

Метод построения системных моделей мультипроекта инвестиционно-строительной корпорации

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ»

Основным методом, используемым для изучения инвестиционно-строительной корпорации, является системный подход – направление методологии специального научного познания, в основе которого лежит исследование объектов как систем, где система понимается как совокупность элементов, находящихся в отношениях и связях друг с другом, образующая определенную целостность, единство. Системный подход к управлению широко рассматривался в работах современных отечественных и зарубежных авторов: В.М. Илюшко [1, 2], О.Е.Федоровича [2], А.Р. Емада [3], Г. Коллинза [4], Р.З. Акберина [5], Р. Уотермана, [6], С. Бира [7] и др.

При определении понятия «система» необходимо учитывать теснейшую взаимосвязь его с понятиями целостности, структуры, связи, элемента, отношения, подсистемы др. Лишь в рамках такого семейства определений удастся отразить основные системные принципы: целостности, структурности, взаимозависимости системы и среды, иерархичности, множественности описания каждой системы. Важной особенностью большинства систем является передача в них информации и наличие процессов управления. Как часть системного подхода используется системный анализ. Привлечение методов системного анализа для решения указанных проблем необходимо, прежде всего, потому, что в процессе принятия решений приходится осуществлять выбор в условиях неопределенности, которая обусловлена наличием факторов, не поддающихся строгой количественной оценке. Процедуры и методы системного анализа направлены именно на выдвижение альтернативных вариантов решения проблемы, выявление масштабов неопределенности по каждому из вариантов и сопоставление вариантов по тем или иным критериям эффективности.

Важнейшие принципы системного анализа сводятся к следующему: процесс принятия решений должен начинаться с выявления и четкого формулирования конечных целей; всю проблему следует рассматривать как целое, как единую систему и выявлять все последствия и взаимосвязи каждого частного решения; необходимы выявление и анализ возможных альтернативных путей достижения цели. Изучение любой системы, обусловленной эффективным существованием целостных комплексов элементов в материальном мире, предполагает наличие объекта исследования, исследователя и цели. Наличие двух последних приводит к тому, что один и тот же реально существующий объект становится источником выявления целого ряда систем.

Как известно, системы классифицируются по степени сложности. В данной работе в качестве объекта исследования будем рассматривать инвестиционно-строительную корпорацию, относящую к сверхсложным (большим) системам, которые практически не поддаются полному, исчерпывающему описанию ввиду своей динамичности, являются вероятностными и проявляют свойства, не вытекающие непосредственно из свойств, составляющих систему элементов. Кроме того, инвестиционно-строительная корпорация как объект изучения относится к классу целенаправленных систем, стремящихся к целесообразному

поведению, к достижению наиболее предпочтительных состояний (стратегия/тактика).

Сложность корпорации как системы предопределяет и методологию её исследования. Важнейшей особенностью при ее исследовании и представлении является неразрывное единство двух подходов: процессного (событийного – динамического) и структурного (статического) [8].

Исследование корпорации как системы образует две составляющие – собственно систему управления (корпорацию) и объект управления (мультипроект) [9]. Функция системы проявляется в ее поведении, и при процессном подходе исследователя интересует не столько то, из чего состоит система, сколько то, как она себя ведет. Это легло в основу современного очень популярного направления исследования – бизнес-моделирование [10, 11], главным в котором являются определенные закономерности (бизнес-процессы, алгоритмы) функционирования (поведения) системы. Разрывы в бизнес-процессах приводят к невозможности или некорректности достижения конечных целей.

При всей важности процессного подхода к исследованию корпорации как сверхсложной системы он далеко не исчерпывает возможности его познания. Из философии известно, что всеобщим принципом развития познания служит переход от явления к сущности вещей. В системном анализе этот принцип конкретизируется в переходе от исследования процессов (функции) систем к изучению их структуры.

Цель анализа – возможно полное познание закономерностей функционирования системы при существующей структуре. Задача синтеза – спроектировать такую структуру, при которой наилучшим образом будут реализованы заданные функции. Центральной процедурой системного анализа является построение обобщенной (системной) модели (или моделей), отображающей все факторы и взаимосвязи реальной ситуации, которые могут проявиться в процессе осуществления решения.

А.М. Гаджинский [12] в своих работах наиболее наглядно показал, что существо системного подхода отчетливо проявляется при его сравнении с классическим индуктивным подходом к формированию систем.

Таким образом, проведенный анализ свидетельствует о том, что для управления любой системой требуется понимание процессов, происходящих внутри ее. Поэтому для эффективного управления инвестиционно-строительной корпорацией важно понимание того, что есть мультипроект как объект управления, как системный бизнес-процесс, а для этого необходим анализ его как системы взаимосвязанных свойств, структур, алгоритмов, уровней иерархии представления и управления и т.д. Решению данной задачи и посвящена эта статья, **целью** которой является разработка метода системного, наглядного и формализованного представления мультипроекта.

Для системного представления мультипроекта используем методику, предложенную В.М. Илюшко [1, 2]. В основу данной методики положены определенные последовательные действия: необходимо представить объект управления (в нашем случае – мультипроект) и систему управления (в нашем случае – корпорацию) в виде логической совокупности свойств (страт) системы, декомпозированных по уровням иерархии системы. Исходя из сложности объекта и системы управления, невозможности сразу описать их полно и достоверно – предложена система страт, регламентирующая последовательность накопления

знаний об объекте. Для описания нашего объекта предлагается выделить следующие страты: целевую (C), функциональную (Φ), организационной структуры (OC), инфологическую ($ИЛ$), алгоритмическую (A). Каждая из этих страт будет, в свою очередь, декомпозироваться (иерархически разделяться) со следующей степенью подчиненности уровней:

$$MultProj \rightarrow Proj \rightarrow UnProj \rightarrow GrTask \rightarrow Task ,$$

где $MultProj$ – мультипроект;

$Proj$ – составная часть мультипроекта – проект;

$UnProj$ – подпроект;

$GrTask$ – группа задач;

$Task$ – единичная задача.

