ЕКОЛОГІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004.413

Н.А.Сидоров

ЭКОЛОГИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Национальный авиационный университет
nikolay.sidorov@livenau.net

Анотація: *представлено застосування екологічного підходу до дослідження програмного забезпечення; розглядаються основні положення екології програмного забезпечення, як частини інженерії програмного забезпечення; наводяться три напрямки екологічних досліджень програмного забезпечення («зелене» програмне забезпечення, сталий розвиток, цифрові екосистеми); результати демонструються на прикладі авіаційного тренажеру.*

Аннотация: *представлено применение экологического подхода к исследованию программного обеспечения; рассмотрены основные положения экологии программного обеспечения как части инженерии программного обеспечения; приводятся три направления экологических исследований программного обеспечения («зеленое» программное обеспечение, устойчивое развитие, цифровые экосистемы); результаты демонстрируются на примере авиационного тренажера.*

Abstract: *Using ecological approach to software research is presented. Basics of the software ecology as a new branch of the software engineering are defined. Three directions of software ecology researchs ("green" software, sustainable development, digital ecosystems) are looked. The a vitiation simulator as example is presented.*

Ключові слова: *програмне забезпечення, інженерія програмного забезпечення, «зелені» інформаційні технології, екосистеми програмного забезпечення*

Введение

В статье рассматривается применение экологического подхода к исследованию программного обеспечения и формулируются основные положения нового раздела инженерии программного обеспечения – экологии программного обеспечения.

Можно указать три причины, вызывающие применение экологического подхода к исследованиям программного обеспечения. Первые две связаны с концепцией устойчивого развития и определяются влиянием программных продуктов и их производств на окружающую среду. Третья причина основана на необходимости наблюдений за программным обеспечением как за организованной системой в контексте реального мира.

Первая причина основывается на предположении, что программные продукты в составе систем и технологий и их производство в индустрии программного обеспечения, как в других инженерных отраслях влияют на окружающую среду [1, 2]. Это влияние может быть вредным и неуправляемым.

Вторая причина основывается на известном парадоксе развития научно-технического прогресса, суть которого заключается в том, что приходится осуществлять деятельность по ликвидации результатов деятельности. При этом, в индустрии программного обеспечения, как в любой отрасли деятельности человека, и понятна и разумна постановка следующих задач – создавать как можно меньше отходов и утилизировать как можно больше из того, что стало не нужным [3, 4].

Третья причина основывается на наблюдении, что эффективное планирование развития и обслуживания программных продуктов требует понимания не только их места в реальном мире, учета их установившихся взаимодействий с реальным миром, а также взаимодействий внутри отдельного продукта, и внутри элементов продукта, но и расширение участников и типов взаимодействий, подлежащих исследованию, понимания места программного обеспечения в историческом контексте [5].

Анализ исследований применения экологического подхода в программном обеспечении

Под экологическим подходом к исследованиям будем понимать методологию исследований объекта, в данном случае программного обеспечения, как целой (системной) части окружающей среды, рассматриваемой, как правило, в форме экосистемы, с учетом устойчивого развития.

Наиболее близким к экологическому подходу является системный подход [6]. Однако, экологический подход, дополняя системный, позволяет исследовать программные продукты и среды, в которых они разрабатываются, используются и эволюционируют в новых ракурсах – экосистемном, эволюционном (историческом) и устойчивого развития. Таким образом, экологический подход – это эконцентрированный системный подход. В широком смысле объектом исследования является взаимодействие программного обеспечения и природы. В узком – объект, это взаимодействие программного обеспечения со средой. Предметом исследованияя являются принципы, методы, организации, объекты и процессы взаимодействия.

В экосистемном ракурсе, в части реального мира – экосистеме, одним из объектов которой является программный продукт исследуется обмен энергией и веществом, функциональные и социальные связи. При этом, принимаются во внимание связи живых организмов экосистемы между собой и с их неживым (техническим) окружением. Таким образом расширяются участники и типы взаимодействий, подлежащих исследованию, появляются новые свойства и характеристики. При этом, можно говорить об исследованиях экосистем программного обеспечения [7]. В таком же ракурсе, видимо, можно исследовать программное обеспечение, говоря о программном обеспечении как экосистеме.

В эволюционном (историческом) ракурсе исследуются экосистемы с программным обеспечением, с целью понять причины их изменений, степень и направление влияния программного обеспечения на элементы той или иной экосистемы, причины особенности и морфологии программных продуктов в контексте экосистем.

