

УДК 004.78; 681.5

М.В. ПОТАПОВА

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

СИСТЕМНАЯ МОДЕЛЬ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ ПРОГРАММНЫХ ПРОЕКТОВ

Проведён анализ основных особенностей программных проектов. Описан процесс работ по созданию и выполнению программных проектов. Выполнен обзор требований, предъявляемых к системам планирования программных проектов. Рассмотрены теоретические подходы к календарному планированию программных проектов. Описана системная модель календарного планирования выполнения программных проектов ограничении. Доказана необходимость применения методов распространения ограничений и математических методов, ориентированных на обработку неточной и неполной информации. Представлен процесс алгоритмизации прототипа программного обеспечения для планирования программных проектов.

Ключевые слова: системная модель, программный проект, неопределенность, ограничения, календарное планирование, управление проектами.

Введение

Эффективное управление программным проектом напрямую зависит от правильного планирования работ, необходимых для его выполнения. В процессе планирования проекта определяются процессы, этапы и полученные на каждом из них результаты, которые должны привести к выполнению проекта. Процесс планирования начинается с определения проектных ограничений (временные ограничения, возможности наличного персонала, бюджетные ограничения и т.д.). Эти ограничения должны определяться параллельно с оцениванием проектных параметров, таких как структура и размер проекта, а также распределение функций среди будущих исполнителей. Затем определяются этапы разработки и то, какие результаты (документация, прототипы, подсистемы или версии программного продукта) должны быть получены по окончании этих этапов. Далее начинается циклическая часть планирования. Сначала разрабатывается график работ по выполнению проекта, затем при возникновении расхождений между реальным и плановым ходом работ возможен пересмотр первоначальных оценок параметров проекта. Это, в свою очередь, может привести к изменению графика работ. Если в результате этих изменений нарушаются сроки завершения проекта, должны быть пересмотрены проектные ограничения.

План проекта должен четко показывать ресурсы, необходимые для реализации проекта, разделение работ на этапы и временной график выполнения этих этапов. В процессе составления графика весь массив работ, необходимых для реализации проекта, разбивается на отдельные этапы и оценивается время, тре-

бующее для выполнения каждого этапа. Обычно многие этапы выполняются параллельно. График работ должен предусматривать это и распределять ресурсы между ними оптимальным образом. Нехватка ресурсов для выполнения какого-либо критического этапа – частая причина задержки выполнения всего проекта.

Кроме временных затрат, руководитель также должен рассчитать другие виды ресурсов, необходимых для успешного выполнения каждого этапа. Особый вид ресурсов – это команда разработчиков, привлеченная к выполнению проекта. Другими видами ресурсов могут быть необходимое свободное дисковое пространство на сервере, время использования какого-либо специального оборудования, либо средства на командировочные расходы персонала, работающего над проектом.

Целью настоящей работы является описание системной модели календарного планирования программных проектов, применение которой совместно с методами распространения ограничений обеспечит повышение эффективности процессов управления программными проектами.

1. Постановка задачи планирования программного проекта в виде задачи удовлетворения ограничений

Исходные допущения. В наиболее общей постановке, задача оптимизации проектного плана включает в себя формирование множеств исполнителей, а затем поиск такого их назначения на проектные роли в конкретных задачах, который обеспе-

чивает максимизацию интегрального показателя эффективности проекта.

Таким образом, оптимизация проектного плана подразумевает последовательное решение задач формирования состава команды, распределения функций (ролей) в команде, а также распределение объемов работ с оценкой эффективности. Перечисленные задачи взаимосвязаны и решаются «циклически».

При распределении функций и объемов работ предполагается, что зафиксирован состав проектной команды и через подграф модели компетенций известны оценки готовности всех членов команды к исполнению определенных ролей. Данная задача относится к классу задач о назначении, в которых критерием оптимальности является минимум суммарных затрат на заработную плату всех членов команды.

Формирование состава команды относится к классу «задач о ранце» дискретной оптимизации и обладает достаточно высокой комбинаторной сложностью. Ее суть – выбрать такой состав участников проектной команды, чтобы обеспечить максимальную эффективность процесса разработки. На допустимые составы команды могут дополнительно накладываться как требования обязательного включения в нее отдельных сотрудников, так и запреты на совместное включение тех или иных людей в одну группу, поскольку именно эти ограничения позволяют обеспечить синергетический эффект командной работы.

