

## 5. МЕТОДОЛОГИЯ ПОСТРОЕНИЯ ЭС

### 5.1. Этапы построения ЭС

Процесс разработки ЭС можно условно разбить на пять этапов: идентификация, концептуализация, формализация, реализация и тестирование.

**На этапе идентификации** решаются следующие задачи: определяются участники процесса проектирования и их роли, идентифицируется проблема, определяются ресурсы и цели. Задача определения участников и их ролей сводится к определению количества экспертов и инженеров по знаниям, а также формы их взаимоотношений (например, эксперт может выступать либо в роли учителя, либо в роли информирующего). Определяются источники знаний: книги, инструкции и т.п.

**Идентификация проблемы** заключается в составлении неформального (вербального) описания решаемой проблемы. В этом описании указываются: общие характеристики проблемы; подпроблемы, выделяемые внутри данной проблемы; ключевые понятия и отношения; входные данные; предположительный вид решения; знания, релевантные решаемой проблеме.

При проектировании ЭС типичными **ресурсами** являются: источники знаний, время разработки, вычислительные средства, объем финансирования. Задача определения ресурсов является весьма важной, поскольку ограниченность какого-либо ресурса существенно влияет на процесс проектирования. Так, например, при недостаточном финансировании предпочтение может быть отдано не разработке оригинальной новой системы, а адаптации существующей.

**Задача идентификации целей** заключается в формулировании в явном виде целей построения ЭС. При этом важно отличать цели, ради которых строится система, от задач, которые она должна решать. Примеры возможных целей: формализация неформальных знаний эксперта; улучшение качества решений, принимаемых экспертом; автоматизация рутинных аспектов работы эксперта (пользователя); повышения компетентности; размножение редко встречающихся способностей и навыков.

На первом этапе разработки инженер по знаниям должен ответить на основной вопрос: "Подходят ли методы инженерии знаний для решения предложенной проблемы?"

**На этапе концептуализации** инженер по знаниям и эксперт решают: какие понятия, отношения и механизмы управления нужны для описания решения задач в избранной области. На этом этапе определяются следующие особенности проблемы: типы доступных знаний, выводимые данные, подпроблемы общей проблемы, используемые стратегии и гипотезы, виды взаимосвязей между объектами области, типы используемых отношений (иерархия, причина\следствие, часть\целое и т.п.), типы ограничений, накладываемых на процесс решений, состав знаний, используемых для получения и обоснования решения. Опыт показывает, что для определения перечисленных характеристик проблемы целесообразно составить детальный протокол действий и рассуждений эксперта в процессе решения хотя бы одной конкретной задачи. Такой протокол обеспечивает инженера по знаниям словарем терминов и некоторым приблизительным представлением о тех стратегиях, которые использует эксперт. Разработчикам не следует пытаться осуществить полный анализ задачи до начала реализации программы. Они многому научатся после первой реализации, которая будет формировать и направлять процесс концептуализации.

**Процесс формализации** состоит в описании ключевых понятий, отношений, подзадач и характеристик информационного потока, выделенных на этапе концептуализации, некоторым формальным способом, обычно в рамках схемы, диктуемой языком - инструментальным средством построения ЭС или моделью представления знаний. Инженер по знаниям определяет, подходят ли имеющиеся инструментальные средства для решения рассматриваемой проблемы или необходимы оригинальные разработки. Выходом этапа формализации является описание процесса решения рассматриваемой проблемы на выбранном формальном языке.

**На этапе реализации** разработчик превращает формальные знания в работающую компьютерную программу: создается один или несколько прототипов ЭС, решающих требуемые задачи. Разработка прототипа состоит в программировании его компонент или выборе их из имеющихся инструментальных средств и наполнение базы знаний. Создание первого прототипа должно: 1) подтвердить, что выбранные методы решений и способы представления пригодны для успешного решения, по крайней мере, ряда задач из области экспертизы; 2) показать, что с увеличением объема знаний и улучшением стратегий поиска экспертная система сможет дать высококачественные и эффективные решения всех задач данной проблемной области. В первом прототипе реализуется простейшая процедура вывода. При его разработке основная цель состоит в том, чтобы получить решение задачи, не заботясь пока о его эффективности. И только после завершения разработки первого прототипа начинаются исследования, направленные на повышение эффективности работы системы.

