

Ответы на вопросы

В чем преимущество семантического моделирования?

Семантическое моделирование – это новый подход в разработке, управлении и сопровождении сложных автономных систем (САС), который позволяет значительно снизить стоимость владения такой системой. Это достигается за счет исполнимых спецификаций, когда логика работы разнородных частей системы выносится на уровень человека, специалиста в предметной области.

Семантическое моделирование позволяет человеку, не вдаваясь в детали, как работает САС внутри, управлять ее поведением.

Исполнимые спецификации являются:

1. Языком, который понятен даже не программисту, и в тоже время задает правила работы САС (или робота, ИИ, или правила организации сложных процессов)
2. Знанием, которое содержит наиболее важные аспекты функционирования САС, и которое может быть отчуждено от конкретной системы.

Семантическое моделирование опирается на теорию гибридных систем. Это означает, что исполнимые спецификации могут соединять вместе разнородные компоненты: например, искусственный интеллект, блокчейн, интернет вещей, роботов. И каждый компонент может быть реализован на любом языке программирования с использованием любых современных подходов.

Преимущество семантического подхода лежит не внутри сложной автономной системы, а снаружи – там, где возникает человек, который с ней взаимодействует. Семантическое моделирование значительно снижает стоимость владения САС, поскольку:

1. Формирует знание о работе САС на понятном человеку языке.
2. Позволяет делиться этим знанием и расширять его. Снижает стоимость доработок и изменения логики спецификаций.
3. Позволяет оптимальным способом находить «наилучшие» спецификации, так как, с одной стороны, может использовать человеческие знания, а с другой – может быть автоматически проверена в системах моделирования с цифровым двойником.

В чем отличие новой парадигмы семантического моделирования от традиционной парадигмы разработки информационных систем?

Традиционная парадигма разработки информационных систем автоматизации основана на алгоритмизации, которая требует анализа и расчленения изначальных требований на алгоритмические модули (куски) (см рис. ниже). При этом различные модули могут быть реализованы на основе различных архитектурных подходов (процедурный, функциональный, ООП, event driven, конечные автоматы, case management, BPM, и тд.). Таким образом, **программисты** разделяют начальные логические требования и критерии на куски и реализуют в разных модулях в виде алгоритмов. В результате возникает следующая проблема: как гарантировать, что в процессе декомпозиции логических требований на алгоритмические модули:

1. не были утеряны части этих требований;

2. не появились некие новые ограничения, изначально не присутствовавшие в постановке задачи.

Другими словами, насколько полученная алгоритмическая система соответствует начальным требованиям? Эта проблема особенно усугубляется для систем с длительным жизненным циклом, поскольку знания о семантике (о том, как логические требования/критерии и данные из предметной области транслируются в структуры данных и алгоритмы внутри ИС), которые есть у разработчиков системы, не могут быть отчуждены в автоматически обрабатываемую форму. В силу того, что разработчики приходят в проект и уходят, с течением времени часть этих знаний начинает теряться и искажаться. Это очевидным образом приводит к повышению стоимости сопровождения системы. В частности, при смене разработчиков, новые разработчики вынуждены тратить время на то, чтобы восстановить эти знания о семантике из исходного кода, вместо того, чтобы сразу реализовывать новые требования. Затраты на такое восстановление знаний часто превосходят затраты на программирование. Даже хорошая документация не может содержать полную картину о семантике и со временем в ней начинают накапливаться несоответствия реально работающей системе.