1.1. Многокритериальность задачи оценки эффективности проекта

Конкретизация общей постановки оптимизационной задачи требует уточнения понятия «эффективность проекта», то есть его результативности.

Эффективность проекта – категория, характеризующая соотношение экономических, социальных и научно-технических результатов проекта с затратами на их достижение [1].

Экономическая эффективность – это результативность экономической системы, выражающаяся в отношении полезных конечных результатов ее функционирования к затраченным ресурсам.

Эффективность программного обеспечения – отношение уровня услуг, предоставляемых программным продуктом пользователю при заданных условиях, к объему используемых ресурсов.

Только наличие нескольких показателей эффективности позволяет в наибольшей мере учесть все свойства системы.

Таким образом, и задача оценки плана программного проекта, и задача синтеза такого плана являются многокритериальными [2].

Принимая во внимание перечисленные выше противоречивые условия, можно назвать три взаимосвязанных интегральных показателя, определяющих эффективность реализации программного проекта:

- выходной эффект (Q),
- время реализации проекта (B),
- затраты на реализацию проекта (З).

Таким образом, предполагается, что эффективность – это некоторый функционал от выходного эффекта, затрат и времени на реализацию проекта, а его конкретное представление определяется экспертным оцениванием проекта.

Общепроектные ограничения. Перечислим дополнительные общепроектные ограничения, которые, как правило, должны выполняться в любом проекте:

1. Ограничение на фонд заработной платы:

$$\sum_i \left(\text{effort}(t_i) \sum_j (\gamma_{t_i, \text{exc}_j} \cdot Z_{\text{exc}_j}) \right) \leq Z_{\Sigma}, \quad (1)$$

где $\text{effort}(t_i)$ – трудоемкость выполнения работы t_i исполнителями;

$\gamma_{t_i, \text{exc}_j}$ – коэффициент, характеризующий относительный объем работы t_i , выполненной j -м исполнителем exc_j , такой, что:

$$\forall \text{exc}_j \in E, \sum_i \gamma_{t_i, \text{exc}_j} = 1, \quad (2)$$

где Z_{exc_j} – заработная плата j -го исполнителя;

Z_{Σ} – предельная величина фонда заработной платы

2. Ограничение на общую трудоемкость работ

$$\sum_i \text{effort}(t_i) \leq T_{\Sigma}, \quad (3)$$

где T_{Σ} – предельно допустимая общая трудоемкость работ.

3. Ограничения на средний уровень занятости исполнителей в проекте предполагают, что могут быть назначены нижняя $\underline{\beta}$ и верхняя $\bar{\beta}$ границы, характеризующие среднюю занятость исполнителей:

$$\underline{\beta} \leq \frac{\sum_i \text{effort}(t_i)}{\|E\| \cdot \tau_{T_{\Sigma}}} \leq \bar{\beta}, \quad (4)$$

где $\tau_{T_{\Sigma}}$ – общая продолжительность работ;

4. Ограничение на степень участия j -го исполнителя в конкретных работах проекта устанавливает предельно допустимые значения относительной занятости любого человека в проектных работах:

$$\forall j, \underline{\beta}_j \leq \sum_i \frac{\text{effort}(t_i, \text{exc}_j)}{\tau_{T_{\Sigma}}} \leq \bar{\beta}_j. \quad (5)$$

5. Ограничение на временное участие j-го исполнителя в проекте предполагает, что может быть задан некоторый период времени $t = (t_s \dots t_f)$, в течение которого исполнитель будет задействован в проекте:

$$\forall i, \forall j, \begin{cases} t_s \leq \min_i (\tau_{start_{i,j}}); \\ t_f \geq \max_i (\tau_{finish_{i,j}}), \end{cases} \quad (6)$$

где $\tau_{start_{i,j}}$, $\tau_{finish_{i,j}}$ – время начала, окончания работ, в которых планируется участие j-го исполнителя.