**Тестирование** включает оценивание качества работы и полезности программы-прототипа и ее пересмотр, если это необходимо. Эксперт обычно оценивает прототип и помогает разработчику его пересмотреть. Как только прототип испытан на нескольких примерах, его следует протестировать на многих задачах, чтобы оценить качество его работы и его полезность. Оценивание качества работы системы-прототипа состоит в ответе на следующие вопросы:

1. Принимает ли система решения, которые обычно признаются экспертами как правильные?
2. Являются ли правила вывода безошибочными, непротиворечивыми и полными?
3. Позволяет ли системе ее стратегия управления, рассматривать обстоятельства в естественном порядке, который предпочитает эксперт? Опрашивание пользователя в неестественном или нелепо выглядящем порядке быстро подрывает его доверие к системе.
4. Адекватно ли объяснения системы поясняют, как и почему были получены те или иные заключения?
5. Охватывают ли тестовые задачи всю запланированную область компетентности системы?

При оценивании полезности системы возникает другой набор вопросов:

1. Помогает ли решение задачи пользователю сколько-нибудь существенным образом?
2. Являются ли заключения системы разумно организованными, упорядоченными и изложенными с нужной степенью подробности?
3. Удовлетворяет ли пользователя скорость работы системы?
4. Является ли интерфейс достаточно "дружественным"?

## 5.2. Уровни разработки ЭС

**Демонстрационный прототип.** Система решает часть поставленной задачи, указывая тем самым, что подход применим и разработка системы осуществима. Представляет собой небольшую программу, которая используется для двух целей: 1) убедить потенциальных кредиторов или финансирующие организации, что технология искусственного интеллекта и экспертных систем может быть эффективно использована для решения рассматриваемых задач; 2) проверить проектные концепции относительно определения задачи, ее постановки и представления знаний в данной области. Для разработки демонстрационного прототипа требуется в среднем примерно 3-6 недель. Типичный демонстрационный прототип системы, основанный на правилах, может содержать от 50 до 150 правил, удовлетворительно работать в одном или двух пробных случаях.

**Исследовательский прототип** - программа среднего размера, решает все требуемые задачи, но неустойчива в работе и не полностью проверена. Такие системы чаще всего: 1) оказываются легко уязвимыми: они могут потерпеть полный провал, когда им предлагают задачу, лежащую вблизи границы, разделяющей те задачи, которые они не могут; 2) могут ошибаться при решении задачи, лежащей в пределах их области компетентности из-за того, что они не были достаточно полно протестированы. Типичный исследовательский прототип системы, основанный на правилах, может содержать от 300 до 800 правил, хорошо работать на большом числе пробных случаев и потребовать на свою разработку 3 -6 месяцев.

**Действующий прототип** надежно решает все задачи, но для решения сложных задач может потребовать чрезмерно много времени и памяти. Размер этих программ варьирует от среднего до большого, они подвергались пересмотру в результате опробования на реальных задачах сообществом пользователей, они умеренно надежны, имеют удобные, дружественные интерфейсы и ориентированы на нужды конечного пользователя. Типичный действующий прототип системы, основанной на правилах, может содержать от 500 до 1500 правил, очень хорошо работать на многих пробных случаях и потребовать на свою разработку порядка года.

**Промышленная система** обеспечивает высокое качество, надежность, быстроедействие и эффективность работы в реальных условиях эксплуатации. Это большие программы, подвергнутые интенсивной проверке в условиях опытной эксплуатации и, вероятно, переписанные на более эффективном языке обычно, на языке более низкого уровня с целью увеличения скорости их работы и уменьшения необходимой машинной памяти. Типичная промышленная система, основанная на правилах, может содержать от 1000 до 2000 правил, обеспечивать точное, быстрое и эффективное принятие решений и потребовать на свою разработку 1-1.5 года.

**Коммерческая система** - это быстро и легко адаптируемая система для продажи другим потребителям. Содержит от 1000 до 3000 и более правил. Доведение ЭС до стадии коммерческой системы может потребовать до 2-х лет.