В ракурсе устойчивого развития исследуются процессы экосистем, имеющих программные продукты, с целью гармонизации экономических, социальных и экологических факторов, влияющих на устойчивое современное состояние общества и удовлетворение его потенциальных потребностей в будущем. В настоящие время программное обеспечение является результатом инженерной технической деятельности [8]. Индустрия программного обеспечения потребляет значительные ресурсы, а программные продукты, функционируют во всех хозяйственных отраслях, в технологиях, средствах и особенно информационных технологиях. Поэтому, прямо или косвенно индустрия и программные продукты оказывают значительное влияние на окружающую среду [2]. Кроме того, программное обеспечение потенциально опасных технических объектов, в случае его неправильного функционирования может быть причиной прямых экологических катастроф.

Постановка проблеми

В контексте концепции устойчивого развития [2, 9], программное обеспечение является его активом, вследствие того, что оно продукт производственной деятельности человека. Известно, что современная производственная деятельность все чаще вступает в противоречие с процессами, поддерживающими устойчивый круговорот в биосфере. В. Вернадский указывал на необходимость решения задачи перехода от стихийных взаимодействий человека и биосферы к сознательным, которые превращают биосферу в ноосферу и обеспечивают устойчивое развитие [10]. Сейчас, эта задача вновь актуальна в контексте программы совместных действий в интересах устойчивого развития, а программное обеспечение играет важную роль в ее решении вследствие того, что оно является следующим:

– основой всех цифровых информационных и коммуникационных технологий;

– наукоемким образованием;

– продуктом коллективной, все чаще глобальной деятельности.

В контексте экологического подхода программное обеспечение рассматривается, как технический объект взаимодействующий с окружающей средой. Такой аспект рассмотрения относится к той части экологии, которая иногда обозначается «относящейся к окружающей среде» (environmental) [9], а отрасль разработки программных продуктов становится объектом исследований инженерной экологии техносферы [3, 11]. Основные цели таких исследований – это сохранение природных ресурсов и защита окружающей среды. Однако, с учетом концепции устойчивого развития достижение целей требует постановки новых задач, решение которых обеспечит переход от онтропоцентрической концепции к тотальной экологизации деятельности при

разработке и исследовании программного обеспечения.

Применение экологических исследований в индустрии программного обеспечения показывает, что их распространение идет на основе трех принципов и в трех основных направлениях.

Общими экологическими принципами, которыми нужно руководствоваться в стремлении достичь экологических результатов являются следующие [12, 13]:

– эко-эффективность (eco-efficiency) – предполагает комбинирование целей эффективности в традиционном смысле с экологическими целями и направление их достижения на улучшение качества жизни людей и уменьшение влияния на окружающую среду. Применение принципа направлено на решение задачи - где и как должно применяться программное обеспечение, что бы уменьшить использование природных ресурсов и снизить вредное влияние на окружающую среду;

– эко-справедливость (eco-equity) – предполагает распределение ресурсов между живущими и будущими поколениями с учетом концепции устойчивого развития [9]. Применение принципа направлено на решение экоцентрических задач, тогда как сейчас решаются, главным образом, антропоцентрические и техноцентрические задачи;

– эко-результативность (eco-effectives) – предполагает создание «чистых» (безотходных) систем и технологий, в которых отходы одних процессов являются источниками для других процессов и в конечном итоге осуществляется их применение с пользой. Применение принципа, направлено на решение задач уменьшения загрязнения путем снижения отходов. Лозунг результативности – «Делайте вещи правильно»! Лозунг эко–результативности - «Делайте правильные вещи». Поэтому, одна из задач экологического проектирования программного обеспечения состоит в том, что бы оценить «правильность» разрабатываемого продукта в экологическом аспекте.

Можно указать направления применения экологического подхода в исследованиях программного обеспечения:

– первое, связано с применением к программному обеспечению общих принципов и требований экологичности производства и использования технических объектов [14, 15];

– второе, связано с реализацией ресурсосберегающих и безотходных производств программного обеспечения [16, 17];

– третье, связано с исследованием программного обеспечения, рассматривая его как часть цифровой экосистемы (digital ecosystem) [7, 18, 19].

Рассмотрим подробно указанные направления.

Экологическое производство и использование программного обеспечения

Данное направление основано на применении принципов эко-эфективность и эко-результативность.

