

## ФОРМАЛИЗАЦИЯ БИЗНЕС-ПРОЦЕССОВ НА ОСНОВЕ АППАРАТОВ РСА И $\pi$ -ИСЧИСЛЕНИЯ

© Илюшко В., Калмыков А., 2011

Розглядаються питання побудови формалізованих моделей виробничих і бізнес-процесів, які спроектовані на основі сервісно-орієнтованого підходу (SOA). Запропоновано досліджувати такі процеси на основі поділу на сильно- та слабопов'язані складові. Такий підхід дозволяє для формалізованого представлення процесів застосувати апарат алгебри алгоритмів та інструментарій  $\pi$ -обчислення, який призначений для опису взаємодії паралельних процесів. Запропонована методика дає змогу одержати компактний формалізований опис виробничих процесів, який забезпечує однозначну відповідність графічним моделям та комп'ютерному представленню мовою програмування BPEL.

**Ключові слова:** регулярні схеми алгоритмів,  $\pi$ -обчислення, BPEL, бізнес-процес, алгоритмічна модель.

The paper considers designing questions of the formalized models industrial and business processes which are constructed on the basis of the service-oriented approach (SOA). The researching of the given processes is offered on the basis of sharing on strong and loosely jointed components. Such approach allows to apply to the formalized representation of such processes the algebra of algorithms and  $\pi$ -calculus, which is intended for the description of interaction of concurrent processes. The offered technique allows to receive the compact readable formalized description of production and business-processes unambiguously corresponding to the computer representation in language BPEL as well as to the graphic models.

**Keywords -** the regular circuits of algorithms,  $\pi$ -calculus, BPEL, business process, algorithmic model.

### Введение

Развитие средств телекоммуникаций и удалённой обработки данных сделало возможным автоматизацию производственных и бизнес-процессов (БП), распределённых между множеством исполнителей. Такое изменение задач потребовало новых подходов к проектированию, тестированию, внедрению и сопровождению БП [1].

Современные методологии описания, моделирования и сопровождения различных производственных процессов ориентированы на языки графического представления с последующим отображением в исполняемый код, обрабатываемый специализированным программным обеспечением. Такая ситуация вызвана необходимостью в наглядном и доступном для заказчиков графическом описании производственных процессов, с одной стороны, и компьютерном представлении, пригодном для задач имитационного моделирования, практической автоматизации, с другой стороны. При этом в существующих нотациях графического представления и моделирования производственных и бизнес-процессов (BPMN,

UML, IDEFx) не предусмотрены стандартные форматы хранения данных [2, 3]. Поэтому для хранения графических моделей, как правило, используются различные несовместимые между собой внутренние форматы, а также компьютерное представление бизнес-процессов, например, на основе синтаксиса языка WSDL, базирующегося на схемах разметки документов XML, и известное как BPEL (Business Process Execution Language). Так как стандарт XML предназначен для компьютерного представления и хранения форм и шаблонов документов, для описания процессов и их взаимодействия между собой добавлены дополнительные классы активностей и сущностей. Фактически BPEL можно рассматривать как промежуточный подход к компьютерному представлению процессов, позиционируемый между декларативными и процедурными языками [4].

Использование конструкций языка WSDL, предназначенного для описания Web-услуг и, соответственно, распределённых процессов обработки данных, позволяет реализовать при помощи BPEL сервисно-ориентированный подход (SOA) в автоматизации бизнес-процессов. В настоящее время BPEL рассматривается как дальнейшее развитие универсальной нотации бизнес-процессов BPMN и позволяет описывать интерактивные процессы с непосредственным участием человека. При этом, обеспечивая перенос представления производственных процессов между различными программными платформами моделирования и проектирования производственных процессов, современные средства компьютерного представления, в частности язык BPEL, не поддерживают прозрачные технологии восстановления наглядных графических моделей из исполняемого кода, что значительно усложняет задачи внедрения и сопровождения производственных процессов [5].

#### **Постановка задачи**

Восстановление исходного графического представления процессов из кода BPEL вызывает значительные затруднения ввиду синтаксических сложностей описания синхронного и асинхронного взаимодействия между различными частями процессов, интерактивными формами ввода информации. Данная задача во многом сопоставима с восстановлением из мнемокода алгоритмов, изначально написанных на языках программирования высокого уровня. Частичная обратимость компьютерного представления усложняет согласование и внедрение производственных процессов, делает невозможным прозрачный автоматический перенос описания производственных процессов между различными программными средами. Всё это приводит к значительному увеличению трудоёмкости технологического цикла проектирования и реализации производственного процесса: от сбора требований, моделирования и до практического внедрения.