### 5.3. Принципы построения ЭС

**Принцип быстрого построения прототипа системы.** Этот принцип обусловлен тем, что: 1) затраты на построение ЭС большие; 2) сроки построения, обычно, длительные; 3) область применения может быть выбрана неудачно; 4) исходных данных для построения мало и они дополняются в ходе разработки и эксплуатации системы. Поэтому этот принцип предполагает, что получить уверенность в возможности и полезности разработки необходимо с наименьшими затратами, т.к. априорно чисто аналитически это выполнить практически невозможно.

**Принцип постепенного развития системы.** Развитие ЭС обычно осуществляется переходом от простых задач к трудным посредством постепенного совершенствования организации и представления знаний системы. Постепенное наращивание означает, что: 1) система сама помогает себя развивать, т.е. как только разработчики приобретут достаточно знаний, чтобы построить хотя бы очень простую систему, они это делают и используют обратную связь от работающей модели прототипа, чтобы направить и сконцентрировать свои усилия; 2) разработчики могут воспользоваться опытом, который они приобретают в процессе реализации системы.

### 5.4. Рекомендации по построению ЭС

Ниже приводятся рекомендации, которые отражают соображения или эвристики, выработанные проектировщиками относительно того, как лучше действовать при разработке ЭС.

1. Сосредоточьтесь на узкой специализированной области, в которой не используется много знаний, относящихся к категории здравого смысла. Реализация рассуждений на уровне здравого смысла является достаточно сложной задачей. И построить систему, компетентную в нескольких проблемных областях, чрезвычайно трудно, так как, скорее всего, потребуется использовать различные парадигмы и формализмы.

2. Выберите задачу, которая является и не слишком легкой, и не слишком трудной для экспертов. Исходя из современных методов, "слишком легкую" задачу можно определить как "решаемую за несколько минут", а "слишком трудную" - как "требующую несколько месяцев".

3. Определите задачу предельно ясно. На выходе этого этапа необходимо описать природу входных и выходных данных как можно точнее, а эксперту следует определить наиболее важные концепции и связи. Необходимо также иметь доступ к конкретным примерам задач и их решений.

4. Существенно сотрудничество с экспертом, умеющим четко формулировать свои выводы. На сегодняшний день инженерия знаний занимается процессом перевода человеческой компетентности в форму, с которой может работать ЭВМ, и это осуществляется при долгосрочном сотрудничестве с экспертом.

5. Ознакомьтесь с проблемой, прежде чем начать напряженную работу с экспертом. Прочитайте соответствующие отчеты и поговорите с другими экспертами, чтобы узнать как можно больше о предметной области. В противном случае процессы идентификации проблемы и выявления основных концепций и связей будут крайне тяжелыми.

6. Четко определите и охарактеризуйте важные стороны проблемы. В ходе первого тура интенсивных бесед с экспертами следует выяснить роли участников процесса приобретения знаний, сформулировать исследуемую проблему, установить имеющиеся для выполнения проекта ресурсы и охарактеризовать цели или задачи создаваемой ЭС.

7. Составьте детальный протокол процесса решения экспертом хотя бы одной типовой задачи. Именно протокол поможет ответить на многие вопросы. На его основе можно будет составить список используемых терминов и наметить стратегии.

8. Выберите метод или проблемно-независимую ЭС, которые минимизируют несоответствие в представлениях подзадач. Различные языки или средства, используемые для построения ЭС, могут быть наиболее подходящими для различных подзадач. Возьмите то средство, которое обладает наименьшими недостатками в целом для всех рассматриваемых проблем.

9. Начните создавать прототипную версию ЭС сразу же, как только первый пример хорошо понят. Типичная ошибка - выжидать тот момент, когда база знаний станет полной, и только после этого приступать к программированию. Следует же сделать начальный вариант, используя простую форму представления и простую структуру управления. Поработайте с ними как с опытными образцами для определения того, какие новые знания необходимо добавить. Следует запланировать получение работающей версии через несколько месяцев после начала проекта, а не через годы.

10. Интенсивно работайте с основным набором представительных задач. Возьмите пять-шесть основных проблем, включая и входные данные, и желаемые выходные данные. Выберите несколько простых проблем, относящихся к различным аспектам задачи, чтобы полученная система не была слишком

узко специализирована. Получите правила для этих проблем прежде, чем приступить к другим проблемам. Эти основные проблемы могут быть использованы как контрольные вешки на этапе реализации.