Любые операции для представления, анализа, оптимизации мультипроекта как сложного объекта на сегодня трудно представить без использования компьютерных средств. В связи с этим следующим шагом к системному (стратифицированному, иерархическому) представлению мультипроекта является создание компьютерной системной модели, т.е. совокупности взаимосвязанных моделей по всем стратам представления объекта [13]. Процесс формирования компьютерной системной модели заключается в последовательном переходе от словесного описания к наглядному и формализованному представлению, а затем к компьютерному представлению моделей на различных стратах и уровнях мультипроекта. Данный подход к процессу формирования компьютерных имитационных моделей для сложных систем высказывался в работах Н.П. Бусленко [14], В.Н. Бусленко [15], И.В. Максимея [16].

В процессе формирования компьютерной системной модели мультипроекта важным вопросом есть выбор методов формализованного представления моделей. Причем наглядность представления и удобство работы с ними на всех уровнях и стратах представления также являются основными требованиями к методу и языку описания моделей. Поэтому для описания моделей можно выдвинуть следующие основные требования:

– наглядность в представлении модели, легкость построения компьютерными средствами;

– возможность проведения анализа, преобразований, оптимизации над моделями, в том числе компьютерными способами.

Анализ мультипроекта на различных стратах показал, что для представления страт (целевой, функциональной, организационной структуры, ресурсной, инфологической) необходимо сформировать структурные модели, а для алгоритмической страты – событийную модель. Структурные модели описывают состав (структуру) мультипроекта на определенной страте представления, а событийные – процессы (с учетом временного параметра, в отличие от структурных моделей), происходящие в мультипроекте. Структурные и событийные модели являются взаимосвязанными – одни являются прообразом для других, как по стратам, так и по уровням декомпозиции проектных решений, вследствие чего необходимо использовать единый способ представления структурных и событийных моделей.

Для описания структур, удовлетворяющих требованию наглядности, используется аппарат теории графов [17, 18]. Математический аппарат теории графов имеет достаточный набор теорем, аксиом, что помогает производить над графовыми моделями разнообразные операции и преобразования [19]. Для

описания событийных процессов предлагается использовать аппарат регулярных схем алгоритмов (РСА), впервые предложенный В.М. Глушковым [20], который позволяет наглядно, однозначно и строго описывать действия алгоритмов, осуществлять равносильные преобразования алгоритмов из любых языков описания в РСА и т.д. В своих работах В.М. Илюшко, на основании проведенного сравнительного анализа формализованных языков описания структур и алгоритмов, показал, что наиболее соответствующим средством для описания и преобразования моделей может служить модифицированный язык регулярных схем алгоритмов с построением на базе его регулярных схем системных моделей (РССМ) [21].

Таким образом, в соответствии с последующим процессом формирования компьютерной системной модели мультипроекта, для наглядного представления моделей применяется теория графов, для формализованного представления моделей используются РССМ, компьютерное представление моделей базируется на основных понятиях и принципах объектно-ориентированного программирования.

Как было сказано ранее, процессы реализации и управления мультипроектом в инвестиционно-строительной корпорации являются системными, поэтому с помощью методов декомпозиции и стратификации представим его в виде набора структурных и событийных моделей, используя теорию графов и язык РССМ.

Наглядное представление структурных моделей мультипроекта инвестиционно-строительной корпорации.

Наглядное представление структурных моделей мультипроекта осуществляется следующим образом.

Фиксируем элемент структуры на уровне *MultiProj* и производим его декомпозицию на элементы уровня *Proj* на основании словесного описания, предоставляемого руководителем мультипроекта: $MultiProj = \Omega_{\dots i} Proj$. Присваиваем каждому элементу структуры уровня *Proj* значение узла графа α_i . Строим систему узлов графа $\Omega \alpha_i$. Получаем графовое представление структурной модели на уровне проектов (рис. 1): $MultiProj = \Omega_{\dots i} Proj \rightarrow \Gamma_{\dots i} Proj$.

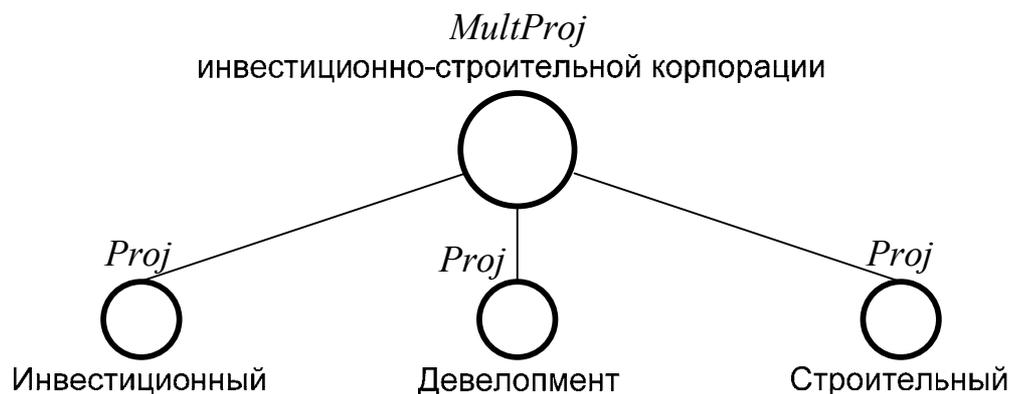


Рис. 1. Графовое представление структурной модели на уровне проектов

Фиксируем элементы структуры на уровне *Proj* и производим их декомпозицию на элементы уровня *UnProj* на основании словесных описаний, предоставляемых руководителями проектов: $Proj = \Omega_{\dots i} UnProj$. Присваиваем

каждому элементу структуры уровня $UnProj$ значение узла графа α_i . Строим систему узлов графа $\Omega\alpha_i$. Получаем графовое представление структурной модели на уровне подпроектов: $Proj = \Omega\dots_i UnProj \rightarrow \Gamma\dots_i UnProj$.

Вставляя подграфы уровня $UnProj$ в граф верхнего уровня $Proj$, получаем графовое представление структурной модели на уровне подпроектов (рис. 2).