6. Ограничение на максимальную продолжительность интервала времени $\tau_{free_{aii}}$, в течение которого простаивание j-го исполнителя (отсутствие загрузки проектными работами) считается допустимым:

$$\forall j, \Delta \tau_j = \max_r (\overline{\tau_{free_{r,j}}} - \underline{\tau_{free_{r,j}}}) \leq \tau_{free_{aii}}, \quad (7)$$

где $\overline{\tau_{free_{r,j}}}$, $\underline{\tau_{free_{r,j}}}$ – время окончания, начала r-того интервала времени, когда j-го исполнитель не принимает участия в проектных работах, то есть

$$\forall \tau_j, \overline{\tau_{free_{r,j}}} \leq \tau_j < \underline{\tau_{free_{r,j}}}, \neg \exists i, \tau_{start_{i,j}} \leq \tau_j < \tau_{finish_{i,j}}.$$

7. Ограничение на планирование работ, предотвращающее появление дефектов плана работ, типа «бутылочное горло», когда существует работа, начало выполнения которой задерживается в течение значительного промежутка времени из-за неготовности незначительной доли ее входных продуктов:

$$\exists T_i \in T, \exists \Delta \tau_j \geq \Delta \tau_{aii}, \frac{\|WP_{in_i, \Delta \tau_j}\|}{\|WP_{in_i}\|} \leq \sigma_{i, \Delta \tau_j}, \quad (8)$$

где T_i – работа;

$\Delta \tau_{aii}$ – максимально допустимый интервал времени ожидания недостающих для начала T_i работы входных продуктов $WP_{in_i, \Delta \tau_j}$;

$\sigma_{i, \Delta \tau_j}$ – минимально допустимая доля входных продуктов, которые необходимы для начала T_i работы.

2. Использование системной модели в задаче планирования программного проекта на основе программирования в ограничениях

Способ представления недоопределенного значения влияет как на качество полученных результатов, так и на вид ограничений, связывающих это значение.

В зависимости от характера представляемой информации недоопределенные значения могут быть представлены в виде целочисленных и вещественных интервалов, множеств, перечислений и других, более специальных, конструкций [3].

Областью определения переменной называется множество всех возможных значений, которые могут быть присвоены этой переменной. В дальнейшем мы будем говорить просто "область переменной" и обозначать эту область переменной x через Dx .

Меткой называется пара переменная-значение, которая определяет присваивание некоторого значения переменной. Метка, определяющая присваивание значения v переменной x , обозначается $\langle x, v \rangle$. Метка $\langle x, v \rangle$ осмысленна только в том случае, когда $v \in Dx$.

Составное меткой называется одновременное присваивание значений некоторому множеству переменных.

Составная метка, присваивающая значения v_1, \dots, v_n соответственно переменным x_1, \dots, x_n , обозначается $\langle x_1, v_1 \rangle, \dots, \langle x_n, v_n \rangle$. Здесь также полагается, что $v_i \in Dx_i$, для $i = 1, \dots, n$.

Ограничением на некоторое множество переменных S называется набор составных меток для всех переменных множества S . Будем обозначать ограничение C_s .

Определение 1. Задача удовлетворения ограничений – это тройка

$$P = (X, D, C),$$

обозначаемая $CSP(P)$, где X – конечное множество переменных $\{x_1 \dots x_k\}$, D – функция, отображающая каждую переменную из X на множество объектов произвольного типа: $D: X \rightarrow \{\text{конечное множество объектов некоторого типа}\}$. Будем рассматривать Dx_i как множество объектов, отображенных из x_i функцией D . Эти объекты называются значениями переменной x_i , а множество Dx_i – областью x_i ,

C – конечное (возможно пустое) множество ограничений на произвольном подмножестве переменных из X , то есть C – это множество наборов составных меток.

Каждое ограничение из C имеет вид одной из следующих формул:

$$y = x; \quad (9)$$

$$y = c; \quad (10)$$

$$V = f\{x_1, \dots, x_n\}, \quad (11)$$

где x, x_i, y – символы переменных из V ,

c – константный символ,

f – функциональный символ аргументности n .

Более сложные ограничения распадаются на множество более простых (вышеприведенных видов) после введения дополнительных переменных [4].

Определение 2. Решением задачи удовлетворения ограничений (V, C) называется такое приписывание каждой переменной из V некоторого определенного значения из ее универсума, при котором выполняются все ограничения из C .

В общем случае все точные решения задачи, если таковые существуют, должны лежать в декартовом произведении таких n -значений.

Определение 3. Недоопределенным расширением (n -расширением) универсума X называется любой конечный набор его подмножеств $*X$, содержащий \emptyset , и являющийся замкнутым относительно пересечения.