Следовательно, существует потребность в инструментарии формализованного представления процессов, который допускает обратимые преобразования представлений процессов из графического описания в исполняемый код и наоборот. При этом необходимо отметить, что современные методологии описания процессов ориентированы на их разделение на сильносвязанные, описываемые традиционными алгоритмическими подходами, и слабосвязанные параллельно выполняющиеся процессы. Так, производственные процессы в BPEL описываются последовательностью атомарных активностей, связанных условными или безусловными переходами. При этом взаимодействие между слабосвязанными и фактически параллельно выполняющимися в реальном времени процессами описывается посредством синхронного или асинхронного обмена информационными сообщениями.

Таким образом, инструментарий формализованного представления должен позволять описывать не только алгоритмы как логику и последовательность выполнения операций, но и поддерживать представление схем информационного обмена между слабосвязанными и параллельно выполняющимися процессами.

### **Существующие инструменты формализованного описания БП**

Отметим, что ограниченные возможности использования графических моделей BPMN и компьютерного представления BPEL при описании производственных процессов является общеизвестным фактором, причиняющим значительные неудобства при реальном проектировании и внедрении. Это привело к возникновению такого смешанного подхода представления процессов, как BPELJ (BPEL with Java), в которых сделана попытка совместить возможности описания слабосвязанных распределённых процессов и представления производственных процессов при помощи объектно-ориентированного языка программирования. Однако в целом предложенный подход не позволяет создать универсальное представление производственного процесса, обладающее достаточной наглядностью и краткостью, доступное для восприятия и понимания, как заказчиками, так и экспертами-разработчиками [6].

В то же время известны инструментарию формализованного представления процессов и алгоритмов, характеризующиеся лаконичностью и однозначностью описания:

– описание процессов как последовательности операций или состояний на основе использования регулярных выражений. Яркими представителями этой группы подходов к формализованному описанию процессов являются аппараты регулярных выражений алгебры событий и регулярных схем алгоритмов (РСА) [7, 8];

– представление процессов как совокупности информационных сигналов между различными состояниями с соблюдением последовательности входа и выхода [9, 10, 11]. К данной методологии относятся методики, основанные на использовании сетей Петри, а также разработанный в конце двадцатого века аппарат  $\pi$ -исчисления, рассматривающий процессы как сущности, взаимодействующие посредством информационных сигналов.

Первый из методов описания алгоритмов обладает высоким уровнем наглядности и читабельности, однако не позволяет представлять взаимодействие независимых процессов. К его основным преимуществам следует отнести компактность представления алгоритмов, возможность представления их структуры, что особенно важно для многоуровневых вложенных процессов. Однако, описание удалённых слабосвязанных процессов вызывает определённые трудности, так как взаимодействие таких объектов нельзя описать строгой алгоритмической последовательностью операций.

Второй метод, предназначенный для описания взаимодействия слабосвязанных процессов и схем обмена информационными сигналами между различными независимыми процессами и алгоритмами, наглядно не отражает особенности структуры алгоритмов, вызывает затруднения при описании и наглядном представлении сложных многоуровневых процессов, описании жёстко определённых последовательностей операций.

Очевидно, что попытка создания инструмента описания производственных и бизнес-процессов, совмещающего достоинства двух формализованных моделей, была предпринята в рамках проекта BPELJ, но не стала полностью успешной ввиду сложности базовой Java-среды. Следовательно, перспективный инструментарию формализованного описания производственных и бизнес-процессов целесообразно формировать с учётом преимуществ вышеперечисленных подходов к описанию производственных процессов.

### **Декомпозиция распределённых производственных процессов**

В некоторых работах [9, 10] встречаются попытки описания процессов только лишь на основе инструментарию  $\pi$ -исчисления. К сожалению, при всей корректности такого представления его практическая целесообразность вызывает сомнения ввиду громоздкости и слабой читабельности. В то же время аппарат  $\pi$ -исчисления является удобным средством описания информационных сигналов между слабосвязанными или параллельными процессами. Поэтому корректное применение аппарата  $\pi$ -исчисления возможно при условии корректной декомпозиции распределённого производственного процесса на такие слабосвязанные компоненты.