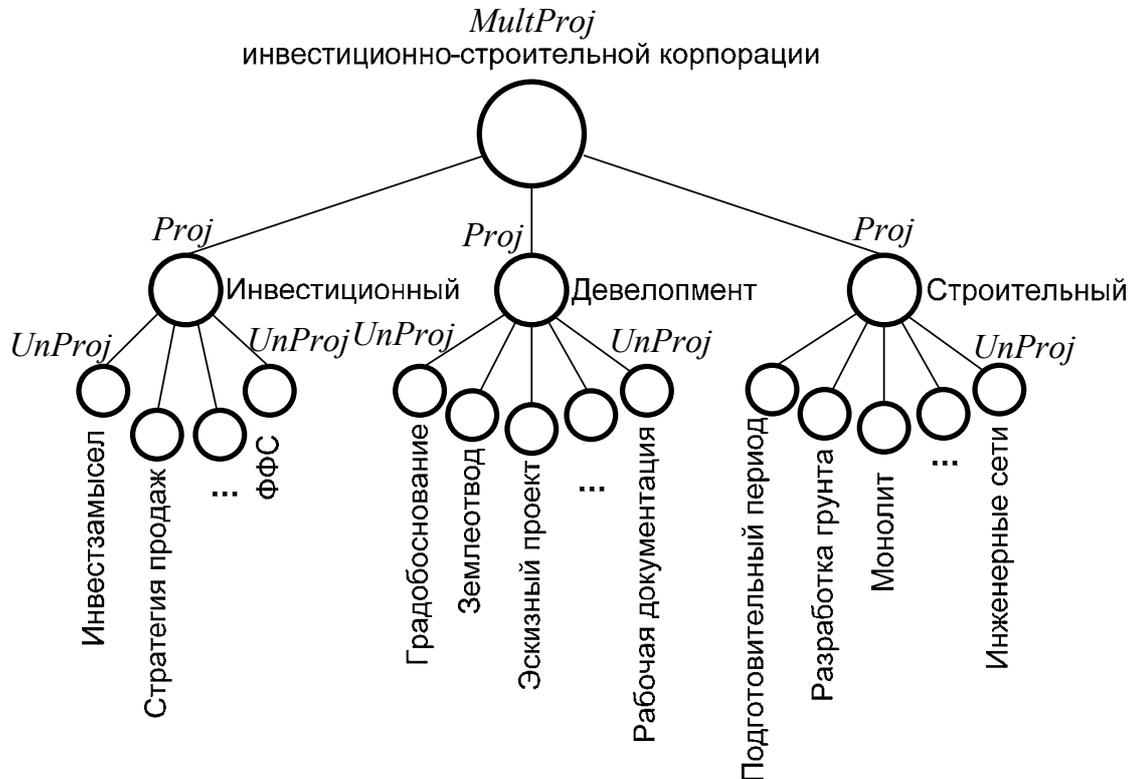


Рис. 2. Графовое представление структурной модели на уровне подпроектов

Фиксируем элементы структуры на уровне $UnProj$ и производим их декомпозицию на элементы уровня $GrTask$ на основании словесных описаний, предоставляемых руководителями подпроектов: $UnProj = \Omega\dots_i GrTask$. Присваиваем каждому элементу структуры уровня $GrTask$ значение узла графа α_i . Строим систему узлов графа $\Omega\alpha_i$. Получаем графовое представление структурной модели на уровне групп элементов: $UnProj = \Omega\dots_i GrTask \rightarrow \Gamma\dots_i GrTask$.

Вставляя подграфы уровня $GrTask$ в граф уровня $UnProj$, получаем графовое представление структурной модели на уровне групп задач (рис. 3).

Фиксируем элементы структуры на уровне $GrTask$ и производим их декомпозицию на элементы уровня $Task$ на основании словесных описаний, предоставляемых руководителями групп задач: $GrTask = \Omega\dots_i Task$. Присваиваем каждому элементу структуры уровня $Task$ значение узла графа α_i . Строим систему узлов графа $\Omega\alpha_i$. Получаем графовое представление структурной модели на уровне задач: $GrTask = \Omega\dots_i Task \rightarrow \Gamma\dots_i Task$.

Вставляя подграфы уровня $Task$ в граф верхнего уровня $GrTask$, получаем графовое представление структурной модели на уровне задач (рис. 4).

