

РАЗРАБОТКА ПРИНЦИПОВ ПОСТРОЕНИЯ АДАПТИВНЫХ СИСТЕМ ЭЛЕКТРОННОГО ОБУЧЕНИЯ НА БАЗЕ СЕМАНТИЧЕСКОГО WEB-а

Босикашвили З., Бежанишвили Л., Гогишвили З., Капанадзе Д.
Грузинский Технический Университет

Резюме

Современные системы электронного обучения в первую очередь базируются на подаче учебного материала в виде метаданных разного стандарта (в основном используются стандарты ADL SCORM 2004 или IMS Learning Design), что приводит к увеличению сложности и стоимости подготовки учебного материала. Обязательным условием роста интеллектуальности электронных систем обучения является наличие в них механизмов формализации процесса структурирования знаний и адаптивности процесса обучения, что должно базироваться на единых синтаксических и семантических представлениях процессов обучения и учебного материала. Такому подходу вполне соответствует построение электронных систем обучения на базе семантического WEB-а. Важно, что в семантическом WEB-е посредством специализированных языков RDF, OWL можно создавать семантические модели произвольного уровня, а также описывать метаданные. Единое семантическое представление процессов обучения и учебного материала дает нам возможность осуществить автоматическую генерацию метаописаний учебного материала, автоматическое составление учебного плана, персонализацию обучения, семантический поиск учебного материала, а также взаимодействие с системой на языке близком к естественному.

ФОРМАЛЬНЫЙ МЕТОД АДАПТАЦИИ ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ ПРЕЦЕДЕНТОВ

Бежанишвили Л.
Грузинский Технический Университет

Резюме

Правильное управление проектированием и разработкой программного обеспечения позволяет избежать непреднамеренных ошибок проектирования и срывов сроков проектов, а также повысить качество и надежность программных продуктов. Менеджер проекта может лучше спланировать свой проект, если имеет доступ к опыту, накопленному в завершенных проектах, на основе которых проект адаптируется в соответствие с реальными потребностями. В данной статье приводится один из методов адаптации процесса разработки программного обеспечения на основе прецедентов.

Ключевые слова: адаптация, прецедент, адаптивное управление.

1. Введение

Эффективное *управление* процессом разработки программного обеспечения очень важно для общего успеха программного проекта. При *планировании* проекта нужно решить, какой процесс должен использоваться для создания ПО. Это важное решение, поскольку оно определяет значительную часть разработки. Существует несколько моделей процессов для разработки ПО. К наиболее общим из них относятся каскадная модель, итеративное расширение, создание прототипов и спиральная модель. Особенно широко используется каскадная модель, которая организует фазы в линейную последовательность, хотя, как правило, при реализации эту модель *адаптируют* в соответствие с реальными потребностями проекта.

2. Основная часть

Определяя используемый процесс для разработки программного обеспечения, менеджер проекта выбирает базовый процесс и решает, как его адаптировать, чтобы получить процесс, подходящий для данного проекта. Ни один определенный процесс, будь то стандартный процесс

организации, невозможно применить ко всем ситуациям и всем проектам. Определенный процесс нужно адаптировать, чтобы он соответствовал потребностям реального проекта.

Адаптация — это приспособление ранее определенного процесса организации, которое позволяет получить процесс, соответствующий конкретной предметной области или техническим потребностям проекта. Адаптацию можно рассматривать как такое добавление, удаление или изменение действий процесса, которое создает результирующий процесс, лучше приспособленный для достижения целей проекта, чем изначальный. Бесконтрольная адаптация фактически означает создание процесса с чистого листа. Для реального повторного применения определенных ранее процессов используются правила адаптации. В них задаются условия и типы изменений, которые следует вносить в стандартный процесс [1]. По существу в них определен набор разрешенных отклонений от стандартного процесса, который позволит получить оптимальный для проекта процесс. На рис.1 наглядно представлена роль инструкций по адаптации.

При рассмотрении новой проблемы (текущего проекта) находится похожий *прецедент* в качестве аналога. Можно попытаться использовать его решение, возможно, адаптировав к текущему случаю, вместо того, чтобы искать решение каждый раз сначала.