Наиболее ярким проявлением направления является Green Software – «Зеленое» программное обеспечение в контексте Green IT – «зеленых информационных технологий» [14, 15, 20-22]. В рамках этого направления основной акцент делается на использовании программного обеспечения, как средства прямого или косвенного уменьшения вредного влияния на окружающую среду. Например, использование электронной почты вместо бумажной, видеоконференций вместо поездок, виртуальных офисов виртуальных работников (работа на дому) вместо реальных. Кроме того, рассматривается влияние на окружающую среду при производстве и эксплуатации программного обеспечения, например, эффективное энергопотребление, уменьшение выделения CO и CO_2 в атмосферу и разогрев оборудования, других подобных аспектов взаимодействия между окружающей средой и вычислительными средствами на всех этапах жизненного цикла программного обеспечения. В рамках Green Software важным также является минимизация использования основных (объем памяти, время и быстродействие процессора) и дополнительных (объем внешней памяти, распределение каналов) вычислительных ресурсов.

Примерами такого подхода являются стратегия корпорации IBM по разработке Green Software [14] и стратегия Green Asus компании Asus [15].

При разработке программного обеспечения корпорация IBM предлагает следующие подходы, большую часть из которых она использует в качестве стратегии IBM - Green Software Strategy, при разработке собственного «зеленого» программного обеспечения:

– сокращение деловых поездок, используя on-line коммуникации;

– загрузка недостаточно загруженных серверов, для уменьшения потребления энергии и рабочей площади;

– планирование графика выполнения рабочей нагрузки во время умеренной нагрузки, для того, чтобы использовать энергию меньшей стоимости;

– эффективное управление ресурсами жизненных циклов;

– оптимизация разных приложений для уменьшения ресурсов и энергии, которые потребляются;

– объединение и укрупнение ресурсов для уменьшения необходимой площади и упрощения вычислительной инфраструктуры;

– оптимизация нагревания, вентиляции и кондиционирования воздуха на рабочих местах для сокращения потребления энергии;

– эффективное управление хранением данных;

– сокращение потребления энергии при уменьшении рабочей нагрузки;

– оптимизация бизнес-процессов для сокращения потребления энергии и операционных затрат;

– сокращение использования бумажных носителей путем введения в бизнес-процессы электронных документов.

Стратегия Green Asus была сформулирована в 2000 году и делится на четыре части. Первая, Green-Design, должна обеспечивать создание и внедрение в производство «зеленых» компонентов, которые состоят из легко восстанавливаемых материалов и эффективно утилизируются. Вторая, Green-Manufacturing обеспечивает разработку и внедрение экономических и экологических технологий на производстве. Например, компания полностью отказалась от использования свинца, а до 2009 года собиралась прекратить применение галогенов. Третья часть, Green-Procurement регламентирует логистику товаров, использование материалов и компонентов от внешних производителей. Кроме этого, стимулируются поставщики на пересмотр своего отношения к проблемам экологии. Четвертая часть, Green-Marketing решает задачи утилизации старых компьютеров. Компания рассматривает последнюю часть важной и в 2010 году собирается начинать свою деятельность по переработке электроники в Украине.

Другие компании также начинают решать задачи этого направления. Например, Intel пересматривает свое отношение к проблемам экологии, разрабатывая новый стандарт, который минимизирует вредное влияние на окружающую среду на всех этапах производства.

В качестве задач этого направления рассматриваются следующее [20 - 22]:

- экономия энергии;
- электронное управление документами;
- виртуализация деятельности;
- «зеленые» центры данных;
- утилизация отходов.

Для реализации могут использоваться сервера-лезвия, нулевые клиенты, минифреймы, виртуальные машины, облачные сервисы, «зеленые» информационные системы, «зеленые» Grid-кластеры вычисления, платформы электронного документооборота.

Собирательным образом средства этого направления является цифровая экосистема [19].

Новыми задачами являются следующие:

- разработка и наследование моделей «зеленых» систем и технологий;
- критерии оценки «зелености»;
- методы, принципы и средства создания «зеленых» систем и технологий;
- методы трансформации «незеленых» систем и технологий в «зеленые».

Ресурсосберегающее и безотходное производство программного обеспечения

Ресурсосберегающее производство – это производство и реализация продуктов с минимальным расходом вещества и энергии на всех этапах производственного цикла и с наименьшим воздействием на человека и природные системы [3]. Такое производство – залог устойчивого развития. Основой ресурсосберегающего производства являются ресурсосберегающие технологи [11].

Рассмотрим существующие ресурсы инженерии программного обеспечения (принципы, процессы, конструкции), которые играют важную роль и могут использоваться как основа для построения ресурсосберегающих и безотходных технологий разработки и сопровождения программного обеспечения (табл. 1).

