

# Представление BPMN-диаграмм на языке темпоральной логики линейного времени

Яременко Ф. В.

Вычислительный центр им. А.А. Дородницына РАН

2014 г.

# Валидация, Верификация

Процесс разработки сложных и масштабных систем подвержен ошибкам.

**Валидация** (от лат. validus - здоровый) - проверка того, что разработано именно то, что требуется. Обычно тестирование и моделирование разработанной системы рассматриваются как часть валидации.

**Верификация** (от лат. verus - верный) - проверка того, что продукт удовлетворяет сформулированным требованиям.

**Формальная верификация системы** - приемы и методы формального доказательства (или опровержения) того, что модель системы удовлетворяет заданной формальной спецификации. Применение формальной верификации позволяет проверить, что все возможные запуски потока работ будут удовлетворять желаемым свойствам.

# Формальная верификация

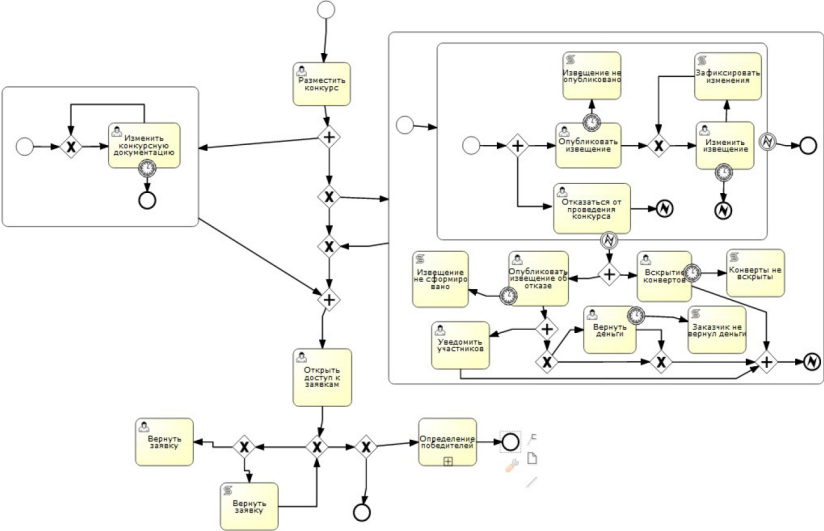
Как правило, верификация используется для проверки пяти свойств потока работ:

1. Отсутствие конфликтов постусловий: существуют ли задачи, выполняющиеся параллельно, с противоречивыми пост-условиями (эффектами)?
2. Отсутствие конфликтов пред- и пост-условий: возможна ли ситуация, что пост-условие предыдущей задачи противоречит предусловию последующей?
3. Достижимость: существуют ли задачи, которые никогда не выполняются.
4. Выполнимость: существуют ли задачи, для которых не будет выполняться предусловие в момент запуска задачи?
5. Устойчивость - проверка этого условия используется для обнаружения ошибок, таких как тупики, отсутствие синхронизации и другие дефекты.

**BPMN (Business Process Model and Notation)** - широко распространённым графической нотацией моделей потоков работ.

BPMN сочетает в себе элементы ряда предложенных ранее нотаций для моделирования бизнес-процессов, в том числе XPDЛ (Process Definition Language) и диаграммы деятельности UML (Unified Modeling Language).

# BPMN



# Верификация BPMN

Графо-ориентированные языки описания потоков работ, подобные BPMN, были формально изучены. Известно, что модели, описанные в этом семействе языков подвержены ряду ошибок, таких как тупики и активные блокировки.

BPMN увеличивает число возможных ошибок, так он совмещает графо-ориентированный подход с функциями, заимствованными из различных источников, например BPEL - стандарта, описывающего поток работ на уровне исполнения. Эти функции дают возможность определять:

- ▶ подпроцессы, которые могут выполнены несколько раз одновременно
- ▶ подпроцессы, которые могут быть прерваны в результате исключения
- ▶ потоки сообщений между процессами

Совместное использование этих функций является источником дополнительных ошибок.

# Верификация BPMN

Разнородность конструкций BPMN и отсутствие однозначного определения обозначений затрудняют анализ моделей BPMN. В то время как синтаксические правила подробно описаны в стандарте BPMN, семантика исполнения задана в повествовательной форме, используя иногда противоречивую терминологию. Строгое определение семантики исполнения является необходимым требованием для формальной верификации BPMN-диаграмм.

Методы и формализмы, используемые для формального представления и верификации BPMN-диаграмм:

- ▶ Сети Петри
- ▶  $\pi$ -исчисления
- ▶ CSP (Communicating Sequential Processes)
- ▶ Темпоральные логики
- ▶ ...

