

УДК 004:007:51-7

Ю. А. Прокопчук

Институт технической механики НАНУ и ГКАУ
ул. Лешко-Попеля, 15, 49005 Днепропетровск, Украина
e-mail: itk3@ukr.net

Построение систем смыслопорождения на основе парадигмы предельных обобщений

Предложен метод построения систем смыслопорождения на основе парадигмы предельных обобщений. В рамках метода для каждой Z-задачи управления создается банк конфигураторов тестов и банк прецедентов. Далее автоматически формируются предельные модели знаний, позволяющие решать целевую Z-задачу. Банк тестов, пространство системопаттернов, орграфы набросков базы прецедентов, модели знаний составляют основу системы смыслопорождения.

Ключевые слова: парадигма предельных обобщений, синдромные и вероятностные модели знаний, системы смыслопорождения.

Введение и анализ существующих решений

С проникновением интеллектуальных систем во все сферы жизни общества все большее значение приобретают когнитивные надстройки, в частности системы смыслопорождения (Sensemaking) [1, 2]. К сфере применения таких систем относится обеспечение максимальной наблюдаемости, управляемости, ситуационной осведомленности и информационно-системной безопасности сложных антропоцентрических и сетецентрических объектов в типовых и нештатных ситуациях [1–13]. Важной сферой применения является автоматизированный аудит, контроль качества выполнения сеансов функционирования и оптимизация технологических процессов, например в медицине, металлургической и химической промышленности [9, 10]. Системы смыслопорождения являются необходимой компонентой перспективных человеко-машинных рефлексивных групп и сред, когнитивных центров (обеспечение гетерогенной среды моделирования ситуаций в рамках сеанса функционирования). Большое внимание развитию систем смыслопорождения уделяется зарубежными ИТ-корпорациями: IBM's Sensemaking System, The Sensemaking platform, Smart Sensemaking Systems [2].

Основным недостатком при использовании интеллектуального управления во многих практических приложениях является сложность реализации всего комплекса функциональных систем. Необходима радикальная редукция, но с сохранением целостности, как восприятия ситуации, так и реакции на ее изменения. Редукция

© Ю. А. Прокопчук

информационного обеспечения предполагает реализацию концепции системного синтеза, которая заключается в выделении и последующем управлении только по ключевым параметрам [3–5]. В работах [9–13] рассматривается системный синтез, опирающийся на синергетическую концепцию единства процессов самоорганизации и управления в рамках парадигмы предельных обобщений (ППО). ППО основана на автоматизированном формировании внутренних минимальных кодов систем смыслопорождения, с помощью которых решаются как задачи различения (распознавания), так и задачи управления. Отметим, что возможность оценки (измерения) любого параметра напрямую зависит от степени обобщенности домена, из которого выбираются значения (особенно это относится к субъективным параметрам). Повышение точности измерения требует дополнительных ресурсов: приборных, временных, вычислительных и т.д. ППО позволяет установить максимально наблюдаемые комплексы ключевых параметров.

Постановка задачи

Требуется разработать метод построения систем смыслопорождения широкого назначения, основанный на парадигме предельных обобщений. Метод должен опираться на совокупность моделей и инструментов, представленных в работах [9–12]. Также необходимо разработать иерархию конфигураторов тестов для задач мониторинга.

Решение поставленной задачи

Предполагается, что произвольную ситуацию мониторинга (прецедент, систему, процесс, образ) можно описать с помощью множества элементарных тестов, представимых в виде «тест = значение». Результаты любого теста τ могут выбираться из разных доменов T (множеств значений со связями), которые образуют ациклический орграф доменов $G(\tau) = \{T \rightarrow T\}_\tau$. Триада $(T \rightarrow T)$ является простейшим системопаттерном и базовым конструктом смысла [9, 10]. В совокупности орграфы доменов образуют Банк тестов $\{G(\tau)\}$. Различные комбинации доменов для всех тестов определяют различные уровни общности описания ситуаций действительности. Для решения Z -задачи управления формируется множество прецедентов с известными исходами $\Omega = \{\alpha(\{\underline{t}/T\}, \underline{z}/Z)\}$, где $Z = \{1, \dots, N\}$ — множество заключений; $\{\underline{t}/T\}$ — множество значений тестов, с помощью которых описываются прецеденты; \underline{z}/Z — исход. Множество Z является базовым доменом орграфа доменов $G(z)$. Пусть банк тестов $\{G(p)\}$ образует полное транзитивно-индукторное вычислительное пространство, при этом $\{G(\tau)\}$, $G(z)$ принадлежат $\{G(p)\}$. Частью этого пространства могут быть когнитивные карты разного вида и разной степени обобщенности.

