года ракета взорвалась спустя 40 секунд после старта. Ущерб тогда был оценен в 7 млрд. долларов, а причиной являлась ошибка при округлении 64-битного float в 16-битный int [3]. Данную трагедию можно было бы избежать, если бы система была проверифицирована.

Любая экспертная система является системой с повышенными требованиями к надежности, так как априори должна выдавать пользователю корректную информацию [4]. Предпалагалось, что строиться экспертные системы будут на основе эксклюзивных знаний экспертов, описывающих узкую проблемную область, модель которой будет создаваться в результате совместной работы инженера по знаниям с экспертом. Однако в практике создания интеллектуальных экспертных систем выяснилось, что объемы знаний, полученных на этапе структурирования баз знаний, с одной стороны, могут быть весьма значительными, а с другой стороны - включать недостоверные знания, содержащие отдельные виды ошибок.

Наличие большого количества ошибок в базе знаний значительно ухудшает качество интеллектуальной системы в целом, что может приводить к частичной или даже полной её неработоспособности. В результате этого, возникает необходимость в постоянной верификации баз знаний интеллектуальных систем на протяжении всего их жизненного цикла.

Разработка интеллектуальных систем является распределенным и параллельным процессом, поэтому человеку трудно оценить корректность функционирования каждой из взаимодействующих компонент системы. Ошибки в базах знаний могут возникать из-за неправильного понимания предметной области разработчиком, несогласованных параллельных процессов разработки одного фрагмента базы знаний и т.д. Из-за вышеприведенных причин, наиболее распространенный метод проверки правильности программных систем – ручное тестирование – окажется трудоемким и неэффективным для большого объема информации. С ростом базы знаний эффективность ручного тестирования будет стремительно снижаться до тех пор, пока не станет абсолютно бесполезным. Поэтому в качестве основного метода повышения качества разраба-

тываемых баз знаний следует применять динамическую верификацию.

II. Динамическая верификация

Динамические методы верификации используют результаты реальной работы проверяемой программной системы или ее прототипов, чтобы проверять соответствие этих результатов требованиям и проектным решениям [5].

Существует два основных вида динамических методов верификации:

- мониторинг, во время которого идет только наблюдение, запись и оценка результатов работы хранилища данных при его обычном использовании;
- тестирование, при котором проверяемое хранилище данных используется в рамках заранее подготовленных сценариев. В данном случае результаты работы тоже записываются, анализируются и оцениваются.

Основное отличие тестирования от мониторинга — целенаправленные попытки создать определенные ситуации (сценарии тестирования), чтобы проверить реакцию хранилища данных на них [6]. Как видно, разделение мониторинга и тестирования несколько условно, тестирование всегда включает в себя и мониторинг. Общим для этих методов верификации является создание контролируемой среды выполнения, обеспечивающей измерение различных характеристик базы знаний, а также оценка этих результатов и характеристик.

Динамическую верификацию, служащую для обнаружения наличия ошибок и оценки качества базы знаний, следует отличать от отладки, основная задача которой — определение точного местоположения и исправление ошибок. Однако в ходе разработки динамическая верификация часто используется как часть отладки, и поэтому, помимо самого факта наличия ошибок, должна давать как можно более детальную информацию об их локализации и нарушаемых ими ограничениях, чтобы облегчить разработчикам их поиск.

Основное достоинство динамических методов верификации — возможность получить информацию о реальной работе базы знаний и о реальных показателях ее функциональности, производительности, надежности или переносимости в режиме реального времени не останавливая и не прерывая работу интеллектуальной системы.

При динамической верификации функциональности основное внимание уделяется протоколированию результатов работы операций, доступных элементов состояния компонентов системы, содержимого сообщений, которыми обмениваются компоненты системы, а также действительного порядка событий, насколько это позволяет сделать архитектура базы знаний интеллектуальной системы. Также могут протоколироваться временные интервалы между отдельными событиями.

При верификационном мониторинге поведения интеллектуальной системы в ходе ее обычной работы протоколируется и оценивается его соответствие требованиям [7]. Частный случай мониторинга — профилирование, при котором обычно измеряются показатели производительности, однако довольно часто можно встретить употребление термина «профилирование» как для мониторинга, включающего контроль операций с памятью и взаимодействие параллельных потоков и процессов в системе.

Техники и инструменты мониторинга различаются по видам протоколируемой ими информации, способу получения данных о работе хранилища данных и способу получения оценок характеристик хранилища данных [8].

Протоколируемая информация зависит от оцениваемых характеристик качества и от других целей проводимой верификации. Чаще всего фиксируются общие данные и метрики, факты вызовов операций, содержимое сообщений передаваемых между системой и окружением или между компонентами системы, время работы определенных процедур, использование ресурсов, значения различных внутренних параметров [9].

Естественно, методы динамической верификации не могут обеспечить всей полноты проверки базы знаний на корректность, но могут существенно снизить затраты на исправление ошибок, что в конечном итоге приведёт к улучшению качества всей интеллектуальной системы в целом.

1. Agrawal, A. Reusable Idioms and Patterns in Graph Transformation Languages / A. Agrawal, A. Vizhanyo, Z. Kalmar, F. Shi, A. Narayanan, G. Karsai // Journal Electronic Notes in Theoretical Computer Science (ENTCS). – № 127. –2005. – P. 181 – 192.
2. Карпов Ю.Г. Model Cheking. Верификация параллельных и распределенных программных систем. - СПб.: БХВ-Петербург, 2010. - 560 с.
3. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. Учебник / Гаврилова Т.А. [и др.]; - СПб. : Изд-во «Питер», 2001.
4. Richard, J. RapidIO: технология для приложений реального времени / J. Richard // Interconnect Strategies – 2007. – №4. – С. 40–43.
5. Рыбина Г.В., Смирнов В.С. Верификация баз знаний в интегрированных экспертных системах // Новости искусственного интеллекта. 2005. №3. С. 7-19.
6. Storey, N. Safety Critical Systems / N. Storey. – NY.: Addison Wesley, –1996. – 453 p.
7. Adrion W., Branstad M., Cherniavsky J. (1982). Validation, verification, and testing of computer software. ACM Computing Surveys, 14(2), 159-192.
8. Bareiss E., Porter B. Wier C. (1988). Protos: An exemplar-based learning apprentice. In Proceedings of the Fourth International Workshop on Machine Learning (pp. 12-23). University of California at Irvine, June 1987.
9. Bench-Capon T.J.M., The Role of Ontologies in the Verication and Validation of Knowledge Based Systems, August 28, 1988, Vienna, Austria, Ed. R.R. Wagner, IEEE Computer Society, 1998.