В работе [7] предлагается методология проектирования сложных технических систем, к ключевым принципам которой отнесено иерархическое представление системы в контексте важнейших аспектов (страт). В том числе рассматривается проектирование по алгоритмической и инфологической стратам, которые, по сути, описывают процессы функционирования системы. Используя предлагаемые подходы иерархического представления систем, выполняется декомпозицию процесса на уровни иерархии:

$$\text{Sys} \rightarrow \text{UnSys} \rightarrow \text{GrEl} \rightarrow \text{El}, \quad (1)$$

где  $\text{Sys}$  – уровень системы (весь производственный процесс в целом);  $\text{UnSys}$  – уровень подсистем (совокупность групп технологических операций, имеющая идентифицированные входы и выходы уровня подсистемы);  $\text{GrEl}$  – уровень группы элементов (группа технологических операций, имеющая идентифицированные входы и выходы);  $\text{El}$  – уровень элементов (отдельная технологическая операция).

Компоненты верхнего уровня иерархии распадается на несколько взаимосвязанных компонент нижнего уровня (рис. 1). При этом на каждом уровне иерархии следует различать алгоритмически жёстко (сильно) связанные составляющие и слабосвязанные, выполняющиеся независимо. Такое разделение компонент процесса соответствует принципам, заложенным в основу современной редакции нотации описания бизнес-процессов BPMN 2.0 [12].

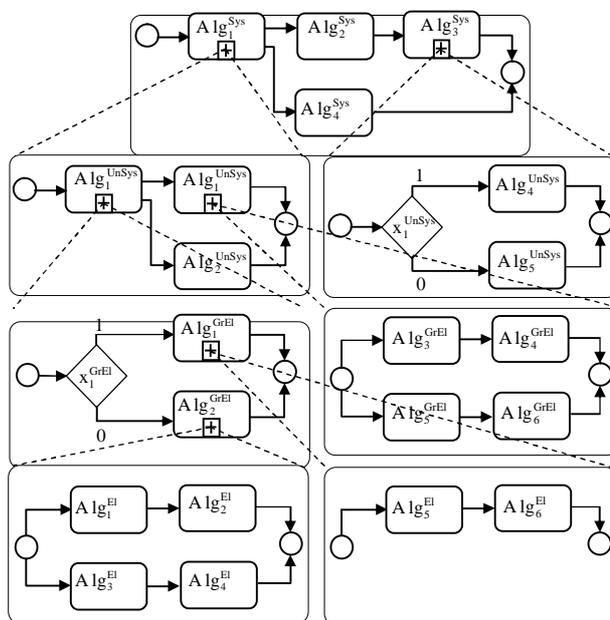


Рис. 1. Декомпозиция процесса на уровни иерархии алгоритмов

Прежде всего следует определить принципы разделения процессов и их компонентов на сильно- и слабосвязанные. Полагаем, что в случае, если не происходит передачи управления компоненту, а осуществляется только одно- или двухсторонний обмен данными между компонентами, то такие компоненты можно определить как слабосвязанные, принадлежащими разным владельцам. В нотации BPMN такая ситуация соответствует информационному взаимодействию между процессами, выполняемыми разными исполнителями. Пример такой декомпозиции с учётом разделения на жёстко- и сильносвязанные составляющие приведен на рис. 2. Взаимодействие слабосвязанных процессов в направлении от  $i$ -го к  $j$ -му на  $h$ -м уровне иерархии описывается информационным каналом  $a_{ij}^h$ . В данном случае общий производственный процесс распределён между несколькими исполнителями, например, процесс  $\text{Alg}_5^{\text{Sys}}$  принадлежит другому исполнителю (подразделению). При этом на рис.2 в целях сохранения наглядности многоуровневого представления общего процесса не использованы стандартные инструменты BPMN для описания принадлежности различным исполнителям (атрибут Lane).

## Особенности аппаратов РСА и $\pi$ -исчисления при описании производственных процессов

Использование  $\pi$ -исчисления для описания слабосвязанных процессов основывается на модели параллельных вычислений, взаимодействующих посредством обмена сообщениями. При этом любой алгоритм или процесс представляется как последовательность посылки и приёма информационных сообщений между отдельными составляющими. Предметом сообщений могут быть не только обрабатываемые данные, но и информация о структуре и характеристиках процессов. Данное обстоятельство позволяет применять  $\pi$ -исчисления для представления сложных алгоритмов, использующих множественные вызовы и динамическое изменение состава процедур, описания компонентов бизнес-процессов, взаимодействующих с внешними носителями информации. Это могут быть Web-агенты ввода данных, интерфейсы внешних репозиториях и баз данных.