Для любого орграфа доменов $G(p)$ могут быть автоматически построены разнообразные структурно-завершенные орграфы доменов $G^+(p)$ и единственный предельный структурно-завершенный орграф доменов $G^{++}(p)$ [9]. Структурно-завершенные орграфы строятся на основе операции расщепления домена, а именно: для любой дискретной вершины $D = \{a_1, a_2, \dots, a_n\}$ такой, что $n > 2$, порождаются n терминальных вершин-листьев вида $\{a_j; \text{не } a_j\} \equiv \{a_j; a_j\}$, где $j = 1, \dots, n$. В результирующем орграфе остаются только те вершины, которые отсутствовали в

орграфе до выполнения операции расщепления. Операцию расщепления можно применять локально — лишь к одной вершине орграфа, а можно последовательно ко всем дискретным вершинам орграфа. Орграф, в котором все вершины-листья соединены со всеми возможными доменами-предками, называется предельным структурно-завершенным орграфом.

Без потери общности примем, что каждый тест входит в описание прецедента (ситуации действительности) один раз. Через $\Omega(\{\tau/T_0\}, Z)$ обозначим априорные описания прецедентов. Множество всех описаний базы прецедентов образует орграф набросков $\Omega(Z)$, в котором $\Omega(\{\tau/T\}, Z)$ — отдельный набросок. Можно также рассмотреть орграф набросков каждого прецедента α , тогда описание $\alpha(\{\tau/T\}, \underline{z}/Z)$ — это отдельный набросок.

Произвольные вычисления и преобразования в рамках единого вычислительного пространства систем смыслопорождения, описываются с помощью *динамических системопаттернов* (или просто системопаттернов) вида [9–11]

$$f/\mu: \{a/A\}, e/E \rightarrow \{b/B\}, \quad \mu \in \{\mu\}_f, \quad (1)$$

где $\{a/A\}$ — входные тесты; $\{b/B\}$ — выходные тесты; e/E — требуемая структурная энергия, ресурсы; μ — механизм реализации. С помощью структурной энергии, в частности, может передаваться возбуждение (активность) в вычислительной среде. Множество механизмов $\{\mu\}_f$ и множество системопаттернов $\{f/\mu\}$ указывают на важное свойство *разнообразия* (Diversity) или *мультиверсионность* вычислений (Multy-Version Computing) на всех системных уровнях. Модель системопаттерна двойственна: она описывает соединение «заказчика» (агента-Потребности f) и «исполнителя» (агента-Возможности μ), что позволяет применять модели агентных ПВ-сетей [7]. Постоянный поиск соответствий между конкурирующими и кооперирующими агентами потребностей и возможностей на виртуальном рынке позволяет строить решение любой сложной задачи как динамическую сеть связей, гибко изменяемую в реальном времени. Таким образом, банк тестов $\{G(p)\}$ и системопаттерны $\{f/\mu\}$ позволяют создавать *математически однородное поле* компьютерной информации любого масштаба и любой физической реализации.

Кортеж $K = \langle \Omega(Z), \{G(\tau)\}, \{G(p)\}, G(z) \rangle_Z$ называется контекстом Z -задачи. *Идеальной закономерностью* V в рамках контекста K называется произвольная совокупность значений тестов, позволяющая однозначно установить заключение [10, 11]:

$$V = (\{\underline{a}/A\} \rightarrow \underline{z}/Z), \quad \exists \alpha(\{\tau/T\}_\alpha, \underline{z}/Z) \in \Omega(Z): \{\underline{a}/A\} \subseteq \{\tau/T\}_\alpha. \quad (2)$$

Компактная запись: $V(\{\underline{a}/A\}, \underline{z}/Z)$. Закономерность является разновидностью системопаттерна и примером *ad hoc гипотезы*. Закономерность $V'(\{\underline{a}/A\}', \underline{z}/Z)$ доминирует закономерность $V(\{\underline{a}/A\}, \underline{z}/Z)$, если $\{a\}' \subseteq \{a\}$, $\forall a \in \{a\}': A \leq A'$ и выполняется хотя бы одно из условий: а) $|\{a\}'| < |\{a\}|$; б) $\exists a \in \{a\}': A < A'$.