11. Стремитесь к простоте дедуктивной машины в прототипной версии. Простые механизмы логического вывода легче строить, они же позволяют быстро начинать эксперименты с правилами. Кроме того, цель инженерии знаний - сделать как можно больше знаний доступными. Знания же, внесенные в сложный механизм логического вывода, труднодоступны.

12. На первых порах не беспокойтесь об эффективности системы в смысле быстрогодействия и требований к памяти. Главной заботой должно стать построение базы знаний и дедуктивной машины, которые решили бы поставленную задачу. Достигнув этого, можно начать беспокоиться и об эффективности.

13. Подберите или разработайте средства, помогающие в процессе написания правил. Например, огромную ценность может иметь редактор правил.

14. Уделяйте внимание документации. Даже первая версия нуждается в описании на одной-двух страницах для того, чтобы коллеги могли самостоятельно работать с программой.

15. Приступайте к построению системы, не дожидаясь, пока неформальные правила приобретут идеальный вид. Это - другой способ сказать, что следует начинать разрабатывать систему как можно раньше. Это утверждение повторено, потому что очень часто поступают наоборот. Помните, что в любом случае не возлагать большие надежды на то, что первые неформальные правила будут сразу же совершенно точными, а стремление практически их реализовать является мощным средством для совершенствования правил в нужных направлениях.

16. Создайте библиотеку обработанных системой случаев, примеров. После каждого набора модификаций необходимо проверить работу системы на всех примерах, хранящихся в библиотеке, для того, чтобы посмотреть, все ли дефекты устранены и не появились ли новые. После того, как первая версия стала функционировать, автоматизируйте хранение, обработку и использование библиотечных примеров.

17. Не просто разговаривайте с экспертом, а наблюдайте за тем, как он или она решает задачи. Отдельные детали экспертизы не всегда бывают очевидными при абстрактном обсуждении задачи или предметной области. Действительные правила эксперта легче распознать, когда он или она решают конкретные проблемы, а примеры позволяют задавать специфические вопросы о том, почему некоторое действие было или, напротив, не было осуществлено.

18. Поставьте перед экспертом задачу спроектировать полезное изделие. Неизбежны трудности и задержки. Если деятельность по построению системы была описана как интеллектуальный вызов, то эксперт будет терпелив и полезен, чем в случае, если у него возникнут иллюзионные представления о том, что все получится очень просто.

19. Сделайте ввод и вывод ЭС выглядящими естественно для пользователей. Стилизованный псевдоестественный язык легко понимать, а также легко анализировать и генерировать. Вид, в котором ответы выводятся на дисплей, должен быть близок к стандартным формам из учебной литературы.

20. Как можно раньше задумайтесь над тем, как вы собираетесь оценить успехи вашей деятельности. Будут ли собираться официальные отзывы? Будут ли требоваться правильные решения для новых проблем? Ожидается ли от системы качество работы, заслуживающее Нобелевской премии?

21. Как можно раньше задумайтесь относительно того, каким образом эксперт мог бы оценить работоспособность системы. Зависит ли этот успех от получения решения данной проблемы? Если нет, то задача недостаточно ясна или эксперт имеет ложные ожидания. Эксперт может ожидать не только правильного ответа, но и правильного способа получения ответа.

22. Интерфейс с пользователем играет важную роль для принятия системы. Система, которая решает даже сверхсложные задачи, может быть отвергнута, если она слишком сложна для обращения, либо слишком хрупка для неопытного пользователя. Небольшие эргономические ухищрения могут определить, станет ли вся разработка успехом или неудачей.

23. Обеспечивайте избыточность. Избыточные данные, структуры гипотез, правила вывода могут оказаться полезными для решения проблем, связанных с ошибочной или отсутствующей информацией. Ни одно заключение не должно основываться только на одном источнике подтверждения.

## Контрольные вопросы и задания

1. Дайте характеристику этапам построения ЭС.
2. Что является источниками знаний? Кто такой эксперт?
3. Чем вызвана многоуровневость разработки ЭС?
4. Зачем необходимо быстрое построение прототипа системы?
5. Когда необходимо начинать создание прототипной версии ЭС и почему?