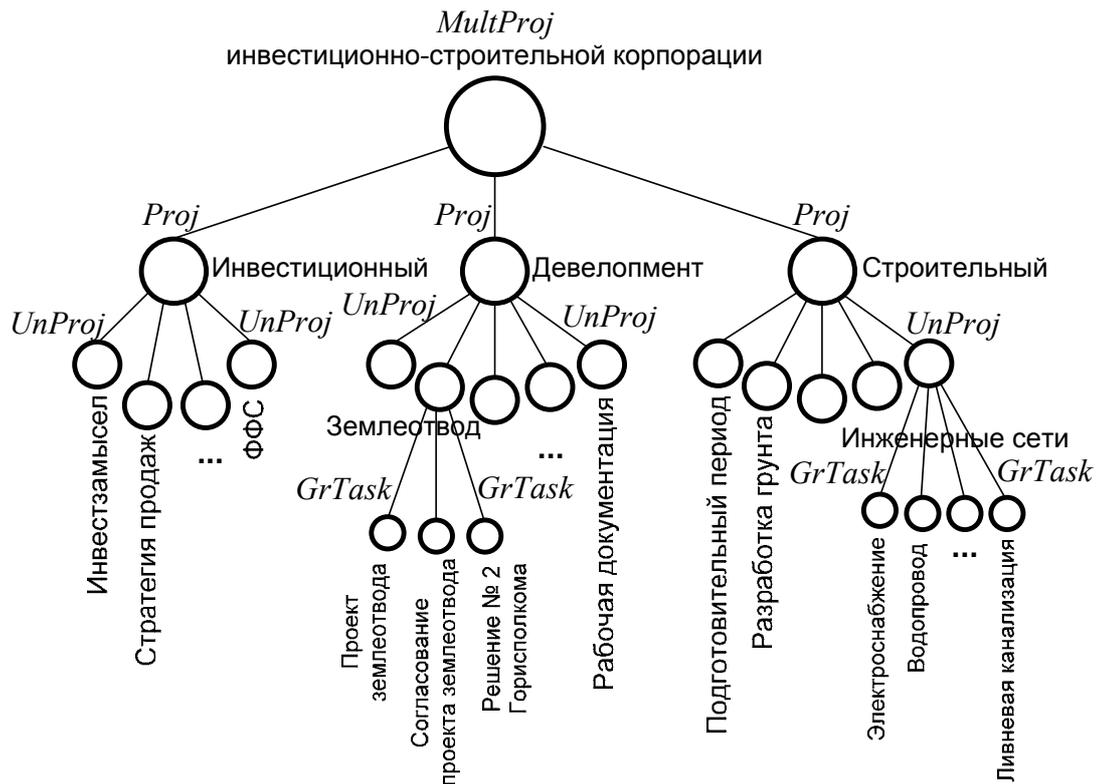


Рис. 3. Графовое представление структурной модели на уровне групп задач

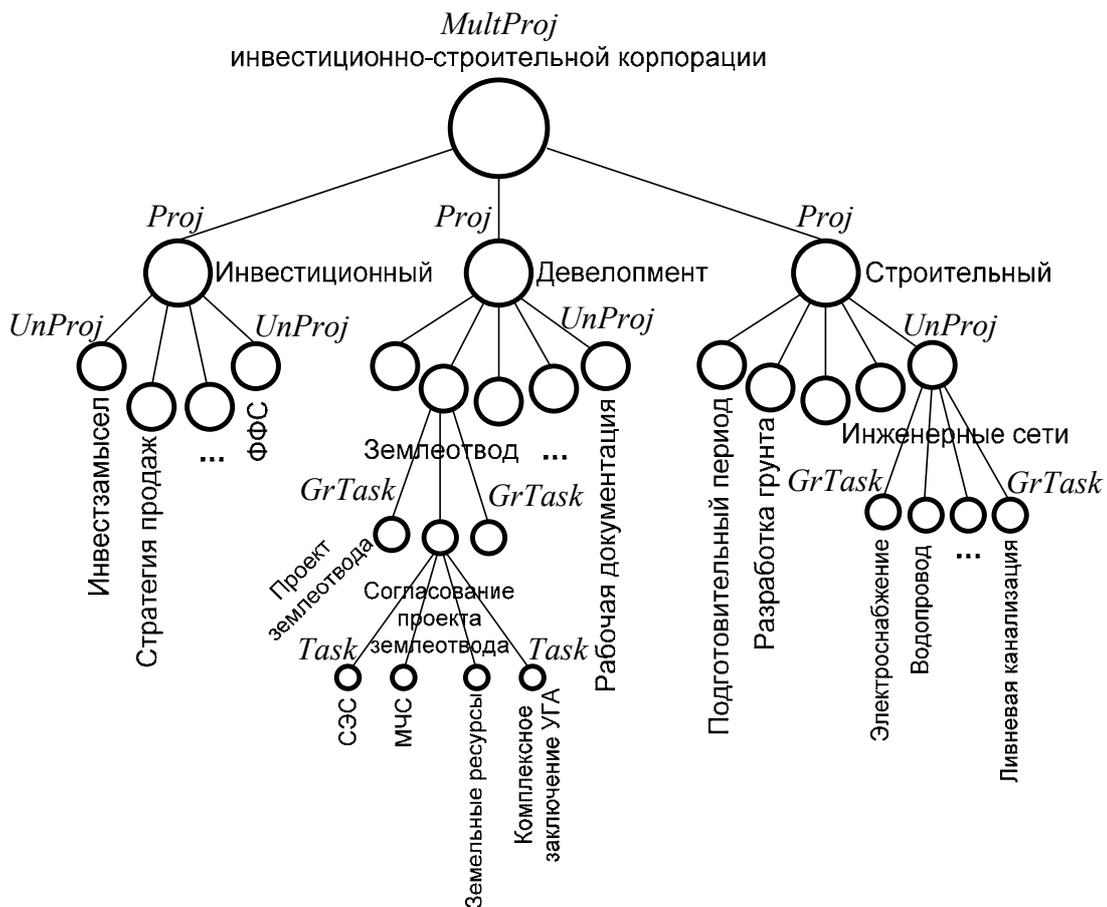


Рис. 4. Графовое представление структурной модели на уровне задач

Наглядное представление событийных моделей мультипроекта.
 Данное представление осуществляется следующим образом.

Фиксируем событие на уровне *MultiProj* и производим его декомпозицию на события уровня *Proj* на основании словесного описания, предоставляемого руководителем мультипроекта: $MultiProj = \Omega \dots_i Proj$. Присваиваем каждому событию уровня *Proj* значение узла графа α_i . Строим систему узлов графа $\Omega \alpha_i$.

Руководитель мультипроекта должен определить связи между событиями уровня *Proj*, т.е. последовательность их выполнения, начиная с первого события $Proj_1$. Возможны следующие варианты:

- последовательное выполнение элементов: $Proj_i \rightarrow Proj_{i+1}$;
- параллельное выполнение элементов: $Proj_i \begin{matrix} \rightarrow Proj_{i+1} \\ \rightarrow Proj_{i+2} \end{matrix}$;
- выполнение элементов по условию x_k : $Proj_i \rightarrow \begin{matrix} x_k^0 \rightarrow Proj_{i+1} \\ x_k^1 \rightarrow Proj_{i+2} \end{matrix}$.

Наносим связи между событиями уровня *Proj* в виде ребер графа на соответствующие узлы графа: $\alpha_i \rightarrow \alpha_j$. Получаем графовое представление событийной модели на уровне проектов (рис. 5): $MultiProj = \Omega \dots_i Proj \rightarrow \Gamma \dots_i Proj$.

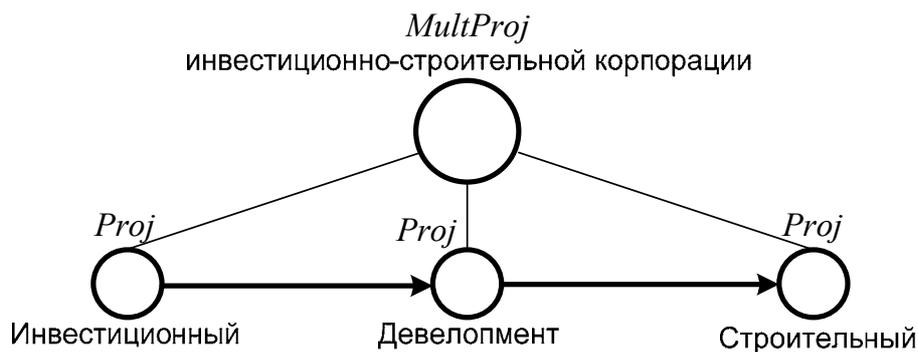


Рис. 5. Графовое представление событийной модели на уровне *Proj*

Фиксируем события на уровне *Proj* и производим их декомпозицию на события уровня *UnProj* на основании словесных описаний, предоставляемых руководителями проектов: $Proj = \Omega \dots_i UnProj$. Присваиваем каждому событию уровня *UnProj* значение узла графа α_i . Строим систему узлов графа $\Omega \alpha_i$.