Эти свойства гарантируют однозначное представление $*[\xi]$ любого подмножества $\xi \subseteq X$ в n -расширении $*X$, а именно: $*[\xi] = \bigcap_{\xi \subseteq \zeta \in *X} \zeta$.

Таким образом, представлением множества ξ в системе $*X$ является минимальное подмножество из n -расширения $*X$, содержащее ζ .

Определение 4. Для вычисления N -модели обобщенная вычислительная модель (n -модель) M состоит из четырех множеств:

$$M = (V, C, W, CORR), \quad (12)$$

где V – множество n -объектов v из заданной предметной области,

C – множество ограничений на n -объектах из V ,

W – множество функций присваивания,

$CORR$ – множество функций проверки корректности.

С каждым объектом из V связаны недоопределенный тип данных (недоопределенное расширение некоторого универсума), начальное значение, функция присваивания и функция проверки корректности.

Функция присваивания – это двухместная функция, работающая при каждой попытке присваивания очередного значения n -объекту и определяющая его новое значение как пересечение текущего и присваиваемого значений.

Функция проверки корректности – это унарный предикат, который проверяет непустоту значения n -объекта.

Алгоритм вычислений, реализованный в n -моделях, является высокопараллельным процессом, который управляется потоком данных. Изменение значений переменных, располагающихся в общей памяти, автоматически влечет интерпретацию тех ограничений, для которых эти переменные являются аргументами. Процесс останавливается, когда сеть стабилизируется или хотя бы один из n -объектов становится

некорректным. В последнем случае устанавливается противоречивость исходной n -модели.

При решении задачи с использованием недоопределенных моделей необходимо определить множество n -объектов из заданной предметной области и множество ограничений на n -объектах на этих объектах.

Каждая задача $(t_{i,j,k})$ является структурным расширением интервального n -объекта, включая: трудоемкость выполнения, время начала, время окончания. Отношения непосредственного предшествования между задачами, которые характеризуют частичную упорядоченность задач, задаваемое в виде неравенств входит в общую систему C – множество ограничений на n -объектах из V . В начале вычислений для всех задач времена начала и окончания совпадают с началом и окончанием всего проекта, в результате вычислений происходит уточнение этих сроков.

Модель компетенций определена в декартовом пространстве $E \times T \rightarrow F_{\text{experience}}$ и представляется целыми константами для каждого сочетания сотрудник-работа, характеризуя имеющийся опыт участников проекта. Модель ролевых запретов определена на пространстве $\dot{O} \times T \rightarrow F_{\text{ban}}$ в множестве констант логического типа.

Искомое решение задачи о назначениях ищется в декартовом произведении пространств сотрудник-задача-время: $E \times T \times \tau \rightarrow Works$ как множество вещественных переменных, характеризующих относительную занятость сотрудника на исполнение определенной роли в определенный момент времени, при выполнении неравенств, входящих в общее множество ограничений:

- роль должна соответствовать модели компетенции и возможности исполнителя с определенным уровнем подчиненности ее исполнять;
- для каждого момента времени сумма занятостей сотрудника во всех исполняемых ролях не должна превышать 1.

Таким образом, исходная постановка оптимизационной задачи при использовании недоопределенных моделей заменена на поиск и определение такого множества вариантов работ, при которых все частные показатели эффективности проекта удовлетворяют соответствующим ограничениям.

Помимо перечисленных, необходимо учитывать дополнительные общепроектные ограничения, к которым относятся: ограничение на фонд заработной платы, на общую трудоемкость работ, на средний уровень занятости исполнителей в проекте, на степень участия каждого исполнителя в конкретных работах проекта и т.д.

Метод вычислений, реализующий вычисления на n -моделях, является высокопараллельным процессом,

который управляется потоком данных. Изменение значений переменных, располагающихся в общей памяти, автоматически влечет интерпретацию тех ограничений, для которых эти переменные являются аргументами. Процесс останавливается, когда сеть стабилизируется или хотя бы один из n -объектов становится некорректным. В последнем случае устанавливается противоречивость исходной n -модели.

3. Алгоритмизация прототипа программного обеспечения для планирования программных проектов

Для рассмотрения алгоритмов распространения ограничений необходимо ввести ряд определений, которые используются в данных алгоритмах.

