

# ПОСТРОЕНИЕ РАСПРЕДЕЛЕННЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ НА ОСНОВЕ РАСПРЕДЕЛЕННОЙ ФРЕЙМОВОЙ ИЕРАРХИИ

Сошников Д.В.

Московский авиационный институт (технический университет)  
E-mail: dmitri@soshnikov.com, WWW: <http://www.soshnikov.com>

**Abstract:** This paper outlines the principle for constructing distributed intelligent systems using production-frame knowledge representation based on the frame hierarchy being distributed over the network. This approach uses homogeneous knowledge representation, and allows viewing individual sub-hierarchies as self-documenting ontologies, thus contrasting with typical agent architectures. Java-based JULIA toolkit has been developed to support the proposed model.

В последние годы наблюдается стремительный рост инфраструктуры компьютерных сетей, как общемирового масштаба (Интернет), так и масштаба предприятия. Расширение технической базы требует соответствующего развития программного обеспечения для адаптации его к сетевой среде, а также разработки методологических подходов к организации процесса распределенной деятельности. Развитие теории интеллектуальных распределенных систем представляет особый интерес для создания эффективных средств представления, структурирования и поиска информации (знаний) в Интернет, а также с точки зрения создания систем искусственного интеллекта следующего поколения по принципу "перехода количества в качество".

В области распределенного искусственного интеллекта получила распространение **агентная архитектура**, в которой взаимодействие сообщества агентов строится в соответствии с некоторым внешним общим языком (как правило KQML/KIF) и эксплицитно заданной структурой знаний предметной области (**онтологией**). Однако в целом классе задач **распределенного накопления и использования знаний** представляется разумным использовать унифицированное представление знаний во всех узлах системы и строить взаимодействие узлов на основе обмена знаниями во внутреннем представлении. В этом случае интерес представляет обмен знаниями на различных уровнях: как статическими знаниями о решаемой задаче (что традиционно соответствует обмену сообщениями между агентами на языках запросов KQML), так и динамическими правилами (соответствует обмену знаниями на языке представления знаний KIF).

Автором предлагается использовать **продукционно-фреймовое представление знаний**, так как такое представление позволяет естественным образом комбинировать фреймовые суб-иерархии, а также обеспечивает кластеризацию динамических знаний вокруг фреймов в форме соответствующих процедур-демонов (соответствующих продукционным правилам прямого вывода) и процедур-запросов (для правил обратного вывода). Фреймовые суб-иерархии в такой модели могут быть произвольным образом распределены по сети, образуя единую **распределенную фреймовую иерархию** (см. рис. 1).

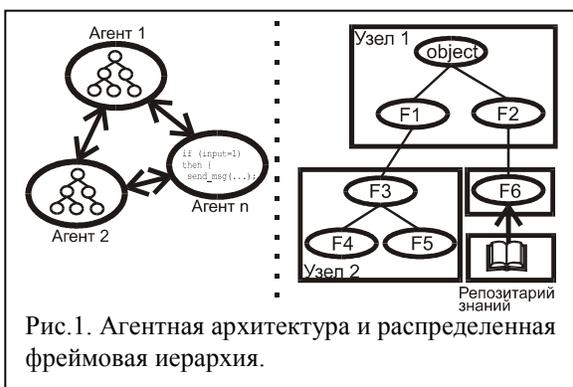


Рис.1. Агентная архитектура и распределенная фреймовая иерархия.

Основным способом использования распределенных знаний является **удаленное наследование**. В этом случае при обратном выводе, если исчерпаны все локальные правила для некоторого слота, происходит удаленный вызов, и процесс логического вывода продолжается на удаленном узле, в случае необходимости запрашивая посредством обратного вызова (callback) значения слотов базового

фрейма. В случае прямого вывода при присваивании значения некоторому слоту процесс активизации демонов включает в себя удаленный вызов соответствующих процедур во всех родительских фреймовых иерархиях. Использование множественного или **псевдомножественного наследования** (при котором единственный родитель некоторого фрейма изменяется динамически в процессе вывода в соответствии с некоторым внешним алгоритмом, внутренними правилами или мета-правилами) позволяет комбинировать знания из нескольких источников. Предложенный метод **индивидуально-слотового наследования** позволяет еще более гибко оперировать распределенными знаниями.

В описанной выше стратегии вывода обмен знаниями по сети происходит на уровне статических знаний, что характерно для многоагентных систем и позволяет классифицировать предложенную архитектуру как **делиберативную статическую агентную архитектуру**. Однако, в отличие от агентной архитектуры, структура знаний во всех узлах системы унифицирована, а ее таксономический характер позволяет использовать базу знаний в родительской суб-иерархии как **авто-онтологию** для удаленного взаимодействия (при наличии соответствующих средств аннотирования и доступа к метаданным структуры иерархии).

Для обмена динамическими знаниями предлагается использовать **мобильное наследование**, при котором по сети передаются необходимые правила во внутреннем представлении, а логический вывод проводится на исходном узле. Также предлагается возможность использования **отложенной загрузки правил** и **репозитариев знаний**, в которых структурирование знаний может быть не строго таксономично, а представление - более наглядно (например, основанное на XML).

Описанные архитектурные решения легли в основу инструментария JULIA (Java Universal Library for Intelligent Applications), реализованного автором на языке Java. Инструментарий является открытым, легковесным (объем необходимого для исполнения runtime-окружения составляет около 100 Кб), допускает гибкую интеграцию с реляционными базами данных и императивным кодом на языках высокого уровня в компонентной модели. В качестве протокола удаленного взаимодействия используется CORBA. Более подробную информацию можно найти на сайте <http://www.soshnikov.com/julia>, или в публикациях автора (например, **arXiv:cs.AI/0106054** на <http://www.arXiv.org>).