Наряду с идеальными закономерностями имеют место *вероятностные закономерности* $R = (\{\underline{a}/A\} \rightarrow J_z \underline{z}/Z)$, где J_z — модификатор истинности заключения.

Вероятностные закономерности являются предвестниками событий и участвуют в формировании управления на основе синдромного принципа [9, 11].

Формальным синдромом S (или просто *синдромом*) в рамках контекста K называется неизбыточная идеальная закономерность. *Предельным синдромом* S^* в рамках контекста K называется синдром, у которого отсутствуют доминирующие закономерности. Предельные синдромы, выступающие в качестве параметров порядка, используются для организации целенаправленного поведения (синдромный принцип управления). Синдромы и предельные синдромы являются результатом *когнитивной категоризации* или обобщения.

Множество всех закономерностей в рамках контекста K обозначим через $\{V\}_{Full}$, множество всех синдромов обозначим через $\{S\}_{Full}$, а множество всех предельных синдромов обозначим $\{S^*\}_{Full}$. Ясно, что $\{S^*\}_{Full} \subseteq \{S\}_{Full} \subseteq \{V\}_{Full}$.

Моделью знаний называется произвольное множество закономерностей $\{V\}$, которое позволяет установить заключение как минимум для каждого прецедента из $\Omega(\{\tau/T_0\}, Z)$, где T_0 — базовые домены. Модель знаний $\{V'\}$ доминирует модель знаний $\{V\}$, если она применима к большему числу набросков прецедентов из $\Omega(Z)$, включая и те наброски, к которым применима $\{V\}$. Модели знаний $\{V\}_{Full}$, $\{S\}_{Full}$ и $\{S^*\}_{Full}$ недоминируемы и эквивалентны в плане доминирования [10]. Минимальные модели знаний на основе $\{S^*\}_{Full}$ приводят к функциональным системам (ФС), которые операционально решают Z -задачу [9–11]. ФС позволяют реализовать технологию *банка взаимодействующих наблюдателей*.

Контекст задачи должен быть таким, чтобы некоторые из предельных синдромов можно было отнести к параметрам порядка благоприятного и неблагоприятного течения изучаемого процесса. В таком случае предельная синдромная модель знаний будет содержать конкурентные наборы параметров порядка, а управление технологическим процессом и технологический аудит заключаются в синдромной диагностике и задании целевых синдромов (параметров порядка), отвечающих благоприятному режиму (выходу на этот режим и его стабилизации) [12]. При этом желательно нейтрализовать предвестники развития неблагоприятного режима, которые содержатся в предельной вероятностной модели знаний. В этом суть синдромного принципа управления (СПУ) сложными процессами любой природы [9, 11] и функциональное назначение системы смыслопорождения.

Уточним постановку Z -задач с учетом введенных сущностей. Под Z -задачами понимаются три основные задачи: а) различения: $\alpha(\{\underline{z}/T\}, z/Z)$; б) перевода в целевое состояние: дано $\alpha(\{\underline{z}/T\}, \underline{z}'/Z)$, найти $\{S\}_U \cup \{R\}_U$ такое, что $\underline{z}'/Z \rightarrow \underline{z}/Z$; в) стабилизации: достигнут целевой результат \underline{z}/Z , требуется его закрепить или удержать (добавляются «хорошие» синдромы и предвестники и удаляются «плохие» предвестники). Все три задачи решаются одинаково на основе ФС и СПУ. Развертывание Z -задач во времени образует Z -поток — Z -Flow.