Определение 3.1. Областью определения переменной называется множество всех допустимых значений, которые могут быть присвоены данной переменной. Если x – это переменная, то через D_x обозначается область ее определения.

Определение 3.2. Меткой называется пара переменная-значение, которая представляет присваивание значения переменной. Для обозначения метки используется запись $\langle x, v \rangle$ (что означает присваивания значения v переменной x). Метка $\langle x, v \rangle$ имеет смысл только тогда, когда v принадлежит области определения переменной x (т.е. $x \in D_x$).

Определение 3.3. Составная метка – это одновременное присваивание значений множеству (возможно пустому) переменных. Запись $(\langle x_1, v_1 \rangle, \langle x_2, v_2 \rangle, \dots, \langle x_n, v_n \rangle)$ обозначает составную метку, которая присваивает значения v_1, v_2, \dots, v_n переменным x_1, x_2, \dots, x_n .

Определение 3.4. k -составной меткой называется составная метка, которая k переменным присваивает k значений.

Определение 3.3. Ограничением на некотором множестве S переменных называется множество составных меток для переменных из множества S . Для удобства используется запись CS , обозначающая ограничение на переменные из множества S .

Определение 3.6. Задачей удовлетворения ограничений (ЗУО) называется тройка

$$(Z, D, C), \quad (5)$$

где Z – конечное множество переменных $\{x_1, x_2, \dots, x_n\}$;

D – функция, которая отображает каждую переменную из множества Z на множество объектов произвольного типа:

$D: Z \rightarrow$ конечное множество объектов (заданного типа);

C – конечное (возможно пустое) множество ограничений на произвольном подмножестве переменных из Z .

Определение 3.7. Гиперграфом ограничений ЗУО (Z, D, C) называется гиперграф, в котором каждая вершина соответствует переменной из Z , а каждое (гипер-)ребро соответствует ограничению из C .

Определение 3.8. ЗУО является совместной в вершинах тогда и только тогда, когда для каждой переменной все значения в области определения этой переменной удовлетворяют всем ограничениям над данной переменной:

$$\forall \text{csp}((Z, D, C)):$$

$$\text{node-consistent}((Z, D, C)) \equiv$$

$$(\forall z \in Z: (\forall v \in D_x: \text{satisfies}(\langle x, v \rangle, C_x))) .$$

Определение 3.9. Путь (x_0, x_1, \dots, x_m) в графе ограничений для ЗУО называется совместным по пути тогда и только тогда, когда для каждой 2-составной метки $(\langle x_0, v_0 \rangle, \langle x_m, v_m \rangle)$, которая удовлетворяет всем ограничениям на x_0 и x_m , существует метка для каждой переменной из x_1, \dots, x_{m-1} такая, что каждое бинарное ограничение на смежных переменных в этом пути удовлетворяется:

$$\forall \text{csp}((Z, D, C)):$$

$$\forall x_0, x_1, \dots, x_m \in Z: \text{PC}((x_0, x_1, \dots, x_m), (Z, D, C)) \equiv$$

$$(\forall v_0 \in D_{x_0}, v_m \in D_{x_m}:$$

$$(\text{satisfies}(\langle x_0, v_0 \rangle, \langle x_m, v_m \rangle), C_{x_0, x_m}) \Rightarrow$$

$$(\exists v_1 \in D_{x_1}, v_2 \in D_{x_2}, \dots, v_{(m-1)} \in D_{x_{(m-1)}}:$$

$$\text{satisfies}(\langle x_1, v_1 \rangle, C_{x_1}) \wedge \dots \wedge$$

$$\text{satisfies}(\langle x_{m-1}, v_{m-1} \rangle, C_{x_{m-1}}) \wedge$$

$$\text{satisfies}(\langle x_0, v_0 \rangle, \langle x_1, v_1 \rangle, C_{x_0, x_1}) \wedge \dots \wedge$$

$$\text{satisfies}(\langle x_1, v_1 \rangle, \langle x_2, v_2 \rangle, C_{x_1, x_2}) \wedge \dots \wedge$$

$$\text{satisfies}(\langle x_{m-1}, v_{m-1} \rangle, \langle x_m, v_m \rangle, C_{x_{m-1}, x_m}) .$$

Определение 3.10. ЗУО называется совместной по путям тогда и только тогда, когда каждый путь в графе этой ЗУО является совместным по пути.