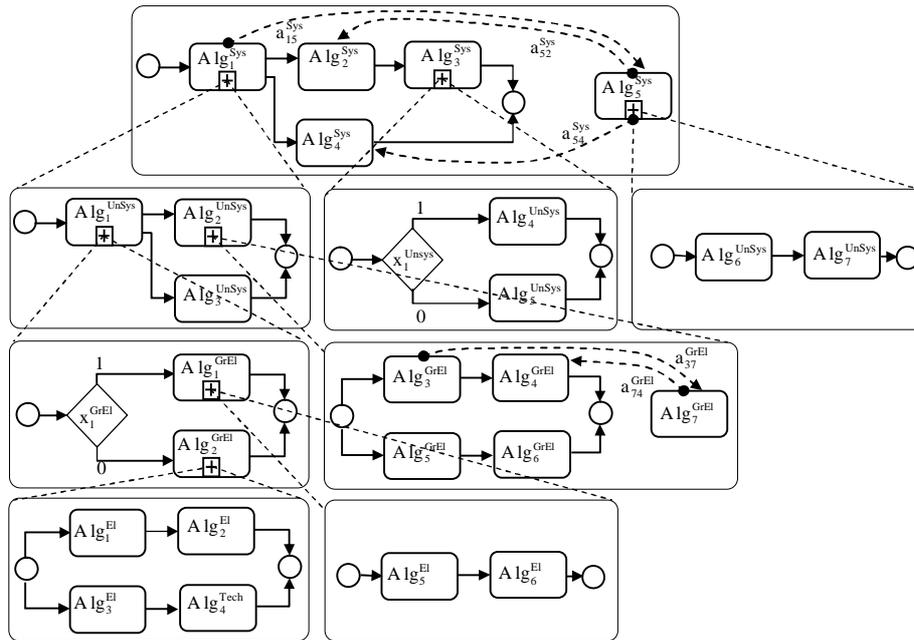


Рис. 2. Декомпозиция процесса с учётом разделения  
на сильно- и слабосвязанные компоненты

Примитивными сущностями  $\pi$ -исчисления являются имена, записываемые как символьные строки, начинающиеся со строчной буквы:  $x, y \in X$  [13].

Процесс или агент  $P$  (выражение  $\pi$ -исчисления) представляет собой одно из следующего списка:

1.  $c(x)P$  – входной префикс, получение данных  $x$  из канала  $c$  непосредственно перед выполнением процесса  $P$ ;
2.  $\bar{c}y.P$  – выходной префикс, передача данных  $y$  по каналу  $c$  после выполнения процесса  $P$ ;
3.  $P_1|P_2$  – совместное последовательное выполнение двух процессов, при котором выходные сигналы первого процесса синхронизированы с входными сигналами второго и/или наоборот;
4.  $P_1 + P_2$  – выполнение одного из двух процессов (бинарное сложение), используется для описания условных переходов;
5.  $(c)P$  – ограничение взаимодействия по каналу  $c$ , разрешено только для компонентов процесса  $P$ ;
6.  $\tau$  – отдельное действие, не влияющее на процесс  $P$ ;
7.  $0$  – пустой процесс.

Тогда взаимодействие слабосвязанных процессов, показанных на рис. 2, запишется как:

$$\begin{aligned}
 & \overline{a_{15}^{Sys} x \cdot A Ig_1^{Sys}} \left| a_{15}^{Sys} (x) \cdot A Ig_5^{Sys} \text{ для } A Ig_1^{Sys} \text{ и } A Ig_5^{Sys}; \right. \\
 & \overline{a_{54}^{Sys} x \cdot A Ig_5^{Sys}} \left| a_{54}^{Sys} (x) \cdot A Ig_4^{Sys} \text{ для } A Ig_5^{Sys} \text{ и } A Ig_4^{Sys}; \right. \\
 & \overline{a_{52}^{Sys} x \cdot A Ig_5^{Sys}} \left| a_{52}^{Sys} (x) \cdot A Ig_2^{Sys} \text{ для } A Ig_5^{Sys} \text{ и } A Ig_2^{Sys}; \right. \\
 & \overline{a_{37}^{GrEl} x \cdot A Ig_3^{GrEl}} \left| a_{37}^{GrEl} (x) \cdot A Ig_7^{GrEl} \text{ для } A Ig_3^{GrEl} \text{ и } A Ig_7^{GrEl}; \right. \\
 & \overline{a_{74}^{GrEl} x \cdot A Ig_7^{GrEl}} \left| a_{74}^{GrEl} (x) \cdot A Ig_4^{GrEl} \text{ для } A Ig_7^{GrEl} \text{ и } A Ig_4^{GrEl} \right.
 \end{aligned} \tag{2}$$

Представление взаимодействия других компонентов производственного процесса только на основе аппарата  $\pi$ -исчисления потребует описания второстепенных информационных сигналов, что значительно усложнит формализованное описание процесса и делает его громоздким и нечитабельным.