Для информационных приложений орграфы доменов тестов удобно представлять в виде конфигураторов. Общую схему конфигураторов с использованием синтаксиса лексических деревьев можно представить следующим образом [9]:

$$\begin{aligned} \text{Тест } [^{\wedge} \text{Тест}...] \text{ } [^{\#} \text{Тест}X...] \{ \\ \text{Dom}_1 [^{\wedge} \text{Dom}_1...] [^{\#} \text{Dom}X...] \{ ; ; \} \\ \text{Dom}_2 [^{\wedge} \text{Dom}_2...] [^{\#} \text{Dom}Y...] \{ ; ; \} \end{aligned} \quad (3)$$

...
 $\text{Dom_N} [\wedge \text{Dom_N...}] \{ ; ; \},$

где «Тест» — название теста; « \wedge Тест...» — список условных обозначений теста; «# ТестХ...» — список ссылок на более общие тесты; «Dom_K» — название K -го домена (произвольный терм, состоящий из букв и/или чисел); « \wedge Dom_K...» — список условных обозначений K -го домена; «#DomХ...» — ссылка на домены предки; { ; ; } — список элементов домена со связями обобщения. Каждый элемент домена может иметь собственный список обозначений, которые также играют роль символов групп обобщения. Элементы доменов могут содержать дополнительные конструкции, которые обеспечивают однозначность вычислительных схем в зависимости от внешних тестов. Порядок размещения доменов в конфигураторе — сверху вниз и слева направо — означает рост точности значений теста за счет большей детализации (увеличения числа элементов). В упорядоченной последовательности доменов метки элементов любого домена явно задают однозначные правила перерасчета значений из домена-предка в домен-потомок, размещенный выше или слева. Конфигураторы тестов, как и орграфы доменов, будем обозначать G . Примеры конфигураторов будут приведены ниже. В работе [9] предложен метод штрих-кода для автоматического формирования конфигураторов. Реализовать конфигураторы можно, например, с помощью электронной таблицы. В работе [9] описан способ реализации конфигураторов и системопаттернов в рамках веб-ориентированного многоцелевого банка знаний (для медицины).

Каждый конфигуратор является, по сути, моделью индуктивного обобщения результатов теста. Для числовых тестов ключевым моментом является переход от непрерывного интервала к дискретному разбиению (фазовый переход от бесконечности к конечности). Подобный переход может быть выполнен разными способами, что, безусловно, отражается на результатах моделирования. Важно отметить, что конфигураторы стирают границу между непрерывным и дискретным: любой непрерывный тест всегда имеет множественное дискретное представление (интервальное, символическое). В конфигураторы вложена система интерпретаций ПрО, т.е. семантический уровень. Интерпретируемость данных и результатов обработки данных в системе понятий предметной области является необходимым условием получения полезного результата. Конфигураторы тестов – это также и набор операций (непрерывных, интервальных, нечетких, фрактальных, лингвистических), которые можно производить со значениями тестов разного уровня общности (разных сортов) с точки зрения рассматриваемой предметной области. Все операции имеют вид (1).

Таким образом, системы смыслопорождения на основе ППО позволяют решать, в частности, следующие практические задачи [9, 12]:

- получение новых системных знаний об изучаемом объекте (в виде предельных синдромных и вероятностных моделей знаний);
- определение конкурирующих наборов управляемых входных параметров, которые минимально достаточны для обеспечения благоприятного течения процесса;
- анализ управляемости процесса (оценка возможностей существующей системы управления обеспечить в каждой реализации процесса заданное качество его функционирования);

— прогноз значений выходных параметров по известным значениям входных параметров, в частности, оперативная оценка значений выходных параметров, приборный анализ которых требует значительных затрат времени и ресурсов;

— определение требуемой точности измерения входных параметров (повышение наблюдаемости).

Рекомендации, полученные в результате аудита и субоптимизации на основе ППО, могут включать [9, 12]: оптимизацию действующих технологических процессов по технологическим, экономическим и экологическим критериям или комплексам этих критериев; определение научно и технологически обоснованных требований к качеству сырья, при заданных требованиях к качеству функционирования конкретного технологического процесса; технологию обеспечения заданного качества функционирования процесса при существенных вариациях состава сырья; технологию обеспечения необходимой технологической гибкости с целью получения продукта, удовлетворяющего индивидуальным требованиям различных заказчиков.