Руководители проектов должны определить связи между событиями уровня *UnProj*, т.е. последовательность их выполнения, начиная с первого элемента $UnProj_1$.

Наносим связи между событиями уровня *UnProj* в виде ребер графа на соответствующие узлы графа: $\alpha_i \rightarrow \alpha_j$. Получаем графовое представление событийной модели на уровне подпроектов: $Proj = \Omega \dots_i UnProj \rightarrow \Gamma \dots_i UnProj$. Вставляя подграфы событий уровня *UnProj* в граф верхнего уровня *Proj*, получаем графовое представление событийной модели в базисе подпроектов (рис. 6).

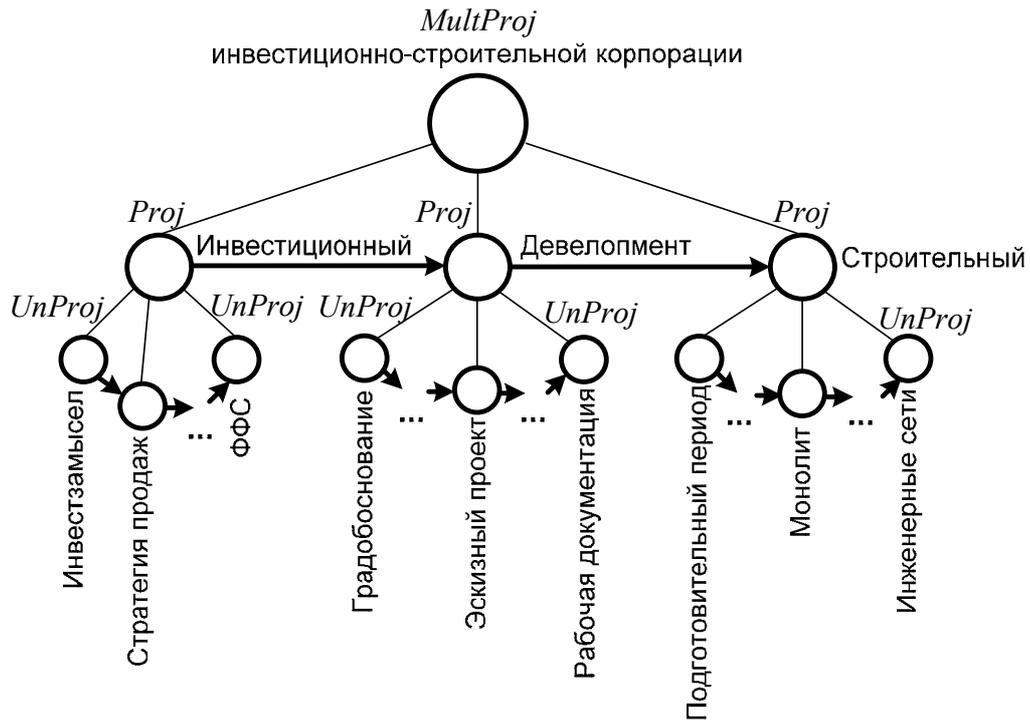


Рис. 6. Графовое представление событийной модели на уровне *UnProj*

Фиксируем события на уровне *UnProj* и производим их декомпозицию на события уровня *GrTask* на основании словесных описаний, предоставляемых руководителями подпроектов: $UnProj = \Omega \dots_i GrTask$. Присваиваем каждому событию уровня *GrTask* значение узла графа α_i . Строим систему узлов графа $\Omega \alpha_i$.