В работе [7] предложено для компактного и функционального полного описания алгоритмов использовать язык регулярных схем алгоритмов (РСА). В контексте описания бизнес-процессов под операторами  $y_i$  понимают структурные элементы и компоненты: простые и составные операции. Для связи между операторами  $y_i$  в РСА используют следующие базовые операции:

- - умножение  $\dot{y}$ , обозначает последовательное выполнение операторов;
- - конъюнкция  $\hat{y}$ , обозначает параллельное выполнение операторов;
- - дизъюнкция  $\check{y}$ , обозначает условное выполнение операторов;
- - итерация  $y^*$ , обозначает циклическое выполнение операторов.

На основе аппарата РСА строятся алгоритмические и информационные модели, которые описывают последовательность передачи управления и выполнения элементов бизнес-процесса. В общем случае модели проектируемых производственных и бизнес-процессов в РСА можно представить в виде [7]:

$$R = f(y_i, x_k, e, \otimes, 1, 0, \dot{y}, \hat{y}, \check{y}, y^*) \tag{3}$$

где  $y_i$  – базис основных операторов модели;  $x_k$  – условия переходов в модели;  $e$  – переход без выполнения основных операторов;  $\otimes$  – пустой оператор;  $1, 0$  – тождественно-истинное, тождественно-ложное условия;  $\dot{y}, \hat{y}, \check{y}, y^*$  – сигнатура базовых операций РССМ.

Тогда сильносвязанные компоненты бизнес-процесса, изображённого на рис.1, 2 запишутся в терминах РСА как:

$$\begin{aligned}
 R &= A Ig_1^{Sys} \cdot \left[ \left( A Ig_2^{Sys} \cdot A Ig_3^{Sys} \right) \wedge A Ig_4^{Sys} \right] = \\
 &= A Ig_1^{UnSys} \cdot \left[ A Ig_2^{UnSys} \wedge A Ig_3^{UnSys} \right] \cdot \left[ A Ig_2^{Sys} \cdot_{x_1^{UnSys}} \left( A Ig_4^{UnSys} \vee A Ig_5^{UnSys} \right)^{x_1^{UnSys}} \wedge A Ig_4^{Sys} \right] = \\
 &=_{x_1^{GrEl}} \left( A Ig_1^{GrEl} \vee A Ig_2^{GrEl} \right)^{x_1^{GrEl}} \cdot \left[ \left[ \left( A Ig_3^{GrEl} \cdot A Ig_4^{GrEl} \right) \wedge \right] \wedge A Ig_3^{UnSys} \right] \cdot \\
 &\left[ \left( A Ig_2^{Sys} \cdot_{x_1^{UnSys}} \left( A Ig_4^{UnSys} \vee A Ig_5^{UnSys} \right)^{x_1^{UnSys}} \right) \wedge A Ig_4^{Sys} \right] = \\
 &= \left( \left( A Ig_5^{El} \cdot A Ig_6^{El} \right) \vee \left[ \left( A Ig_1^{El} \cdot A Ig_2^{El} \right) \wedge \right] \right)^{x_1^{GrEl}} \cdot \left[ \left[ \left( A Ig_3^{GrEl} \cdot A Ig_4^{GrEl} \right) \wedge \right] \wedge A Ig_3^{UnSys} \right] \cdot \\
 &\left[ A Ig_2^{Sys} \cdot_{x_1^{UnSys}} \left( A Ig_4^{UnSys} \vee A Ig_5^{UnSys} \right)^{x_1^{UnSys}} \wedge A Ig_4^{Sys} \right]
 \end{aligned} \tag{4}$$

**Формализация описания производственных процессов  
в соответствии с сервисно-ориентированным подходом (SOA)**

В табл. 1 показано сопоставление базовых конструкций языка разметки документов BPEL, который фактически становится общепринятым стандартом компьютерного представления бизнес-процессов, с элементами аппаратов  $\pi$ -исчисления и PCA.