Пример. Мониторинг состояния сложных систем и процессов

В связи с неуклонным ростом числа контролируемых параметров полета космического аппарата (КА) актуальными являются следующие темы исследований: «Разработка моделей для проведения математического и натурального моделирования по обоснованию принципов предварительной обработки и сжатия целевой информации в бортовых и наземных высокопроизводительных командно-информационных системах»; «Создание прикладной интеллектуальной системы информационной поддержки мониторинга и смыслопорождения». Успех в реализации данных исследований создаст необходимые предпосылки обеспечения максимальной ситуационной осведомленности и информационно-системной безопасности авиационно-космических комплексов [2–6].

Одной из прикладных задач исследований является разработка иерархии схем конфигураторов тестов подсистемы смыслопорождения системы мониторинга КА, согласованных с оценками экспертов [8].

Множество возможных состояний контролируемого объекта — системы, подсистемы, узла представляет совокупность некоторого количества заключений Z , в простейшем случае трех: $Z = \{1 — Sшт — штатное состояние контролируемой системы; 2 — Sзам — замечание к работе контролируемой системы, 3 — Sнс — нештатное состояние контролируемой системы\}$.

Под *технологическим условием* U понимается некоторая логическая функция технологических параметров, принимающая значения «истинно» или «ложно». Технологические условия делятся на три типа: $U = \{UX; UL; UB\}$, где UX — удержание технологического параметра ($x = Const$) или скорости его изменения ($dx/dt = Const$) на заданном значении; UL — удержание технологического параметра или скорости его изменения в заданных границах ($x \in [x_{min}; x_{max}]$ или $dx/dt \in [x'_{min}; x'_{max}]$); UB — удержание статического или динамического баланса между параметрами. Под *технологическим событием* (E) понимается определенное событие, значимое с точки зрения выполняемых задач диагностики и управления. Типичным технологическим событием является нарушение технологического условия

(*EU*) (например, технологический параметр вышел за уставки). Еще один важный тип технологического события — дискретное явление, связанное с работой оборудования (*EW*), например, срабатывание защиты. Далее ограничимся технологическим условием *UL*.

Рассмотрим способы построения конфигуратора одного непрерывного параметра с двусторонним ограничением области допустимых значений. Приведем несколько простых схем, образующих иерархию.

Область всех возможных значений параметра p обозначим $[p_{\min}; p_{\max}]$. Во всех приведенных схемах домен «1» является базовым (наиболее точным).

Схема 1. Обозначим регламентный (или допустимый) интервал значений p через $P_{\text{don}} = [p1, p2]$. Конфигуратор:

$p \{3 \{ \text{Норма}^2; \text{Отклонение}^1 \} 2 \{ \text{Ниже нормы}^1 [p_{\min}; p1]; \text{Норма}^2 [p1, p2]; \text{Выше нормы}^3 (p2; p_{\max}) \} 1 \{ [p_{\min}; p_{\max}] \}$.

Предельный структурно-завершенный орграф $G^{++}(p)$ показан на рис. 1,а. Самая нижняя вершина (красная) — это базовая вершина; светлые вершины (зеленые) — это домены-листья вида $\{a, -a\}$.

Схема 2. Оценка непрерывного параметра с двусторонним ограничением с разбиением области на среднестатистические и предельно допустимые значения. Среднестатистический диапазон (СД) обозначим $[p_{cp1}, p_{cp2}]$. Ясно, что $[p_{cp1}, p_{cp2}] \subset [p1, p2]$. Конфигуратор представим в виде (ОПФ — область пограничного функционирования):

$p \{4 \{ \text{Норма}^2; \text{Отклонение}^1 \} 3 \{ \text{Ниже нормы}^1; \text{Норма}^2 \} 4; \text{Выше нормы}^5 \} 2 \{ \text{Ниже нормы}^1 [p_{\min}; p1]; \text{ОПФ1}^2 [p1, p_{cp1}]; \text{СД}^3 [p_{cp1}, p_{cp2}]; \text{ОПФ2}^4 (p_{cp2}, p2]; \text{Выше нормы}^5 (p2; p_{\max}) \} 1 \{ [p_{\min}; p_{\max}] \}$.

Предельный структурно-завершенный орграф $G^{++}(p)$ показан на рис. 1,б.