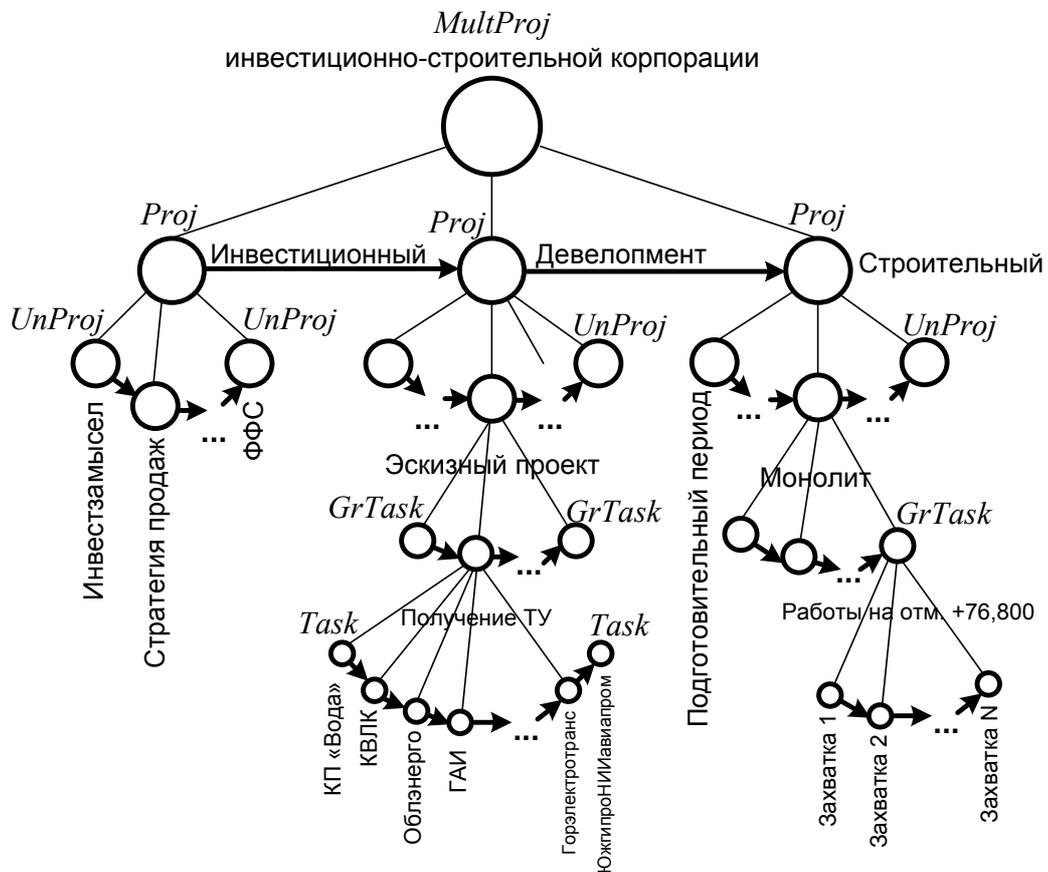
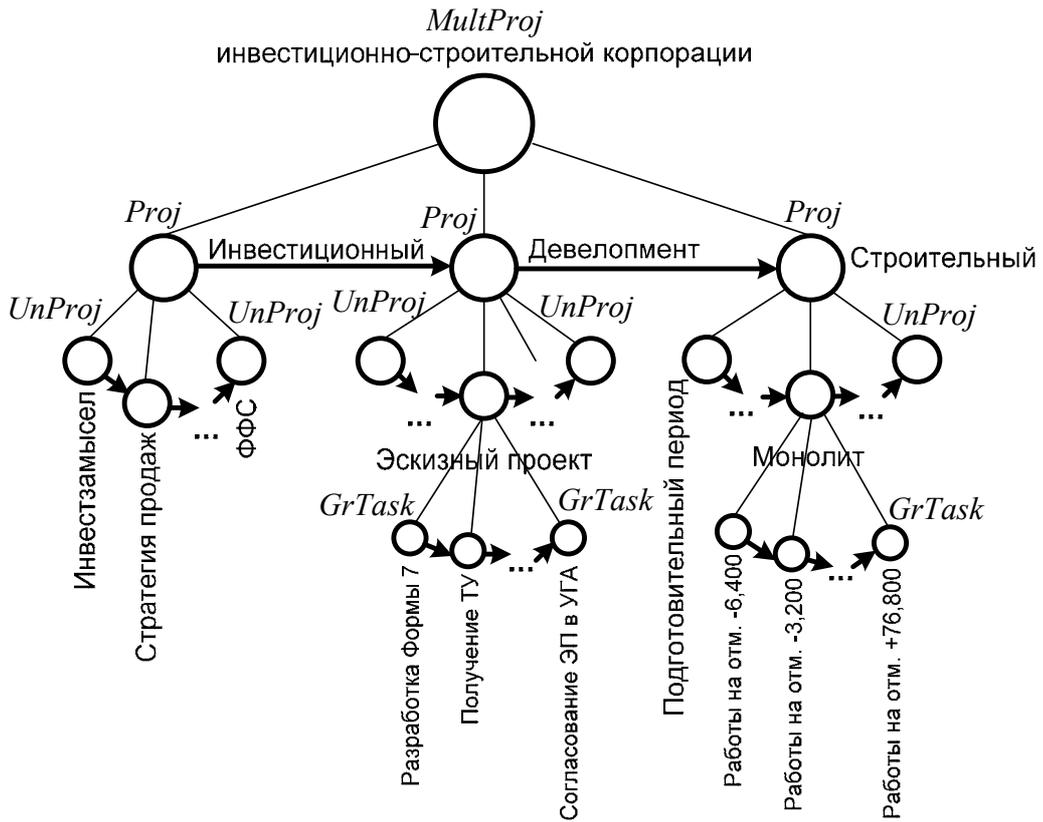
Руководители подпроектов должны определить связи между событиями уровня *GrTask*, т.е. последовательность их выполнения, начиная с первого элемента $GrTask_1$.

Наносим связи между элементами уровня *GrTask* в виде ребер графа на соответствующие узлы графа: $\alpha_i \rightarrow \alpha_j$. Получаем графовое представление событийной модели на уровне групп задач: $UnProj = \Omega \dots_i GrTask \rightarrow \Gamma \dots_i GrTask$. Вставляя подграфы уровня *GrTask* в граф уровня *UnProj*, получаем графовое представление событийной модели в базе групп задач (рис. 7).

Фиксируем события на уровне *GrTask* и производим их декомпозицию на события уровня *Task* на основании словесных описаний, предоставляемых руководителями групп задач: $GrTask = \Omega \dots_i Task$. Присваиваем каждому событию уровня *Task* значение узла графа α_i . Строим систему узлов графа $\Omega \alpha_i$.

Руководители групп задач должны определить связи между событиями уровня *Task* т.е. последовательность их выполнения, начиная с первого элемента $Task_1$.

Наносим связи между элементами уровня *Task* в виде ребер графа на соответствующие узлы графа: $\alpha_i \rightarrow \alpha_j$. Получаем графовое представление событийной модели на уровне задач: $GrTask = \Omega \dots_i Task \rightarrow \Gamma \dots_i Task$. Вставляя подграфы уровня *Task* в граф верхнего уровня *GrTask*, получаем графовое представление событийной модели в базе задач (рис. 8).



Таким образом, используя предложенные правила, формируется наглядное представление системных стратифицированных моделей мультипроекта для последующего анализа компьютерными методами на любом уровне декомпозиции.

Формализованное представление структурных и событийных моделей мультипроекта инвестиционно-строительной корпорации.

Формирование формализованного представления структурных и событийных моделей мультипроекта осуществляется «сверху вниз», с помощью языка РССТМ. Структурные модели мультипроекта в РССТМ запишутся в виде

$$R = f(y_i, e, \otimes, \hat{y}).$$

Формализованное представление структурных моделей мультипроекта производится следующим образом.

Фиксируем графовое представление структурной модели на уровне *Proj*. Ставим в соответствие каждому узлу графа *Proj_i* оператор y_i из РССТМ. Так как структурные модели описывают состав элементов мультипроекта на определенной страте представления, то для связи между операторами y_i будем использовать базовую операцию конъюнкции \hat{y} . Таким образом, формализованное представление структурной модели на уровне проектов записываем через операторы РССТМ в виде

$$R \text{ MultProj} = [y_1^{Proj} \wedge y_2^{Proj} \wedge y_3^{Proj}].$$

Фиксируем графовое представление структурной модели на уровне *UnProj*. Ставим в соответствие каждому узлу графа *UnProj_i* оператор y_i из РССТМ, а для каждой связи между операторами y_i будем использовать базовую операцию конъюнкции \hat{y} . Таким образом, формализованное представление структурной модели на уровне подпроектов записываем через операторы РССТМ в виде

$$R \text{ Proj} = [y_1^{UnProj} \wedge \dots \wedge y_n^{UnProj}].$$

Вставляя алгоритмы уровня *UnProj* в алгоритм верхнего уровня *Proj*, получаем формализованное представление структурной модели в базисе подпроектов:

$$R \text{ MultProj} = [y_1^{Proj} \wedge y_2^{Proj} \wedge y_3^{Proj}] = [[y_1^{UnProj} \wedge \dots \wedge y_n^{UnProj}] \wedge \dots \wedge [y_1^{UnProj} \wedge \dots \wedge y_n^{UnProj}]].$$