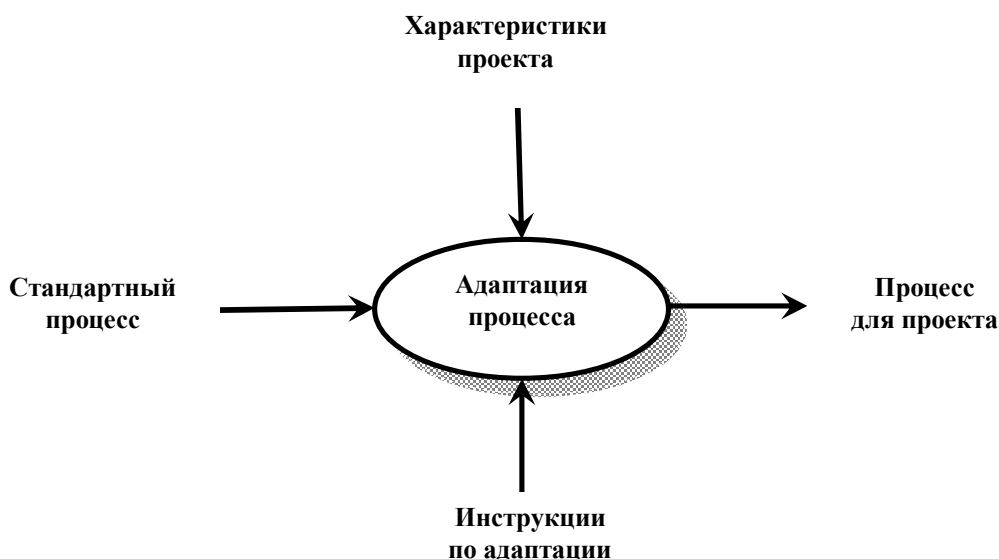


Рис. 1.

Предлагается следующая структура проектного прецедента для адаптивного управления:

1. Состояние до воздействия. Описание объекта (набор признаков, принадлежность к классу состояний).
2. Управляющее воздействие. Описание воздействия (здесь возможна формализация, в частности, классификация управляющих воздействий). Как частный случай, возможно отсутствие воздействия.
3. Состояние после воздействия. Описание объекта (набор признаков, принадлежность к классу состояний).
4. Исход (положительный исход/отрицательный/спорный).

Наполнение базы прецедентов может происходить как до момента начала управления на основе априорной информации, с помощью реальных или смоделированных прецедентов, так и в процессе управления, после обработки итога управляющего воздействия.

Классификация состояний может производиться с привлечением экспертного знания или путем предварительной кластеризации в этом случае будет выглядеть так (рис. 2). После применения регулирующего воздействия и оценки итога этого воздействия текущая ситуация превращается в прецедент, который заносится в базу прецедентов.

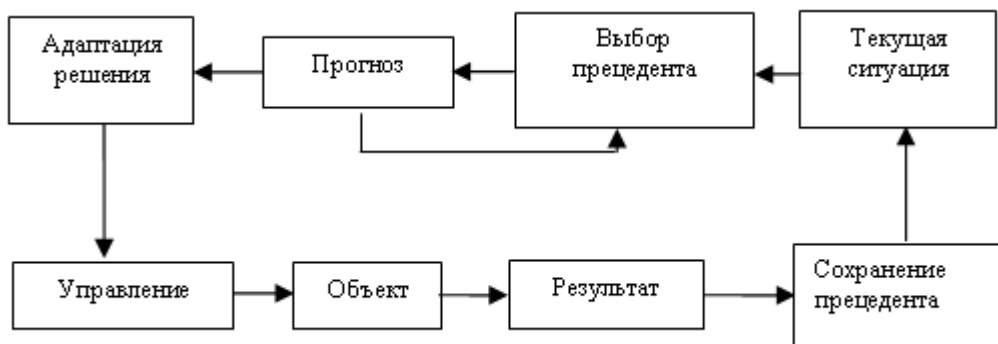


Рис.2. Схема адаптивного управления по прецедентам.