Первой такой конструкцией была подпрограмма, которую рассматривали как средство сокращения текста программы. Гибкость конструкции обеспечивала параметризация. Вскоре взгляд на подпрограмму начал изменяться, и её стали

рассматривать, как средство структурной организации программ.

Приспособление программы к ожидаемым изменениям постановки задачи или сопровождения сводилось к замене тела подпрограмм или к изменению значений некоторых фактических параметров в обращении к подпрограммам.

Таблица 1. Методы, принципы, конструкции экологического программного обеспечения

№ п/п	Методы / Ресурсы	Подпрограммное (процедурное) программирование	Объектно-ориентированое программирование	Модульное программирование
1	Конструкция	Подпрограмма (закрытая, открытая), шаблон	Класс (C++, C)	Модуль (Модула 2,3), пакет (Ada)
2	Принципы реализации конструкций	Параметризация	Полиморфизм, наследование, класификация	Раздельная компиляция
3	Принципы использования конструкций	Процедурная абстракция абстракция выполнения, шаблонирование	Класификация	Композиция
4	Фундаментальная абстракция	Абстрактный тип даних		

Поэтому, подпрограммы применялись как процедурные абстракции, абстракции выполнения (закрытые подпрограммы) или шаблоны (открытые подпрограммы). В качестве метода применения подпрограммы использовалось подпрограммное (процедурное) программирование. Позднее, следуя стремлению улучшить конструкцию для реализации абстрактных типов данных были созданы модуль и класс.

В 1984 году при развертывании исследований, связанных с повторным использованием программного обеспечения обращалось внимание на то, что с годами количество принципиально новых применений вычислительных машин уменьшается [23, 24]. Это свидетельствовало о том, что сопровождаемое программное обеспечение, которое впоследствии стали называть наследуемым, содержало опыт, и его следовало повторно использовать при создании нового программного обеспечения. С учетом этого, жизненный цикл программного обеспечения был расширен дополнительными процессами (рис. 1).

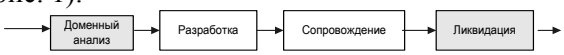


Рис. 1 Расширение жизненного цикла программного обеспечения

С одной стороны, он был дополнен доменным анализом [25], цель которого – путем анализа опыта, накопленного в домене строить повторно используемые решения для применения в разработке нового программного обеспечения. С другой стороны, в жизненный цикл, в контексте фазы ликвидации были введены процессы, которые связаны с утилизацией программного обеспечения (рис.2, таб. 2) [17]

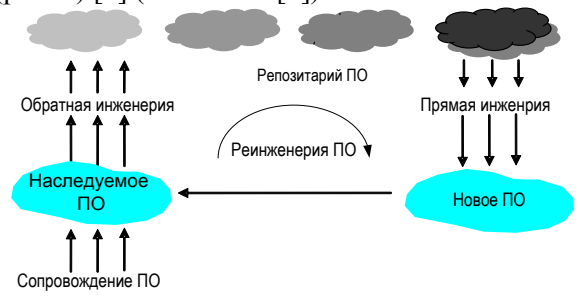
Утилизация включает три процесса – повторное использование, восстановление и переработку наследуемого программного обеспечения. Все неутилизированные компоненты уничтожаются (рис. 2, табл. 2).

Рис. 2 Содержание фазы ликвидации

Таблица 2 Процессы утилизации

Процессы утилизации	Общая характеристика процессов утилизации	Методы инженерии программного обеспечения	Средства инженерии программного обеспечения
Повторное использование	Создание компонентов, хранение и применение компонентов	Классификация, хранение, поиск	Среда программирования, мониторинговые системы, экспертные системы
Восстановление	Анализ программного обеспечения, преобразование программного обеспечения на одном уровне	Синтаксический, семантический анализ, методы прямой и обратной инженерии	Анализаторы преобразователи абстрактори экстракторы
Переработка	Преобразование программного обеспечения на разных уровнях представления. Анализ программного обеспечения	Синтаксический и семантический анализ, методы реверсивной инженерии	CASE, преобразователи, абстрактор, экстрактор.

Реализация процессов утилизации привела к появлению обратной (реверсивной) инженерии программного обеспечения, соединение которой с прямой инженерией дало реинженерию и позволило в экологическом аспекте рассматривать разработку и сопровождение программного обеспечения как его круговорот (рис. 3) [4] (horseshoe [5]).



ПО – программное обеспечение

Рис. 3 «Круговорот» программного обеспечения

В доменном анализе и утилизации программного обеспечения сложилось несколько способов создания (табл. 3) и аспектов применения (рис. 4, [4]) компонентов повторного использования.