# Верификация BPMN

Предлагается расширение работы **Lam V. S. W. "A precise execution semantics for bpmn 2012** по следующим причинам:

- ▶ Полностью охвачена нотация BPMN 1.2, с возможностью расширения данного подхода для описания BPMN 2.0
- ▶ Дальнейшая работа направлена на формальную верификацию с использованием знаний о предметной области и описаний операций над данными. В этом случае использование темпоральных дескриптивных логик (tDL) или логик линейного времени первого порядка (LTL-FO) позволяет на одном языке описать и поведенческую семантику BPMN-диаграмм, и представление знаний из предметной области и субъектов, участвующих в потоке работ.



# LTL

LTL (Linear Temporal Logic) - темпоральная логика линейного времени

Темпоральные операторы:

$\mathbf{X}\phi$  - neXt -  $\phi$  будет истинно в следующий момент времени

$\mathbf{F}\phi$  - Future -  $\phi$  будет истинно когда-нибудь в будущем

$\mathbf{G}\phi$  - Globally -  $\phi$  будет истинно всегда в будущем

Формула LTL:

$\phi ::= p \mid \neg\phi \mid (\phi) \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \mid \mathbf{X}\phi \mid \mathbf{F}\phi \mid \mathbf{G}\phi \mid \phi \mathbf{U}\phi$

# BPMN → LTL

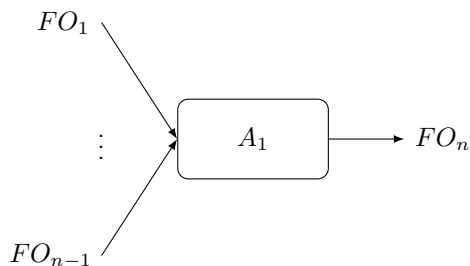
Предложенное расширение работы **Lam V. S. W. "A precise execution semantics for bpmn 2012:**

1. Пред- и пост-условия для запуска и завершения процессов
2. Решена проблема окружения (frame problem) - проблема представления всего того, что остаётся неизменным

# BPMN $\rightarrow$ LTL

$tokenAt_x$	токен находится на элементе диаграммы $x$
$tokenAt_{x_1, x_2}$	токен находится на потоке управления, идущем от элемента $x_1$ к элементу $x_2$
$F_{EE}^{Term}$	множество событий останова
$tokenConsumedBy_{TE}$	токен достиг события останова
$S_F$	множество объектов потока управления
$C_{SF}$	множество соединяющих объектов
$F_T$	множество заданий

# BPMN → LTL



$A_1 \in F_T$ ,  $FO_i \in S_F$  и  $TE_k \in \overline{F_{EE}^{Term}}$  для  $i = 1, \dots, n$  и  $k = 1, \dots, m$ ,  
 $(FO_j, A_1), (A_1, FO_n) \in C_{SF}$  для  $j = 1, \dots, n - 1$ , тогда

$$\mathbf{G} \left( \bigwedge_{i=1}^{n-1} \left( (tokenAt_{FO_i, A_1} \wedge \bigwedge_{k=1}^m \neg tokenConsumedBy_{TE_k}) \rightarrow \right. \right. \\
\left. \left. \mathbf{X} (tokenAt_{A_1} \wedge \neg tokenAt_{FO_i, A_1}) \right) \right) \wedge \\
\left( (tokenAt_{A_1} \wedge \bigwedge_{k=1}^m \neg tokenConsumedBy_{TE_k}) \rightarrow \mathbf{F} (tokenAt_{A_1, FO_n}) \right)$$

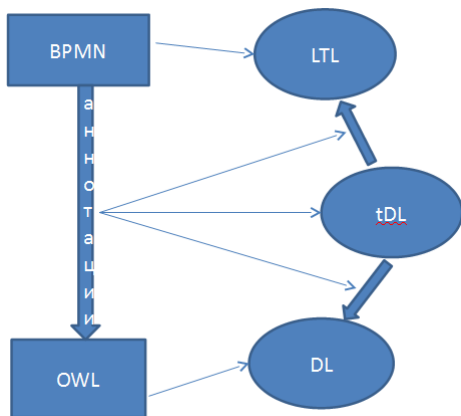
## BPMN $\rightarrow$ LTL

Недостатком этого утверждения является то, что  $tokenAt_{A_1, FO_n}$  может принимать истинное значение в любой момент времени, т.е. не решена проблема фреймов. Добавить ограничение:

$$\mathbf{G} \left( \left( \mathbf{X} tokenAt_{A_1} \rightarrow \left( \bigwedge_{k=1}^m \neg tokenConsumedBy_{TEK} \wedge \right. \right. \right. \\ \left. \left. \left. \left( tokenAt_{A_1} \vee \bigvee_{i=1}^{n-1} tokenAt_{FO_i, A_1} \right) \right) \right) \wedge \right. \\ \left. \left( \mathbf{X} tokenAt_{A_1, FO_n} \rightarrow \left( tokenAt_{A_1} \wedge \bigwedge_{k=1}^m \neg tokenConsumedBy_{TEK} \right) \right) \right)$$

## Дальнейшая работа

Дальнейшая работа состоит в формальной верификации семантически-аннотированных диаграмм на основе tDL или LTL-FO



Вопросы?