Фиксируем графовое представление структурной модели на уровне *GrTask*. Ставим в соответствие каждому узлу графа *GrTask_i* оператор y_i из РССТМ, а для каждой связи между операторами y_i будем использовать базовую операцию конъюнкции \hat{y} . Таким образом, формализованное представление структурной модели на уровне групп задач записываем через операторы РССТМ в виде

$$R \text{ UnProj} = [y_1^{GrTask} \wedge \dots \wedge y_n^{GrTask}].$$

Вставляя алгоритмы уровня *GrTask* в алгоритм верхнего уровня *UnProj*, получаем формализованное представление структурной модели в базисе групп задач:

$$R \text{ MultProj} = [y_1^{Proj} \wedge y_2^{Proj} \wedge y_3^{Proj}] = [[y_1^{UnProj} \wedge \dots \wedge y_n^{UnProj}] \wedge \dots \wedge [y_1^{UnProj} \wedge \dots \wedge y_n^{UnProj}]] = [[[[y_1^{Gr} \wedge \dots \wedge y_n^{GrTask} \dots]] \wedge \dots \wedge [y_1^{GrTask} \wedge \dots]] \wedge \dots \wedge [[y_1^{Gr} \wedge \dots \wedge y_n^{GrTask} \dots]] \wedge \dots \wedge [y_1^{GrTask} \wedge \dots]].$$

Фиксируем графовое представление структурной модели на уровне *Task*. Ставим в соответствие каждому узлу графа $Task_i$ оператор y_i из PCCM, а для связи между операторами y_i будем использовать базовую операцию конъюнкции \hat{y} . Таким образом, формализованное представление структурной модели на уровне задач записываем через операторы PCCM в виде:

$$R \text{ GrTask} = [y_1^{Task} \wedge \dots \wedge y_n^{Task}].$$

Вставляя алгоритмы уровня *Task* в алгоритм верхнего уровня *GrTask*, получим формализованное представление структурной модели в базисе задач:

$$\begin{aligned} R \text{ MultProj} &= [y_1^{Proj} \wedge y_2^{Proj} \wedge y_3^{Proj}] = [[y_1^{UnProj} \wedge \dots \wedge y_n^{UnProj}] \wedge \dots \wedge [y_1^{UnProj} \wedge \dots \wedge y_n^{UnProj}]] = \\ &= [[[[y_1^{GrTask} \wedge \dots \wedge y_n^{GrTask} \dots] \wedge \dots \wedge [y_1^{GrTask} \wedge \dots]] \wedge \dots \wedge [[y_1^{GrTask} \wedge \dots \wedge y_n^{GrTask} \dots] \wedge \dots \wedge [y_1^{GrTask} \wedge \dots]]]] = \\ &= [[[[[[y_1^{Task} \wedge \dots \wedge y_n^{Task}] \wedge \dots \wedge [y_1^{Task} \wedge \dots]] \wedge \dots \wedge [[y_1^{Task} \wedge \dots \wedge y_n^{Task}] \wedge \dots \wedge [y_1^{Task} \wedge \dots]]]] \wedge \\ &\wedge [[[[y_1^{Task} \wedge \dots \wedge y_n^{Task}] \wedge \dots \wedge [y_1^{Task} \wedge \dots]] \wedge \dots \wedge [[y_1^{Task} \wedge \dots \wedge y_n^{Task}] \wedge \dots \wedge [y_1^{Task} \wedge \dots]]]]]. \end{aligned}$$

Событийные модели мультипроекта в PCCM записываем в виде

$$R = f(y_i, x_k, e, \otimes, 1, 0, \dot{y}, \hat{y}, \check{y}).$$

Формализованное представление событийных моделей мультипроекта производится следующим образом.

Фиксируем графовое представление событийной модели на уровне *Proj*. Ставим в соответствие каждому узлу графа $Proj_i$ оператор y_i из PCCM. Так как событийные модели описывают процессы происходящие в мультипроекте, то для связи между операторами y_i , начиная с оператора y_1 , будем использовать базовые операции:

- умножение \dot{y} – последовательное выполнение операторов;
- конъюнкцию \hat{y} – параллельное выполнение операторов;
- дизъюнкцию \check{y} – условное выполнение операторов.

Таким образом, формализованное представление событийной модели на уровне проектов запишем, используя операторы PCCM, в виде

$$R \text{ MultProj} = [y_1^{Proj} \cdot \dots \cdot y_n^{Proj}].$$

Фиксируем графовое представление событийной модели на уровне *UnProj*. Ставим в соответствие каждому узлу графа $UnProj_i$ оператор y_i из PCCM, а для связей между операторами y_i , начиная с оператора y_1 , будем использовать базовые операции \dot{y} , \hat{y} , \check{y} . Таким образом, формализованное представление событийной модели на уровне подпроектов запишем, используя операторы PCCM, в виде

$$R \text{ Proj} = [y_1^{UnProj} \cdot \dots \cdot y_n^{UnProj}].$$

Вставляя алгоритмы уровня *UnProj* в алгоритм верхнего уровня *Proj*, получим формализованное представление событийной модели в базисе подпроектов:

$$R \text{ MultProj} = [y_1^{Proj} \cdot \dots \cdot y_n^{Proj}] = [[y_1^{UnProj} \cdot \dots \cdot y_n^{UnProj}] \cdot \dots \cdot [y_1^{UnProj} \cdot \dots \cdot y_n^{UnProj}]].$$

Фиксируем графовое представление событийной модели на уровне *GrTask*. Ставим в соответствие каждому узлу графа *GrTask_i* оператор y_i из РССМ, а для связей между операторами y_i , начиная с оператора y_1 , будем использовать базовые операции \dot{y} , \hat{y} , \check{y} . Таким образом, формализованное представление событийной модели на уровне групп задач запишем, используя операторы РССМ, в виде

$$R \text{ UnProj} = [y_1^{GrTask} \dots y_n^{GrTask}].$$

Вставляя алгоритмы уровня *GrTask* в алгоритм верхнего уровня *UnProj*, получим формализованное представление событийной модели в базе групп задач:

$$\begin{aligned} R \text{ MultProj} &= [y_1^{Proj} \cdot y_2^{Proj} \cdot y_3^{Proj}] = [[y_1^{UnProj} \dots y_n^{UnProj}] \dots [y_1^{UnProj} \dots y_n^{UnProj}]] = \\ &= [[[[y_1^{GrTask} \dots y_n^{GrTask} \dots] \dots [y_1^{GrTask} \dots]] \dots [y_1^{GrTask} \dots y_n^{GrTask} \dots] \dots [y_1^{GrTask} \dots]]]. \end{aligned}$$