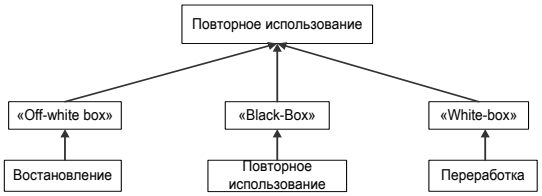


Рис. 4 Аспекты применеия компонентов повторного использования

Таблица 3. Создание компонентов программного обеспечения

Характеристика	Способы создания компонентов программного обеспечения		
	Утилитарный	Индуктивный	Дедуктивный
Характер способа	Экстенсивный	Интенсивный	Интенсивный
Метафора	«Свалка»	«Фабрика компонентов»	«Высокая технология»
Тип компонентов	Пассивные	Больше пассивные	Больше активные
Техника применения	Композиционная	Композиционная и адаптивная	Адаптивная и композиционная
Метод разработки компонентов	Утилизация	Доменный анализ, снизу вверх	Структуроное проектирование, сверху вниз
Тип применения	Случайный	Случайный, систематический	Систематический
Средства переработки и возобновления	Не нужны	Нужны	Не нужны
Мощность инфраструктуры применения	Высокая	Высокая	Низкая
Форма повторного использования	«Черный ящик»	«Белый ящик»	«Черный ящик»

Индуктивный и дедуктивный способы создания компонентов впоследствии позволили превратить компоненты повторного использования в компоненты многократного использования, которые сейчас составляют основу компонентной разработки программного обеспечения [8].

Появились две парадигмы применения компонентов: адаптивная и сборочная. В соответствии с адаптивной парадигмой – применяется модель программного обеспечения, похожая на фрейм, где компоненты многократного использования заполняют слоты. В соответствии с сборочной парадигмой – компоненты многократного использования объединяются программным кодом как «клеем».

Таким образом, видно, что в инженерии программного обеспечения есть принципы, процессы и конструкции, которые могут использоваться при построении безотходных, ресурсосберегающих технологий и продуктов. Следует разрабатывать показатели ресурсосбережения, устанавливая нормы выполнения заданий ресурсосбережения [26, 27].

Экосистемы программного обеспечения и программное обеспечение как экосистема

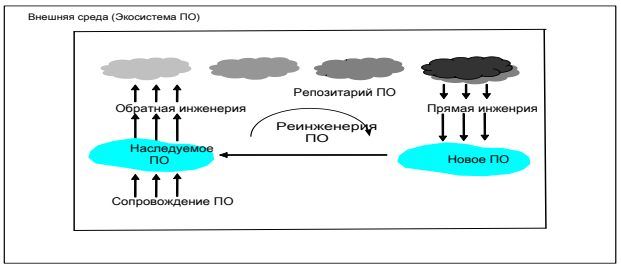
Программное обеспечение с помощью экологического подхода может исследоваться в двух аспектах – программное обеспечение в составе экосистемы и программное обеспечение как экосистема.

В работе [5] показано, что программное обеспечение характеризуется следующим:

– изменением (развитием) – непременное свойство программного обеспечения, обусловленное наличием обратных связей и связанное с законами эволюции программ;

– наличием метасистемы, которая включает субъекты и продукты деятельности, процессы и организацию, содержит большое количество обратных связей, стабилизирующих внутренних механизмов, влияющих на процессы планирования, управления и повышения их эффективности; эффективное планирование и обслуживание программы требует понимания ее места в метасистеме, а также взаимодействий как между элементами, так и внутри них.

При этом, программы, о которых идет речь, по классификации SPE [5] являются E – программами, а метасистема, – это их внешняя среда – реальный мир (рис. 5).



ПО- программное обеспечение

Рис. 5 E – программа в реальном мире

С учетом указанных характеристик для изучения программного обеспечения целесообразно применить экологический подход, исследуя экосистемы содержащие программное обеспечение, и решая, например, следующие задачи:

– оценки качества исследуемого программного обеспечения – как части экосистем;

– понимания причин влияющих на программное обеспечение, изменений компонентов, источников и факторов воздействий;

– прогноза устойчивости программного обеспечения как части экосистем и допустимости изменений.