Таблица 1

**Сопоставление аппаратов формализованного описания с конструкциями BPEL**

Функция	Конструкция BPEL [12]	Конструкция $\pi$ -исчисления	Конструкция PCA
Последовательные операции	<pre>&lt;sequence ..."&gt; &lt;invoke name=" P<sub>1</sub> " ... /&gt; &lt;invoke name=" P<sub>2</sub> " ... /&gt; &lt;/sequence&gt;</pre>	$P_1   P_2$	$P_1 \cdot P_2$
Параллельные операции	<pre>&lt;flow ...&gt; &lt;invoke name=" P<sub>2</sub> " ... /&gt; &lt;invoke name=" P<sub>2</sub> " ... /&gt; &lt;/flow&gt;</pre>	$(P_1   P_2)$	$[P_1 \wedge P_2]$
Условный переход	<pre>&lt;if name=" x<sub>1</sub> "&gt; &lt;condition&gt; "1" &lt;/condition&gt; &lt;invoke name=" P<sub>1</sub> " ... /&gt; &lt;else&gt; &lt; invoke name=" P<sub>2</sub> " ... /&gt; &lt;/else&gt; &lt;/if&gt;</pre>	$P_1 + P_2$ в явном виде не указывается условие	$(P_1 \wedge P_2)^{x_1}$ $x_1$ – условие перехода
Цикл	<pre>&lt;repeatUntil&gt; &lt;sequence&gt; &lt;invoke name=" P<sub>1</sub> " ... /&gt; &lt;/sequence&gt; &lt;condition&gt; <math>x_1 \geq 0</math> &lt;/condition&gt; &lt;/repeatUntil&gt;</pre>	$!P_1$ в явном виде не указывается условие окончания	$x_1 \{P_1\}_{x_1}$ $x_1$ – условие окончания
Взаимодействие слабосвязанных процессов.	<pre>&lt;process name=" P<sub>1</sub> " &lt;partnerLink name=" c " partnerLinkType="исходящий" myRole=" P<sub>2</sub> " /&gt; &lt;/process&gt;  &lt;process name=" P<sub>2</sub> " &lt;partnerLink name=" c " partnerLinkType="входящий" myRole=" P<sub>1</sub> " /&gt; &lt;/process&gt;</pre>	$c(x).P_2$ – входящий префикс;  $\bar{c}y.P_1$ – исходящий префикс	нет

В табл. 1 показано, что аппарат PCA предпочтителен при описании последовательных, параллельных и условно ветвящихся процессов, в то время как аппарат  $\pi$ -исчисления позволяет представить в кратком и наглядном формализованном виде взаимодействие независимых или слабосвязанных компонентов процессов. Фактически при помощи аппарата  $\pi$ -исчисления

появляется возможность формализованного представления многократно вызываемых компонентов, оформленных в виде процедур, а также совместно используемых другими производственными процессами. Это соответствует принципам сервисно-ориентированного подхода (SOA), являющегося частью современной методологии построения интерактивных информационных систем поддержки производственных процессов [14]. Таким образом, композиция двух вышеописанных инструментариев, сохраняя их преимущества, позволит формализовать производственные процессы и обеспечит при этом однозначное обратимое соответствие полученного описания реализациям на языках программирования, в том числе и BPEL.

Выражения (5) – (8) являются формализованными представлениями взаимодействия сильно- и слабосвязанных составляющих для процесса, показанного на рис. 2, и композицией из (2) и (4) для уровней иерархии процессов Sys, UnSys, GrEl, El, соответственно.

$$R = \overline{a_{15}^{Sys} x} \cdot A Ig_1^{Sys} \left| a_{15}^{Sys} (x) \cdot A Ig_5^{Sys} \right. \cdot \left[ \overline{a_{52}^{Sys} x} \cdot A Ig_5^{Sys} \left| a_{52}^{Sys} (x) \cdot A Ig_2^{Sys} \cdot A Ig_3^{Sys} \right. \wedge \right. \\ \left. \left. \overline{a_{54}^{Sys} x} \cdot A Ig_5^{Sys} \left| a_{54}^{Sys} (x) \cdot A Ig_4^{Sys} \right. \right] = \right. \\ = \overline{a_{15}^{Sys} x} \cdot A Ig_1^{UnSys} \cdot [A Ig_2^{UnSys} \wedge A Ig_3^{UnSys}] a_{15}^{Sys} (x) \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] \cdot \quad (5)$$