Фиксируем графовое представление событийной модели на уровне *Task*. Ставим в соответствие каждому узлу графа *Task_i* оператор y_i из РССМ, а для связей между операторами y_i , начиная с оператора y_1 , будем использовать базовые операции \dot{y} , \hat{y} , \check{y} . Таким образом, формализованное представление событийной модели на уровне задач запишем, используя операторы РССМ, в виде

$$R \text{ GrTask} = [y_1^{Task} \dots y_n^{Task}].$$

Вставляя алгоритмы уровня *Task* в алгоритм верхнего уровня *GrTask*, получим формализованное представление событийной модели в базе элементов:

$$\begin{aligned} R \text{ MultProj} &= [y_1^{Proj} \cdot y_2^{Proj} \cdot y_3^{Proj}] = [[y_1^{UnProj} \dots y_n^{UnProj}] \dots [y_1^{UnProj} \dots y_n^{UnProj}]] = \\ &= [[[[[y_1^{GrTask} \dots y_n^{GrTask} \dots] \dots [y_1^{GrTask} \dots]] \dots [y_1^{GrTask} \dots y_n^{GrTask} \dots] \dots [y_1^{GrTask} \dots]]] = \\ &= [[[[[[y_1^{Task} \dots y_n^{Task} \dots] \dots [y_1^{Task} \dots]] \dots [y_1^{Task} \dots y_n^{Task} \dots] \dots [y_1^{Task} \dots]]] \times \dots \\ &\dots \times [[[[y_1^{Task} \dots y_n^{Task} \dots] \dots [y_1^{Task} \dots]] \dots [y_1^{Task} \dots y_n^{Task} \dots] \dots [y_1^{Task} \dots]]]]. \end{aligned}$$

Используя предложенные правила, формируем формализованное представление системных стратифицированных моделей мультипроекта, которое можно анализировать компьютерными методами на любом уровне декомпозиции.

Таким образом, в статье предложен метод наглядного и формализованного представления мультипроектов инвестиционно-строительной корпорации. Данный метод прошел практическую апробацию на реальных мультипроектах и продемонстрировал реальную эффективность по сравнению с другими методами.

Список литературы

1. Илюшко В.М. Системный анализ проектных решений разработчика по проектированию сложных систем // Управление и связь. – Х.: НАНУ ПАНИ «ХВУ». – 1996. – С. 35 – 40.

2. Илюшко В.М., Федорович О.Е., Губка С.А., Коваленко В.И. Проекты сложных систем. Системные средства моделирования, анализа и контроля // Proceeding sixth international conference "New Leading - adge technologies in machine building". – Rybachie, Ukraine. – 1997. – P.139 – 144.
3. Емад А.Р., Мелешенко С.Ю., Махер Ниаф Х. Хилал. Модели представимости формальных языков в сложных проектах // Открытые информационные и компьютерные интегрированные технологии. – Х.: НАКУ «ХАИ». – 2004. – Вып. 22. – С. 157 – 165.
4. Коллинз Г., Блей Дж. Структурные методы разработки систем: от стратегического планирования до тестирования. – М.: Финансы и статистика, 1986. – 264 с.
5. Акберин Р.З., Кибанов А.Я. Совершенствование структуры функций и экономических взаимоотношений управленческих подразделений предприятий при разных формах хозяйствования: Учеб. пособие. – М.: ГАУ, 2006. – 258 с.
6. Уотерман Р. Фактор обновления: как сохраняют конкурентоспособность лучшие компании: Пер. с англ. – М.: Прогресс, 1988. – 368 с.
7. Бир С. Мозг фирмы: Пер. с англ. – М.: Радио и связь, 1993. – 416 с.
8. Чудаков А.Д. Логистика. – М.: Изд-во РДЛ, 2001. – 480 с.
9. Илюшко В.М., Некрасов И.Б. Мультипроект, как субъект управления корпорации // Радиоэлектронные компьютерные системы. – 2005. – Вып. 4(12). – С. 133 – 136.
10. Реинжиниринг бизнес-процессов. Полный курс MBA / Н.М. Абдикеев, Т.П. Данько, С.В. Ильдеменов, А.Д. Киселев. – М.: Эксмо, 2007. – 592 с.
11. Хаммер Майкл, Чампи Джеймс. Реинжиниринг корпорации: Манифест революции в бизнесе: Пер. с англ. Ю.Е. Корниловича. – М.: Манн, Иванов и Фербер, 2006. – 287 с.
12. Гаджинский А.М. Логистика: Учебник для высших и средних специальных учебных заведений – 2-е изд. – М.: Информационно-внедренческий центр «Маркетинг», 1999. – 160 с.
13. Латкин М.А. Формирование компьютерной модели представления систем управления // Авіаційно-космічна техніка та технологія. – Х.: ХАІ. – 1998. – Вип. 6. – С.261 – 264.
14. Бусленко Н.П. Моделирование сложных систем. – М.: Наука, 1978. – 400 с.
15. Бусленко В.Н. Автоматизация имитационного моделирования сложных систем. – М.: Наука, 1977. – 240 с.
16. Максимей И.В. Математическое моделирование больших систем. – М.: Высш. шк., 1985. – 119 с.
17. Басакер Р., Саати Т. Конечные графы и сети: Пер. с англ. – М.: Наука, 1974. – 366 с.
18. Харарри Ф., Палмер Э. Перечисление графов: Пер. с англ. – М.: Мир, 1977. – 324 с.
19. Вунш Г. Теория систем. – М.: Сов. радио, 1978. – 288 с.
20. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра, языки программирования. – К.: Наук. думка, 1974. – 327 с.
21. Смирнов О.Л., Падалко С.Н. САПР: формирование и функционирование проектных модулей – М.: Машиностроение, 1987. – 272 с.