

18. Инструментальные средства проектирования ИС.

Ссылки на материалы:

1. http://www.mstu.edu.ru/study/materials/zelenkov/ch_5_2.html
2. <http://citforum.ru/database/kbd96/42.shtml>
3. <http://wiki.zunder.ru/doku.php/gos:79>

CASE (англ. Computer-Aided Software Engineering) — набор инструментов и методов программной инженерии для проектирования программного обеспечения, который помогает обеспечить высокое качество программ, отсутствие ошибок и простоту в обслуживании программных продуктов. Также под CASE понимают совокупность методов и средств проектирования информационных систем с использованием CASE-инструментов. Все CASE-средства делятся на типы, категории и уровни. Отметим, что многие кейс средства предназначены не только для проектирования данных, но и для решения других задач, например моделирования потоков данных или бизнес-процессов, функционального моделирования, прототипирования приложений, их документирования, управления проектами и т.д.

Классификация по типам отражает функциональную ориентацию CASE-средств в технологическом процессе:

1. Анализ и проектирование. Средства данной группы используются для создания спецификаций системы и ее проектирования. Они поддерживают широко известные методологии проектирования. К таким средствам относятся:

- CASE.Аналитик (Эйтэкс), The Developer (ASYST Technologies), POSE (Computer Systems Advisers), ProKit*Workbench (McDonnell Douglas);
- Excelerator (Index Technology), Design-Aid (Nastec), Design Machine (Optima), MicroStep (Mela Systems), vsDesigner (Visual Software), Analist/Designer (Yourdon);
- Design/IDEF (Meta Software), BPWin (Logic Works), SELECT (Select Software Tools), System Architect (Popkin Software & Systems);
- Westmoint I-CASE Yourdon (Westmount Technology B.V. & CADRE Technologies), CASE/4/0 (microTOOL GmbH).

Их целью является определение системных требований и свойств, которыми система должна обладать, а также создание проекта системы, удовлетворяющей этим требованиям и обладающей соответствующими свойствами. На выходе продуцируются спецификации компонент системы и интерфейсов, связывающих эти компоненты, а также "калька" архитектуры системы и детальная "калька" проекта, включающая алгоритмы и определения структур данных.

2. Проектирование баз данных и файлов. Средства данной группы обеспечивают логическое моделирование данных, автоматическое преобразование моделей данных в Третью Нормальную Форму, автоматическую генерацию схем БД и описаний форматов файлов на уровне программного кода:

ERWin (Logic Works), Chen Toolkit (Chen & Associates), S-Designer (SDP), Designer2000 (Oracle), Silvernm (Computer Systems Advisers).

3. Программирование. Средства этой группы поддерживают этапы программирования и тестирования, автоматическую кодогенерацию из спецификаций, получая полностью документированную выполняемую программу: COBOL2/Workbench (Mikro Focus), DECASE (DEC), NETRON/CAP (Netron), APS (Sage Software).

Помимо диаграммеров различного назначения и средств поддержки работы с репозиторием, в эту группу средств включены и традиционные генераторы кодов,

анализаторы кодов (как в статике, так и в динамике), генераторы наборов тестов, анализаторы покрытия тестами, отладчики.

4. Сопровождение и реинжиниринг. К таким средствам относятся документаторы, анализаторы программ, средства реструктурирования и реинжиниринга: Adpac CASE Tools (Adpac), Scan/COBOL u Superstructure (Computer Data Systems), Inspector/Recoder (Language Technology).

Их целью является корректировка, изменение, анализ, преобразование и реинжиниринг существующей системы. Средства позволяют осуществлять поддержку всей системной документации, включая коды, спецификации, наборы тестов; контролировать покрытие тестами для оценки полноты тестируемости; управлять функционированием системы и т.п. Особый интерес представляют средства обеспечения мобильности (в CASE они получили название средств миграции) и реинжиниринга. К средствам миграции относятся трансляторы, конверторы, макрогенераторы и др., позволяющие обеспечить перенос существующей системы в новое операционное или аппаратное окружение.