Тогда, одна из основных задач исследований – мониторинг будет состоять в сборе, накоплении, систематизации и анализе информации о количественном характере взаимосвязей внутри программного обеспечения, между программным обеспечением и средой - экосистемой. Таким образом, для выполнения исследований должен быть организован мониторинг программного обеспечения в контексте экосистемы, а исследование программного обеспечения, как части экосистемы должно осуществляться на основе экологического подхода, например, в следующих аспектах:

– морфологическом – исследуется устройство экосистемы;

– функциональном – исследуются функции компонентов экосистемы, обычно в терминах входных, выходных и управляющих параметров, возмущающих воздействий и параметров состояния.

Принимая во внимание, что исследуется наследуемое программное обеспечение важную роль в экологическом исследовании будет играть реверсивная инженерия [18].

Программное обеспечение можно исследовать как экосистему. В этом аспекте рассматривается два типа взаимодействий – внешние и внутренние.

Внешние взаимодействия обусловлены наличием других экосистем. Например, очень часто программное обеспечение используется в составе или рядом с другим программным обеспечением, о существовании которого разработчик не мог знать. При этом эволюция такого программного обеспечения зависит от эволюции других приложений, а интерес представляют задачи создания моделей таких экосистем и моделей их эволюции [5].

Внутренние взаимодействия обусловлены наличием в программном обеспечении клонов программ, программ-агентов, «обществ» программ [28]. Возникают задачи исследования устройства такого программного обеспечения, принципов взаимодействия членов «обществ» программ и «обществ» между собой [28].

Пример. Программное обеспечение авиационного тренажера.

Рассмотрим применение экологического подхода к исследованиям программного обеспечения авиационного тренажера, который эксплуатируется в Национальном авиационном университете [29]. На рис. 6 приведена схема холистической модели экосистемы программного обеспечения авиационного тренажера. Модель представляет компоненты экосистемы, указывает связи между ними и пути обмена информацией, энергией и воздействиями. Модель включает три подсистемы (рис. 6, 7) – программное обеспечение тренажера, университет и государство.

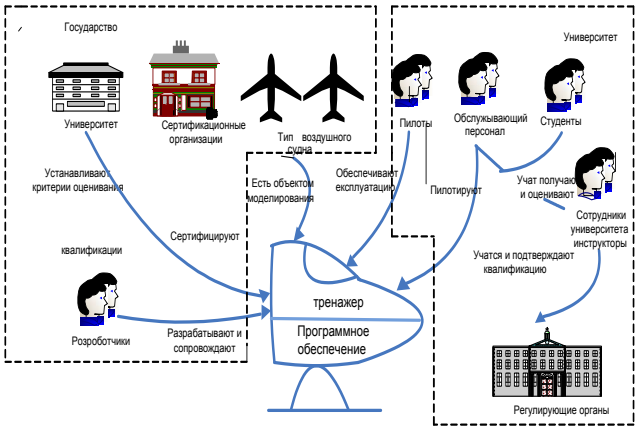


Рис. 6 Холистическая модель экосистемы программного обеспечения авиационного тренажера

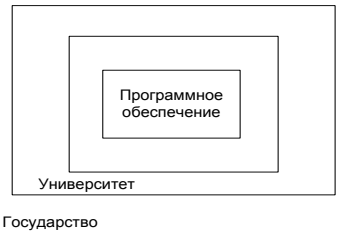


Рис. 7 Подсистемы модели экосистемы программного обеспечения

Очевидно, в зависимости от целей исследований в экосистемы могут вовлекаться новые компоненты и пути обмена в границах указанных подсистем.

Используя, приведенную модель можно, например, показать «жизнь» программного обеспечения в контексте указанной экосистемы применяя экологический подход к исследованиям в историческом аспекте. На рис. 8 показана во времени, в границах двух экоподсистем (государство, университет) «жизнь» программного обеспечения.

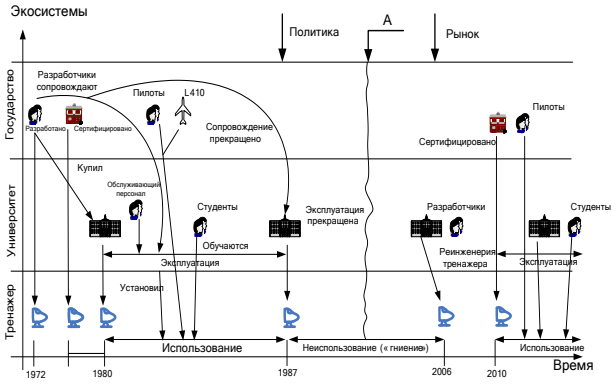


Рис. 8 Экосистемы тренажера. Исторический аспект

Разрыв «А» соответствует коренным изменениям, которые происходили в 90х годах прошлого столетия и повлияли на экосистему тренажера. Особенностью рассматриваемого тренажера является то, что после длительного периода «застоя» была выполнена его реинженерия, вследствие которой эксплуатация была возобновлена. В связи с этим представляет интерес исследование программного обеспечения как экосистемы.