Это предполагает, что если ЗУО является совместной по путям, то для всех переменных x и y , всякий раз, когда составная метка $(\langle x, a \rangle, \langle y, b \rangle)$ удовлетворяет ограничениям на x и y , существует метка $(\langle z, c \rangle)$ для каждой переменной z такая, что $(\langle x, a \rangle, \langle y, b \rangle, \langle z, c \rangle)$ удовлетворяет всем ограничениям на x, y и z .

Определение 3.11. Графом ограничений задачи ЗУО (Z, D, C) называется неориентированный граф, в котором каждая вершина соответствует переменной из Z , а ребра между вершинами заданы для тех пар пе-

ременных их Z , которые являются частью k -ограничения из C .

$$\forall \text{graph}((V,E)): (V,E)=G((Z,D,C)) \equiv \\ \equiv ((V=Z) \wedge E=\{(x,y)|x,y \in Z \wedge (\exists C_s \in C: x,y \in S))).$$

Выводы

В статье представлена формализованная модель плана программного проекта, которая содержит 2 основных блока:

1. Модель объектов управления – процессов создания продукта и управления проектом.

2. Модель субъектов управления – проектной команды, которая предназначена для учета ограничений, характеризующих: опыт решения задач различного типа, запрет на одновременное выполнение двух определенных ролей одним исполнителем, а также фактическую возможность выполнения кон-

кретной задачи определенным исполнителем описание команды проекта.

Литература

1. Нарусбаев, А.А. Введение в теорию проектных решений [Текст] / А.А. Нарусбаев – Л. Судостроение, 1976. – 221 с.
2. Эддоус, М. Методы принятия решений [Текст] / М. Эддоус, Р. Стэнсфилд – М.: Аудит, Юнити, 1997. – 590 с.
3. Телерман, В.В. Удовлетворение ограничений в задачах математического программирования [Текст] / В.В. Телерман, Д.М. Ушаков // Вычислительные технологии. – 1998. – Т. 3, N 2. – С. 45 – 54.
4. Телерман, В.В. Неопределенные модели: формализация подхода и перспективы развития [Текст] / В.В. Телерман, Д.М. Ушаков // Проблемы представления и обработки не полностью определенных знаний. – М.-И-ск, 1996. – С. 7 – 32.

Поступила в редакцию 18.05.2012

Рецензент: д-р техн. наук, проф., проф. кафедры ПО ЭВМ С.Ю. Шабанов-Кушнарченко, Харьковский национальный университет радиоэлектроники, Харьков, Украина.

СИСТЕМНА МОДЕЛЬ КАЛЕНДАРНОГО ПЛАНУВАННЯ ПРОГРАМНИХ ПРОЕКТІВ

М.В. Потапова

Проведений аналіз основних особливостей програмних проектів. Описаний процес робіт із створення і виконання програмних проектів. Виконаний огляд вимог, що пред'являються до систем планування програмних проектів. Розглянуті теоретичні підходи до календарного планування програмних проектів. Описана системна модель календарного планування виконання програмних проектів обмеженні. Доведена необхідність вживання методів поширення обмежень і математичних методів, орієнтованих на обробку неточної і неповної інформації. Представлений процес алгоритмізації прототипу програмного забезпечення для планування програмних проектів.

Ключові слова: системна модель, програмний проект, невизначеність, обмеження, календарне планування, управління проектами.

SYSTEM MODEL OF SCHEDULING OF PROGRAM PROJECTS

M.V. Potapova

The analysis of the main features of program projects is carried out. Process of works on creation and implementation of program projects is described. The review of the requirements shown to systems of planning of program projects is executed. Theoretical approaches to scheduling of program projects are considered. The system model of scheduling of implementation of program projects restriction is described. Need of application of methods of distribution of restrictions and the mathematical methods focused on processing of inexact and incomplete information is proved. Process of algorithmization of a prototype of the software for planning of program projects is presented.

Keywords: system model, program project, uncertainty, restrictions, scheduling, management of projects.

Потапова Марина Викторовна – заведуюча лабораторією кафедри фінансів Національного аеро-космічного університету ім. Н.Е. Жуковського «ХАІ», Харьков, Украина, e-mail: marina-k604@yandex.ru.