Средства реинжиниринга включают:

- статические анализаторы для продуцирования схем системы ПО из ее кодов, оценки влияния модификаций (например «эффекта ряби» - внесение изменений с целью исправления ошибок порождает новые ошибки);
- динамические анализаторы (обычно компиляторы и интерпретаторы с встроенными отладочными возможностями);
- документаторы, позволяющие автоматически получать обновленную документацию при изменении кода;
- редакторы кодов, автоматически изменяющие при редактировании и все предшествующие коду структуры (например, спецификации);
- средства доступа к спецификациям, их модификации и генерации нового (модифицированного) кода;
- средства реверсного инжиниринга, транслирующие коды в спецификации.

5. Окружение. Средства поддержки платформ для интеграции, создания и придания товарного вида CASE-средствам: Miilti/Cam (AGS Management Systems), Design/OA (Mela Software).

6. Управление проектом. Средства, поддерживающие планирование, контроль, руководство, взаимодействие, иными словами, функции, необходимые в процессе разработки и сопровождения проектов: Open Plan Professional (WelcomSoftware), Project Workbench (Applied Business Technology).

7. Средства документирования: SoDA – Software Document Automation – автоматизированное документирование (фирма Rational Software).

Классификация по категориям определяет уровень интегрированности по выполняемым функциям и включает вспомогательные программы (tools), пакеты разработчика (toolkit) и инструментальные средства (workbench). Категория tool обозначает вспомогательный пакет, решающий небольшую автономную задачу, принадлежащую проблеме более широкого масштаба. Категория toolkit представляет совокупность интегрированных программных средств, обеспечивающих помощь для одного из классов программных задач, использует репозиторий для всей технической и управляющей информации о проекте, концентрируясь при этом на поддержке, как правило, одной фазы или одного этапа разработки ПО. Категория workbench представляет собой интеграцию программных средств, которые поддерживают системный анализ, проектирование и разработку ПО; используют репозиторий, содержащий всю техническую и управляющую информацию о

проекте; обеспечивают автоматическую передачу системной информации между разработчиками и этапами разработки; организуют поддержку практически полного ЖЦ (от анализа требований и проектирования ПО до получения документированной выполняемой программы). Workbench, по сравнению с toolkit, обладает более высокой степенью интеграции выполняемых функций, большей самостоятельностью и автономностью использования, а также наличием тесной связи с системными и техническими средствами аппаратно-вычислительной среды, на которой workbench функционирует. По существу, workbench может рассматриваться как автоматизированная рабочая станция, используемая как инструментарий для автоматизации всех или отдельных совокупностей работ по созданию ПО.

Классификация по уровням связана с областью действия CASE в пределах жизненного цикла ПО. Однако четкие критерии определения границ между уровнями не установлены, поэтому данная классификация имеет, вообще говоря, качественный характер.

Верхние (Upper) CASE часто называют средствами компьютерного планирования. Они призваны повышать эффективность деятельности руководителей фирмы и проекта путем сокращения затрат на определение политики фирмы и на создание общего плана проекта. Этот план включает цели и стратегии их достижения, основные действия в свете целей и задач фирмы, установление стандартов на различные виды взаимосвязей и т.д. Использование верхних CASE позволяет построить модель предметной области, отражающую всю существующую специфику. Она направлена на понимание общего и частного механизмов функционирования, имеющихся возможностей, ресурсов, целей проекта в соответствии с назначением фирмы. Эти средства позволяют проводить анализ различных сценариев (в том числе наилучших и наихудших), накапливая информацию для принятия оптимальных решений.