На рис. 9 показан состав авиационного тренажера, а на рис. 10 состав его программного обеспечения до начала «гниения» (1987 год, рис.8).

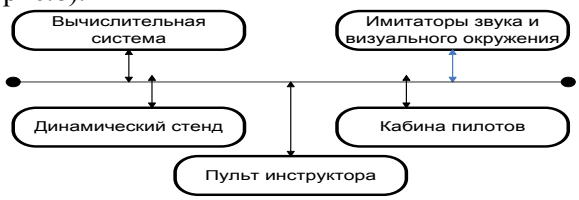


Рис. 9 Состав авиационного тренажера

Авиационный тренажер как экосистему можно рассматривать состоящим из двух подсистем – аппаратной и программной.

Рис. 10 Состав программного обеспечения

Значительная часть оборудования тренажера была аналоговой. Программное обеспечение было написано на специальном языке низкого уровня с применением специальных методов алгоритмизации и программирования. Это показали результаты реверсивной инженерии. На рис. 11 представлены результаты исследования тренажера как экосистемы в историческом аспекте.

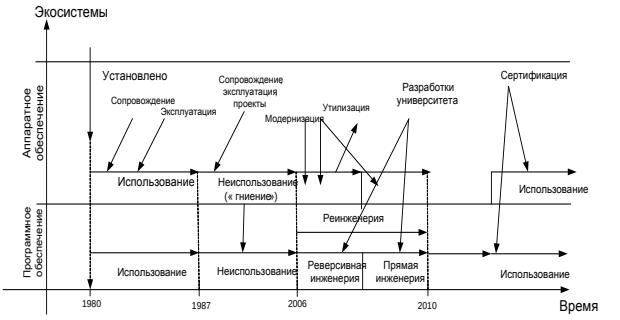


Рис. 11 Авиационный тренажер как экосистема. Исторический аспект

В результате модернизации оборудования и реинженерии программного обеспечения изменились состав и связи компонентов авиационного тренажера (рис. 12).

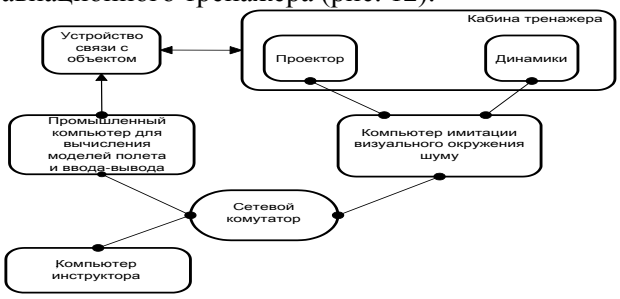


Рис. 12 Состав авиационного тренажера после модернизации

При этом, следуя указанным в начале статьи трем направлениям применения экологического подхода были достигнуты следующие результаты:

– ликвидируемые аппаратные компоненты тренажера были утилизированы;

– за счет полного перехода с аналоговой обработки на цифровую энергопотребление снижено;

– за счет применения реинженерии программного обеспечения достигнуты устойчивость и экономия ресурсов.

Заключение

Таким образом, принимая во внимание представленные направления экологических исследований и ключевую роль программного обеспечения, в информационных технология, видимо, в целом, можно говорить об экологии программного обеспечения как разделе инженерии программного обеспечения.

Практическим результатом экологических исследований была бы разработка научно-обоснованных технологий создания и применения программного обеспечения, основанных на оптимальном использовании ресурсов, в том числе и природных, обеспечивающих их поддержку, восстановление и контролированное устойчивое развитие.

Однако, такой результат может быть получен только путем широкой пропаганды экологического подхода среди разработчиков и пользователей программного обеспечения и данная статья посильный вклад в этом направлении.

Перечень ссылок

1. [Баландин, 1999] Р.К. Баландин, В.Н. Вернадский: жизнь, мысль, бессмертие. М., «Знание», 1979.

2. [Сидоров, 2006] М.О.Сидоров. Экология программного обеспечения. – Материалы Всеукраинской конференции аспирантов и студентов «Инженерия программного обеспечения 2006» – К.: НАУ, 2006.

3. [Мазур, 2006] И.И. Мазур, О.И. Молданов. Курс инженерной экологии: - М.: Высш.шк., 1999.- 447 с.