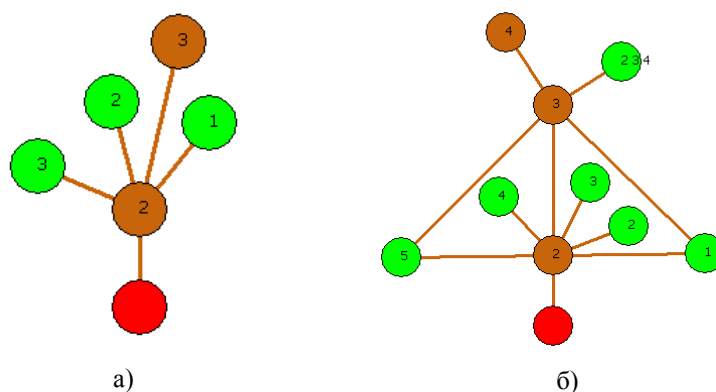


Рис. 1. Орграфы доменов: а) схема 1; б) схема 2

Схема 3. Схема опирается на экспертный подход к построению оценочной функции на интервалах ОПФ1 и ОПФ2 (рис. 2), а именно: с помощью параметров

a_1, b_1 для ОПФ1 и a_2, b_2 для ОПФ2 данные области разбиваются на три подинтервала. В итоге получим следующий конфигуратор:

- p {
- 5 {Норма ²; Отклонение ^{1 9}}
- 4 {Ниже нормы ¹; Норма ^{2 5 6}; Выше нормы ⁹}
- 3 {Ниже нормы ¹; ОПФ1 ^{2 3 4}; СД ⁵; ОПФ2 ^{6 7 8}; Выше нормы ⁹}
- 2 {Ниже нормы ¹ [$p_{\min}; p_1$]; ОПФ1_1 ² [p_1, a_1]; ОПФ1_2 ³ [a_1, b_1]; ОПФ1_3 ⁴ [b_1, p_{cp1}]; СД ⁵ [p_{cp1}, p_{cp2}]; ОПФ2_1 ⁶ (p_{cp2}, a_2); ОПФ2_2 ⁷ (a_2, b_2); ОПФ2_3 ⁸ (b_2, p_2); Выше нормы ⁹ ($p_2; p_{\max}$)}
- 1 {[$p_{\min}; p_{\max}$]}

Предельный структурно-завершенный оргграф $G^{++}(p)$ показан на рис. 3.

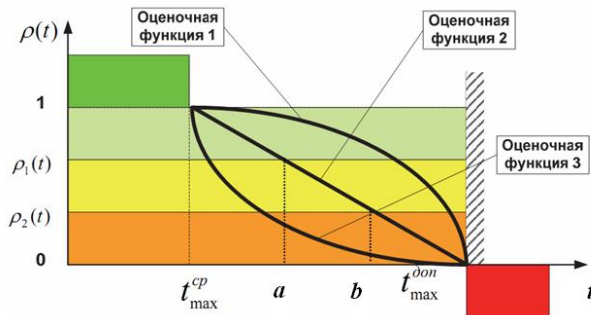


Рис. 2. Пример оценочной функции

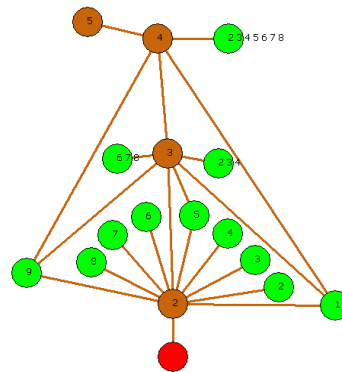


Рис. 3. Оргграф. Схема 3

На рис. 2 $\rho(t)$ — оценочная функция, формируемая экспертами [8]. Конфигураторы тестов могут служить основой нечетких моделей [9].

Если контролируемый параметр p выходит из области среднестатистического диапазона [p_{cp1}, p_{cp2}], но еще не выходит из области допустимых значений [p_1, p_2], оперативной группе управления необходимо принять решение по дальнейшему управлению объектом контроля, поскольку есть тенденция к переходу от его штатного состояния к нештатному и для оперативной группы управления он находится в некотором промежуточном состоянии $S_{пром}$. При этом необходимо учитывать, что с точки зрения требования к работоспособности объекта контроля, параметр находится в пределах нормы, т.е. изменение параметра в данных пределах важно именно для специалистов оперативной группы управления [8].