Средние (Middle) CASE считаются средствами поддержки этапов анализа требований и проектирования спецификаций и структуры ПО. Их использование значительно сокращает цикл разработки проекта; при этом важную роль играет возможность накопления и хранения знаний, обычно имеющихся только в голове разработчика-аналитика, что позволит использовать накопленные решения при создании других проектов. Основная выгода от использования среднего CASE состоит в значительном облегчении проектирования систем, - проектирование превращается в итеративный процесс, включающий следующие действия:

- пользователь обсуждает с аналитиком требования к проектируемой системе;
- аналитик документирует эти требования, используя диаграммы и словари входных данных;
- пользователь проверяет эти диаграммы и словари, при необходимости модифицируя их;
- аналитик отвечает на эти модификации, изменяя соответствующие спецификации.

Кроме того, средние CASE обеспечивают возможности быстрого документирования требований и быстрого прототипирования.

Нижние (Lower) CASE являются средствами разработки ПО (при этом может использоваться до 30% спецификаций, созданных средствами среднего CASE). Они содержат системные словари и графические средства, исключающие необходимость разработки физических спецификаций. Имеются системные спецификации, которые непосредственно переводятся в программные коды разрабатываемой системы (при этом автоматически генерируется до 80-90% кодов). На эти средства возложены также функции тестирования, управления конфигурацией, формирования документации. Главными преимуществами нижних CASE являются: значительное сокращение времени на разработку, облегчение модификаций, поддержка возможностей прототипирования (совместно со средними CASE).

Необходимые отдельные вспомогательные программы выбираются разработчиком ПО обычно по своему усмотрению. Инструментальные средства, как правило, поддерживают определенные методы разработки в соответствии с некоторой моделью процесса создания ПО и содержат наборы правил и нормативных указаний, которыми следует руководствоваться в процессе разработки. Рабочие среды разработчика я разделил на интегрированные и экспертные. Интегрированные рабочие среды предоставляют инфраструктуру поддержки для данных, управления и интеграции системных представлений. Экспертные рабочие среды более интеллектуальны. Они включают базу знаний о процессах создания ПО и механизм, который в соответствии с выбранной моделью процесса создания ПО предлагает разработчику для применения те или иные вспомогательные программы и инструментальные средства.

Процесс управления конфигурацией обычно стандартизирован и включает выполнение заранее определенных процедур. Они требуют детализированного контроля за очень большим количеством данных. При сборке системы единственная ошибка в управлении может привести к некорректной работе системы. Поэтому очень важна поддержка процесса управления конфигурацией соответствующими CASE-средствами.

Процесс управления изменениями заключается в заполнении форм запросов на изменения, проведении анализа изменений и передаче этих форм и соответствующих конфигурационных элементов команде управления качеством и команде по управлению конфигурацией. Этот алгоритмический по своей природе процесс позволяет сравнительно легко интегрировать его с системой управления версиями, поскольку, упрощая, можно сказать, что задача управления изменениями заключается в передаче нужных документов нужным людям в нужное время.

Сборка систем — это очень трудоемкий вычислительный процесс. Например, процесс компиляции большой системы, состоящей из сотен компонентов, может занять несколько часов. Если компиляцию и связывание компонентов такой системы выполнять вручную, то оператор обязательно сделает какие-либо ошибки. Средства сборки систем автоматизируют этот процесс, что исключает потенциальные ошибки, совершаемые при ручном компилировании, и, возможно, сокращает время сборки системы.

Управление конфигурацией — это управление изменениями в программной системе. Основная роль команды по управлению конфигурацией заключается в правильном внесении изменений в систему. В больших проектах для отслеживания различных версий всех проектных документов разрабатывается и используется формальная схема именования документов.

Для управления конфигурацией необходима база данных конфигураций, где хранится вся информация о проведенных в системе изменениях и запросы на изменения. Для управления конфигурацией необходима схема идентификации версий. Версии можно идентифицировать по номерам, на основе значений атрибутов и на основе изменений, внесенных в систему. Выходная версия системы включает исполняемый код, файлы данных, конфигурационные файлы и документацию. Управление выходными версиями предусматривает определение даты выпуска системы, подготовку всей информации для распространения системы и документирование каждой выходной версии.