Традиционная парадигма



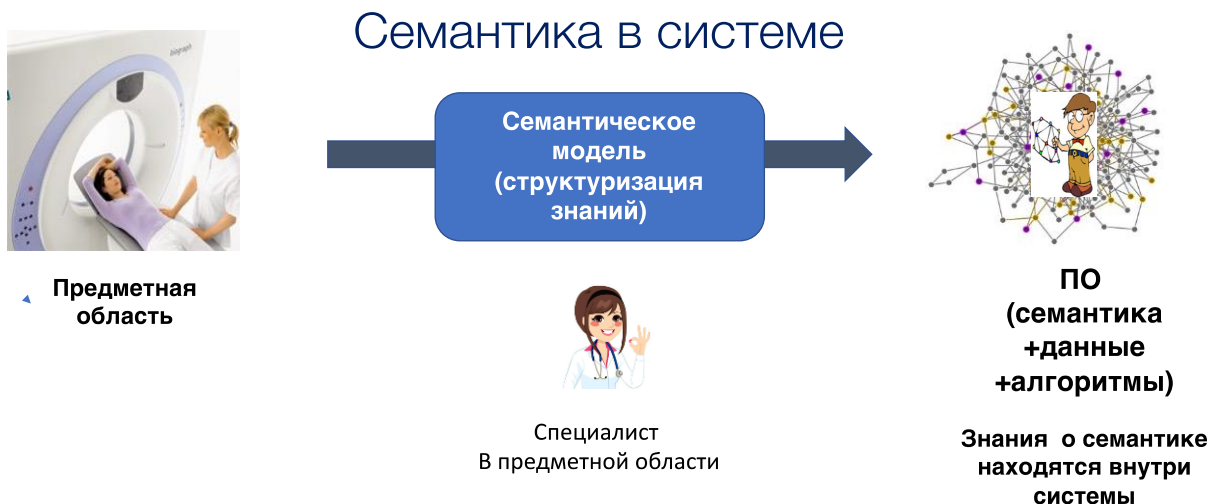
Гарантировать соответствие требований полученной системе при таком подходе невозможно в силу фундаментальных ограничений, доказанных в математике (одно из следствий теоремы Гёделя о неполноте говорит, что невозможно создать автоматические средства предсказания поведения алгоритмов).

Новая парадигма, основанная на семантическом моделировании, предполагает другой подход к проектированию ИТ системы. **Специалисты в предметной области** или **аналитики** сами без привлечения программистов строят спецификации на логическом языке, которые не требуют ручного программирования и могут непосредственно исполняться. Это позволяет:

1. Целостность: контролировать логическую целостность и непротиворечивость начальным требованиям в течение жизненного цикла ИТ системы
2. Исполнимые спецификации: отчуждать знания специалистов
3. Исполнимые спецификации: работать с этими знаниями в автоматическом режиме

4. Самообучение: автоматически извлекать новые знания из потоков данных и дополнять ими систему
5. Рефлексия: автоматически анализировать поведение ИС и модифицировать поведение в соответствии с требованиями

Семантическая парадигма



Что значит, что семантическое моделирование – это задачно-ориентированная технология?

Это означает применение следующих принципов при создании систем на основе семантического моделирования:

1. От задачи -- построение ИС отталкиваясь от формализации задач, для которых эта ИС предназначена
2. Ничего лишнего – решение задач по мере поступления.
3. Гибридность – интеграция с различными существующими информационными системами, основанными на совершенно различных технологических подходах (машинное обучение и нейросети, событийные системы, транзакционные системы, BPM и конечные автоматы и т.п.) для максимизации их использования.
4. Целостность – знания о предметной области необходимые для решения целевой задачи, их семантике, в т.ч. логические критерии и ограничения должны быть в явном виде, доступном для автоматического исполнения
5. Иерархичность смыслов: на каждом уровне смыслы семантических моделей должны быть понятны соответствующим специалистам в предметной области без дополнительной документации.

Например, пусть строится система управления базовыми станциями мобильного оператора состоящая из таких уровней как уровня диагностики и управления аварийными ситуациями, уровня управления питанием, уровня управления климатом. В этом случае семантические (логические) модели соответствующих уровней должны быть понятны соответствующим специалистам службы эксплуатации – связистам, энергетикам.

6. Имитационное моделирование – возможность выявления сложных аспектов поведения системы путем создания и проверки поведения информационных двойников.

Чем отличается семантическое моделирование от Adaptive Case Management (ACM), Business Process Management (BPM), Complex Event Processing (CEP), ...?

Семантическое моделирование -- это технология создания гибридных информационных систем и управления их жизненным циклом, позволяющая интегрировать информационные подсистемы, основанные на различных технологиях и подходах в единую с логической точки зрения систему. Семантические модели создаются на универсальном логическом языке. Это означает, что его можно использовать для создания систем на основе любых формализуемых подходов таких как конечные автоматы, событийные модели, нечеткие логики, темпоральные логики, которые лежат в основе таких технологий как BPM, ACM, CEP, ML. Однако с прагматической точки зрения часто разумнее использовать специализированные технологические решения в рамках гибридных семантических моделей. Другими словами, если уже есть подходящая и проверенная платформа для CEP, то проще интегрировать ее с верхнеуровневой логикой задаваемой семантическими моделями.

Гибридность



С другой стороны, если в контексте создаваемой ИС требуется в каком-то ее аспекте использовать для примера event driven модель, и при этом нет подходящего узкоспециализированного решения для этой цели, то такой модуль можно формализовать и реализовать на основе семантической моделирования.