Таким образом, схема 1 является самой простой, а схема 3 позволяет в максимальной степени учесть опыт специалистов группы управления при построении предельных моделей знаний в рамках подсистемы смыслопорождения. По аналогии можно детализировать «запредельные» интервалы. Отметим, что усложнение схем конфигураторов приводит к увеличению вычислительных ресурсов.

Выводы

Построение систем смыслопорождения на основе ППО позволяет определить перечень конкурентных наборов управляемых параметров (параметров по-

рядка), которые минимально достаточны для решения целевой задачи. На основе предельных моделей знаний проводят оценку возможностей существующей системы управления обеспечить в каждой реализации технологического процесса заданное качество его работы; определяют нормы технологического режима по всем входным параметрам, обеспечивающим снижение себестоимости выпускаемой продукции при условии соблюдения требований по качеству, выбросам вредных веществ в окружающую среду и другим ограничениям.

1. *Weick Karl E.* Making Sense of the Organization: Volume 2: The Impermanent Organization / Weick Karl E. — John Wiley & Sons, 2012. — 310 p.

2. *Leedom D.K.* Final Report: Sensemaking Symposium (Technical Report Prepared Under Contract for Office of Assistant Secretary of Defense for Command, Control, Communications & Intelligence). — Vienna, VA: Evidence Based Research. Inc., 2001. — Режим доступа: http://www.dodccrp.org/files/sensemaking_final_report.pdf

3. *Потюпкин А. Ю.* Научно-методические основы решения задач анализа состояния объектов РКТ в условиях неопределенности / А.Ю. Потюпкин — М.: ВА РВСН им. Петра Великого, 2003. — 187 с.

4. Интеллектуальное управление динамическими системами. Монография / [С.Н. Васильев, А.К. Жерлов, Е.А. Федосов, Б.Е. Федун]. — М.: Физматлит, 2000. — 352 с.

5. *Неделько В.Н.* Прогнозирование информационных потребностей операторов особо сложных систем управления / В.Н. Неделько // Искусственный интеллект. — 2002. — № 3. — С. 420–427.

6. *Додонов А.Г.* Живучесть информационных систем / А.Г. Додонов, Д.В. Ланде. — К.: Наук. думка, 2011. — 256 с.

7. *Виттих В.А.* Мультиагентные модели взаимодействия для построения сетей потребностей и возможностей в открытых системах / В.А. Виттих, П.О. Скобелев // Автомат. и телемех. — № 1. — 2003. — С. 177–185.

8. *Матюшин М.М.* Построение оценочной функции для поддержки принятия оперативных решений при контроле параметров состояния космического аппарата / М.М. Матюшин, Х.В. Саркисян // Наука и образование: электронное научно-техническое издание — 2011. — № 4.

9. *Прокопчук Ю.А.* Принцип предельных обобщений: методология, задачи, приложения. Монография / Ю.А. Прокопчук. — Днепрпетровск: ИТМ НАНУ и НККАУ, 2012. — 384 с.

10. *Прокопчук Ю.А.* Модели когнитивных архитектур и процессов на основе парадигмы предельных обобщений / Ю.А. Прокопчук // Кибернетика и вычисл. техника. — 2013. — Вып. 171. — С. 37–51.

11. *Прокопчук Ю.А.* Спиральная когнитивная метадинамика: пути реализации в информационных технологиях / Ю.А. Прокопчук: Міжвузівський зб. наук. пр. «Проблеми інформаційних технологій». — 2013. — № 01 (013). — С. 6–17.

12. *Прокопчук Ю.А.* Технологический аудит и субоптимизация на основе принципа предельных обобщений / Ю.А. Прокопчук, А.С. Белецкий // Системні технології: Регіональний міжвузівський зб. наук. пр. — Вип. 2 (79). — Дніпропетровськ, 2012. — С. 82–87.

13. *Прокопчук Ю.А.* Системный синтез на основе парадигмы предельных обобщений в задачах интеллектуального управления подвижными антропоцентрическими объектами / Ю.А. Прокопчук: зб. наук. пр. міжнар. конф. «Сучасні інформаційні та інноваційні технології на транспорті» (Херсон, 28–30 травня 2013 р.). — Херсон: ХДМА, 2013. — Т. 1. — С.31–34.

Поступила в редакцию 15.01.2014