4. [Сидоров, 1990] Н.А. Сидоров, А.Н. Шарепа. Средство для утилизации программного обеспечения // УСиМ.- 1990.- №5.- с. 50-54.

5. [Lehman, 1985] M.M. Lehman, L.A. Belady. Program Evolution.- Academic Press.- 1985.- 532 p.

6. [Николаев, 1985] В.Н. Николаев, В.Н. Брук. Системотехника: методы и приложения.- Ленинград.- Машиностроение.- 1985.- 190 с.

7. [Messershmitt, 2003] D.G. Messershmitt, C. Szyperski. Software Ecosystems: Understanding an Indispensable Technology and Industry. – MIT press. – 2003. – 233 p.

8. [Sommerville, 2001] I. Sommerville. Software Engineering.- Addition- Wesley.- 2001.- 625 p.

9. [Угольницкий, 2002] Г.Л. Угольницкий. Иерархическое управление устойчивым развитием социальних организаций. – Общественная наука и современность. - №3. – 2002. – с. 133 – 140.

10. [Вернадский, 1994] В.Н. Вернадский. Несколько слов о неосфере. – Успехи современной биологии. - №18. – 1994. –с. 113 – 120.

11. [2000] Словарь терминов по экологии.- 2000 г. – 210 с.

12. [Dyllick, 2002] I. Dyllick, K. Hockerts Beyond the business case for corporate sustainability. – Busenes strategy and the Environment. – vol. 11. – pp. 130-141.

13. [Chen, 2008] I. W. Chen, R.T. Watson Information systems and ecological sustainability. – Journal of systems and Information technology.- v. 140. – n.3. – 2008.- pp. 186-201.

14. [IBM Software, 2008] IBM Software: A green strategy for your entire organization. IBM Software for a greener world June. 2008. NY 10589. U.S.A. Produced in the United States of America. May 2008.

15. [Турнов, 2008] Н. Турнов. Зеленый свет для ASUS // PC WEEK/UE.- 24(93).- 2008.

16. [Сидоров, 1989] Н.А. Сидоров. Повторное использование программного обеспечения// Кибернетика. – 1989. - №3 – с.46-51

17. [Сидоров, 1994] Н.А. Сидоров. Утилизация программного обеспечения - экономический аспект // Кибернетика и системный анализ- 1994.- №.3.- с. 151-166.

18. [Lungu, 2009] M.F. Lungu. Reverse Engineering Software Ecosystems. – Doct. Diss. – USI.- 2009. –208 p.

19. [Florina, 2000] C. Florina The digital ecosystem – creating digital divides. – Conf. Proc. – oct. 16. 2000.

20. [Webber, 2009]. L. Webber, M. Wallance Green Tech: thow two and to plan implement sustain – able IT solutions.- FBACOM.- 2209.- 29 p.

21. [Velye, 2008] Y. Velye, A. Velye, R. Elsenpeter Green IT: Rednee your Information systems Environmental input while adding to the bottom line.- 2008

22. [Schulz, 2009] G. Schulz. The Green and Virtual Data Center.- Taylors Francis G.- 2009.- 218 p.

23. [Boehm, 1987] B.W. Boehm. Improving Software Productivity // Computer, v. 20, n. 9, 1987, г. 43-57

24. [Biggerstaff, 1984] T. Biggerstaff. Foreword //IEEE Trans. on Software Engeneer. v. 10, n. 5, - 1984 г. 474-476

25. Prieto-Diaz, 1987] R. Prieto-Diaz Domian. Analysis For Reusability // Compsac 87 Oct., 7-9.-1987.- г. 23-30.

26. ГОСТ 52107-2003 Ресурсосбережение. Классификация и определение показателей.

27. ГОСТ 52104-2003 Ресурсосодержание. Термины и определения.

28. [Любимский, 2009] Э.З. Любимский. На пути к построению общества программ.- Программирование. - №1. - 2009. - с. 4-10.

29. [Луцький М., 2010] М. Луцький, М. Сидоров, Ю Рябокінь Підтримка придатності та супроводження експлуатації програмного забезпечення авіаційної техніки. – Проблеми програмування. - 2 - 3. – Київ. – 2010. – с. 229-239.

Сведения об авторе

Сидоров Николай Александрович, д.т.н., проф., декан факультету компьютерных наук, заведующий кафедры инженерии програмного обеспеченния Национального авиационного университета, научные интиресы – инженерия програмного обеспечения, обучение, e-mail nikolay.sidorov@livenau.net

Стаття надійшла до редакції 15.02.2010