$$\left[ \overline{a_{52}^{Sys} x} \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] a_{52}^{Sys} (x) \cdot A Ig_2^{Sys} \cdot_{x_1^{UnSys}} (A Ig_4^{UnSys} \vee A Ig_5^{UnSys})^{x_1^{UnSys}} \wedge \right. \\ \left. \overline{a_{54}^{Sys} x} \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] a_{54}^{Sys} (x) \cdot A Ig_4^{Sys} \right] = \\ = \overline{a_{15}^{Sys} x} \cdot A Ig_1^{UnSys} \cdot [A Ig_2^{UnSys} \wedge A Ig_3^{UnSys}] a_{15}^{Sys} (x) \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] \cdot \\ \left[ \overline{a_{52}^{Sys} x} \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] a_{52}^{Sys} (x) \cdot A Ig_2^{Sys} \cdot_{x_1^{UnSys}} (A Ig_4^{UnSys} \vee A Ig_5^{UnSys})^{x_1^{UnSys}} \wedge \right. \\ \left. \overline{a_{54}^{Sys} x} \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] a_{54}^{Sys} (x) \cdot A Ig_4^{Sys} \right] = \quad (6)$$

$$= \overline{a_{15}^{Sys} x} \left( \left[ \overline{a_{37}^{GrEl} x} \cdot A Ig_3^{GrEl} \left| a_{37}^{GrEl} (x) \cdot A Ig_7^{GrEl} \right. \cdot \right. \right. \\ \left. \left. \overline{a_{74}^{GrEl} x} \cdot A Ig_7^{GrEl} \left| a_{74}^{GrEl} (x) \cdot A Ig_4^{GrEl} \right. \wedge \right. \right. \left. \left. A Ig_5^{GrEl} \cdot A Ig_6^{GrEl} \right] \wedge A Ig_3^{UnSys} \right) \left| a_{15}^{Sys} (x) \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] \right. \\ \left. \left( A Ig_1^{GrEl} \vee A Ig_2^{GrEl} \right)^{x_1^{GrEl}} \right. \quad (7)$$

$$\left[ \overline{a_{52}^{Sys} x} \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] a_{52}^{Sys} (x) \cdot A Ig_2^{Sys} \cdot_{x_1^{UnSys}} (A Ig_4^{UnSys} \vee A Ig_5^{UnSys})^{x_1^{UnSys}} \wedge \right. \\ \left. \overline{a_{54}^{Sys} x} \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] a_{54}^{Sys} (x) \cdot A Ig_4^{Sys} \right] = \\ = \overline{a_{15}^{Sys} x} \cdot \left( \left[ \overline{a_{37}^{GrEl} x} \cdot A Ig_3^{GrEl} \left| a_{37}^{GrEl} (x) \cdot A Ig_7^{GrEl} \right. \cdot \right. \right. \\ \left. \left. \overline{a_{74}^{GrEl} x} \cdot A Ig_7^{GrEl} \left| a_{74}^{GrEl} (x) \cdot A Ig_4^{GrEl} \right. \wedge \right. \right. \left. \left. A Ig_5^{GrEl} \cdot A Ig_6^{GrEl} \right] \wedge A Ig_3^{UnSys} \right) \left| a_{15}^{Sys} (x) \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] \right. \\ \left. \left( A Ig_5^{El} \cdot A Ig_6^{El} \vee [A Ig_1^{El} \cdot A Ig_2^{El} \wedge A Ig_3^{El} \cdot A Ig_4^{El}] \right)^{x_1^{GrEl}} \right. \quad (8)$$

$$\left[ \overline{a_{52}^{Sys} x} \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] a_{52}^{Sys} (x) \cdot A Ig_2^{Sys} \cdot_{x_1^{UnSys}} (A Ig_4^{UnSys} \vee A Ig_5^{UnSys})^{x_1^{UnSys}} \wedge \right. \\ \left. \overline{a_{54}^{Sys} x} \cdot [A Ig_6^{UnSys} \wedge A Ig_7^{UnSys}] a_{54}^{Sys} (x) \cdot A Ig_4^{Sys} \right]$$

Такое представление позволяет кратко, наглядно и однозначно описать практически любой производственный процесс с учётом его технологического разделения на сильно- и слабосвязанные составляющие.

## Заключение

Предложенный подход к формальному описанию производственных процессов предоставляет возможность кратко и наглядно отразить производственные процессы, организуемые на основе различных принципов, в том числе сервисно- и процессно-ориентированные. Важным достоинством является однозначное обратимое соответствие между базовыми конструкциями аппарата  $\pi$ -исчисления, с одной стороны, и базовыми конструкциями языков разметки бизнес-процессов BPEL (WSDL) – с другой стороны. Данное обстоятельство позволяет в перспективе обеспечить обратимость между графическими моделями процессов и их компьютерной реализацией. Кроме того, получаемое формализованное описание производственных процессов обладает большей наглядностью и компактностью, чем конструкции на языке BPEL, большей строгостью представления и однозначностью восприятия по сравнению с графическими моделями.