Отрицательный результат также является информативным и заносится в базу. Данная модель должна обеспечить решение следующих задач:

1. Формирование обобщенных образов состояний на основе априорной информации (обучение).
2. Идентификация состояния по его выходным параметрам.
3. Определение влияния входных параметров на перевод в различные состояния.

После того, как прецеденты извлечены, нужно выбрать "наиболее подходящий" из них. Это определяется сравнением признаков в текущей ситуации и в выбранных прецедентах. Затем выбирается способ измерения степени близости прецедента и текущего случая по каждому признаку, который пользователь сочтет полезным для достижения цели.

Из всех этих рассуждений вытекает алгоритм сопоставления прецедентов, представленный ниже на псевдокоде:

```

Set := ∅ {Set – множество близких стереотипов}
FOR Ster := Ster1 TO SterN DO {Ster1, ..., SterN – все стереотипы в
                               базе знаний}
  IF mun цели Ster совпадает с типом текущей цели THEN
    Match1 := частичная строка сопоставления объектов из описания
              уточняемой стереотипом цели с реальными объектами из текущей цели
    FOR Str := Str1 TO StrM DO
      {Str1, ..., StrM – строки сопоставления из памяти узла Store
       ситуации стереотипа Ster, которых нет в памяти узла End}
      IF Match1 является подстрокой Str THEN
        Node := узел-потомок узла Store
        WHILE mun Node <> End DO
          Выполнить проверку в узле Node
          IF проверка неуспешна THEN
            приписать к Str пометку о невыполненном условии
          END IF
          Node := узел-потомок Node
        END WHILE
        построить список недостающих фактов Facts по пометкам Str
        Set := Set + (Ster, ситуация Ster, Str, Facts)
      END IF
    END FOR
  END IF
END FOR
  
```

Алгоритмы адаптации предполагают наличие зависимости между признаками прецедентов и признаками содержащихся в них решений.

3. Заключение

В ходе адаптации проекта устанавливается последовательность выполняемых для проекта действий, которая затем используется для того, чтобы спланировать действия, составить для них график и сформировать основу для выполнения проекта. Проведение адаптации особо выделяется в плане проекта, поэтому когда пересматривается план, определение и адаптация процесса также пересматриваются.

Литература:

1. Панкаж Джалота. “Управление программным проектом на практике. Изд. "Лори", 2005
2. “Руководство к Своду знаний по управлению проектами (Руководство PMBOK®)”, третье издание, *Project Management Institute, Inc.*, 388, Американский национальный стандарт ANSI/PMI 99-001-2004.

პროგრამული უზრუნველყოფის შექმნის პროცესის ადაპტაციის ფორმალური მეთოდი პრეცედენტების გამოყენებით

ლოლიტა ბეჯანიშვილი
საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

პროგრამული უზრუნველყოფის დაპროექტების და დამუშავების სწორი მართვა იძლევა დაპროექტებისას დაუგეგმავი შეცდომების და ვადების ჩავარდნის თავიდან აცილების საშუალებას და ამის გარდა პროგრამული პროდუქტების ხარისხისა და საიმედოობის გაუმჯობესების გარანტიას. პროექტის მენეჯერს შეუძლია უკეთესად დაგეგმოს თავისი პროექტი, თუ მისთვის ხელმისაწვდომია დასრულებულ პროექტებში დაგროვებული გამოცდილება, რომელთა საფუძველზე ხდება პროექტის ადაპტაცია რეალური მოთხოვნების მიხედვით. ამ სტატიაში მოყვანილია პროგრამული უზრუნველყოფის შექმნის ადაპტაციის პრეცედენტებზე დაფუძნებული ერთ-ერთი მეთოდი.

A FORMAL ADAPTATION METHOD OF SOFTWARE DEVELOPMENT PROCESS BASED ON PRECEDENTS

Bejanishvili Lolita
Georgian Technical University

Summary

Correct management of software design and development helps to avoid inadvertent design errors and failures in meeting deadlines, while at the same time raises quality and reliability of the software. Project manager can plan better if has access to the experience piled up from completed projects, on which basis the project must be adapted according to real requirements. Adaptation method of software development based on precedents is represented in given article.