Перспективное развитие предлагаемой методики состоит, по всей видимости, в следовании развивающимся стандартам описания производственных и бизнес-процессов, в более полном соответствии спецификации BPEL. Это может быть трактовано, как представление расширенных конструкций языка в виде выражений алгебры алгоритмов и аппарата  $\pi$ -исчисления, так и жесткое определение их соответствия элементам графических моделей, с тем, чтобы обеспечить сквозное функционально полное и прозрачное отображение процессов на уровнях “графическая модель – формализованная модель – компьютерная модель”.

1. Сеинс, Р. Сравнение четырёх ведущих методологий построения архитектуры предприятия [электронный ресурс] / Р. Сеинс. – Режим доступа : <http://msdn.microsoft.com/ru-ru/library/ee914379.aspx>. 2. Репин В. В. Сравнительный анализ нотаций ARIS/IDEF и продуктов их поддерживающих (ARIS Toolset/BPWin) [Электронный ресурс] / В.В. Репин. – Режим доступа : [http://www.iteam.ru/publications/it/section\\_51/article\\_2518](http://www.iteam.ru/publications/it/section_51/article_2518). 3. BPM-Xchange®. Технология моделирования и конвертирования бизнес моделей [электронный ресурс]. – Режим доступа : <http://www.lionsoft.ru/bpmx1.shtml> 4. Ландэ Д.В. Основы интеграции информационных потоков: Монография / Д.В. Ландэ – К. : Инжиниринг. – 2006. – 240с. 5. Silver В. BPMN-BPEL in Perspective / Bruce Silver [электронный ресурс]. – Режим доступа : <http://www.brsilver.com /2008/10/25/bpmn-bpel-in-perspective>. 6. Smith Н. Enough is enough in the field of BPM: We don't need BPELJ: BPML semantics are just fine Or is it back to the 3GLs? A response to a white paper by IBM and BEA entitled BPELJ: BPEL for Java, March 2004 / Howard Smith [электронный ресурс]. – Режим доступа : <http://www.fairdene.com/bpelj/BPELJ-Enough-Is-Enough.pdf>. 7. Илюшко В.М. Методы и модели информационной технологии проектирования метасистем: дис. д-ра техн. наук: 05.13.06 / В.М. Илюшко. – Харьков, 1998. – 451 с. 8. Чижухин Г.Н., Кулагин О.В. Синтез алгоритмов программ регулярными выражениями алгебры событий / Г.Н. Чижухин, О.В. Кулагин [электронный ресурс]. – Режим доступа : <http://systech.miem.edu.ru /2003/n1/Chizhuhin.htm>. 9. Рычков А.Ю. Язык описания бизнес-процессов, основанный на  $\pi$ -исчислении / А.Ю. Рычков // Математика программных систем : межвуз. сб. науч. ст. – Пермь : ПермГУ. – 2008. – М43. – с.69-76. 10. Маторин С.И.. Формализация моделей процессов на основе  $\pi$ -исчисления / С.И. Маторин, М.В. Михелев [электронный ресурс]. – Режим доступа : <http://dSPACE.bsu.edu.ru/bitstream/123456789/367/1/Matorin.S.I..pdf>. 11. Сеницын С.В., Хлытчиев О.И. Переход от IDEF0 диаграмм к  $\pi$ -исчислению / С.В. Сеницын, О.И. Хлытчиев // Научная сессия МИФИ. – М. : МИФИ. – 2008. – Том 11. – с.111-112. 12. Web Services Business Process. Execution Language Version 2.0. Primer, 9 May 2007 [Электронный ресурс] – Режим доступа : <http://docs.oasis-open.org/wsbpel/2.0/Primer/wsbpel-v2.0-Primer.pdf>. 13. Milner R. Communication and Mobile Systems: the  $\pi$ -calculus / Robert Milner – Cambridge : Cambridge University Press. – 1999. – 161 p. 14. Vasiliev Y. SOA and WS-BPEL. Composing Service-Oriented Solutions with PHP and ActiveBPEL / Yuli Vasiliev. – Birmingham : Packt Publishing Ltd. – 2007. – 301 p.