

BPMN 2.0

Firemní proces

2

- Posloupnost aktivit vykonávaná jedním či více účastníky za účelem přinést organizaci nějakou hodnotu
- Proces může se může rozložit na sekvenci několika jednodušších aktivit
- Aktivity musí někdo (nebo něco) provést (účastník)
- Cílem je vytvořit pro organizaci nějakou hodnotu, ať už přímo či nepřímo

Typy firemních procesů

3

- Manažerské procesy
 - ▣ Řídí běh organizace
- Operační procesy
 - ▣ Tvoří vnitřní (core) aktivity organizace a vytvářejí primární přínosné hodnoty
- Podpůrné procesy
 - ▣ Podporují vnitřní operační procesy

Proč modelovat procesy

4

- Princip: **Nelze řídit to, čemu nerozumíte**
- Chápat a řídit aktuální procesy
 - ▣ Měření času, nákladů, zdrojů
- Zlepšovat současné procesy
 - ▣ Zefektivnění, identifikace chybějících kroků, racionalizace
- Návrh nových procesů
 - ▣ Realizace business požadavků novými procesy
- Zprostředkovat existující a nové procesy
 - ▣ Procesní modely jsou velmi efektivním způsobem jak komunikovat existující a nové procesy
- Automatizace procesů
 - ▣ Aplikace strojového zpracování procesů

BPMN struktura

5

- BPMN 2 dovoluje modelovat čtyři související aspekty business procesů:
 - Proces – sekvence aktivit tvořících business proces
 - Zaměření: sekvence aktivit a událostí
 - Spolupráce (collaboration) – proces, který má dva nebo více účastníků
 - Zaměření: sekvence aktivit a událostí a/nebo zpráv zasílaných mezi účastníky
 - Choreografie – posloupnost interakcí mezi účastníky
 - Zaměření: účastníci v business procesech a posloupnost jejich vzájemných interakcí
 - Konverzace – zachycuje B2B interakce

BPMN sumarizace

6

- 4 typy událostí (+různé druhy triggerů a catch&throw sémantiky)
- 4 typy aktivit (+jejich varianty)
- 5 typů rozhodovacích uzlů
- 3 typy propojovacích objektů (sequence flow, message flow, association)
- 2 artefakty (annotations, groups)
- 4 typy datových objektů
- Konverzace, choreografie, atd.

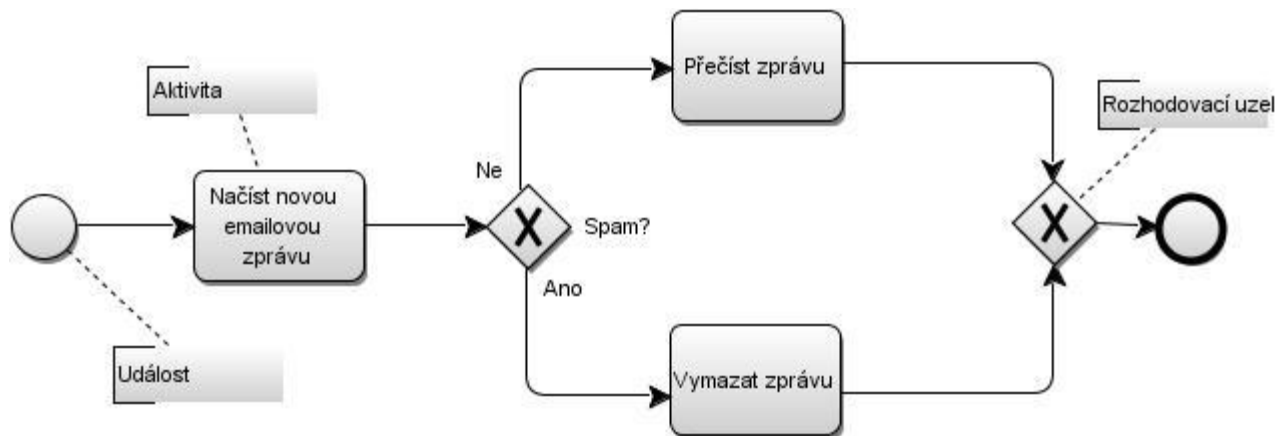
Procesní diagram

7

- BPMN proces – množina elementů sekvenčního toku, které mohou být propojeny sekvenčním tokem a dalšími spojovacími objekty a tvoří souhrnně určitý druh vývojového diagramu
- Objekty sekvenčního toku:
 - ▣ Události – něco co v průběhu procesu nastane
 - ▣ Aktivity – práce, která je během procesu vykonána
 - ▣ Rozhodovací uzly – řízení (sekvenčního) toku během procesu

Procesní diagram - příklad

8

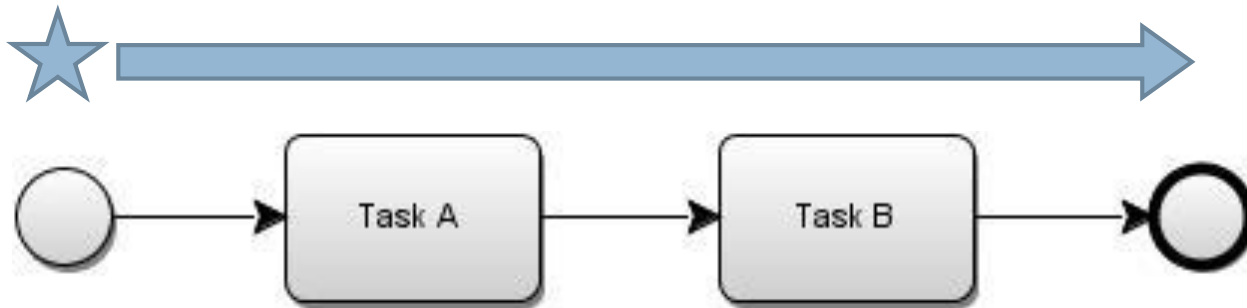


Procesní sémantika - tokeny

- Token – imaginární bod reprezentující aktuální okamžik průchodu grafem (procesním diagramem)
- Tokeny procházejí od zdrojového objektu k cílovému objektu prostřednictvím sekvenčního toku
- Objekt sekvenčního toku je aktivován (proveden) v okamžiku kdy je na jeho vstup přiveden token

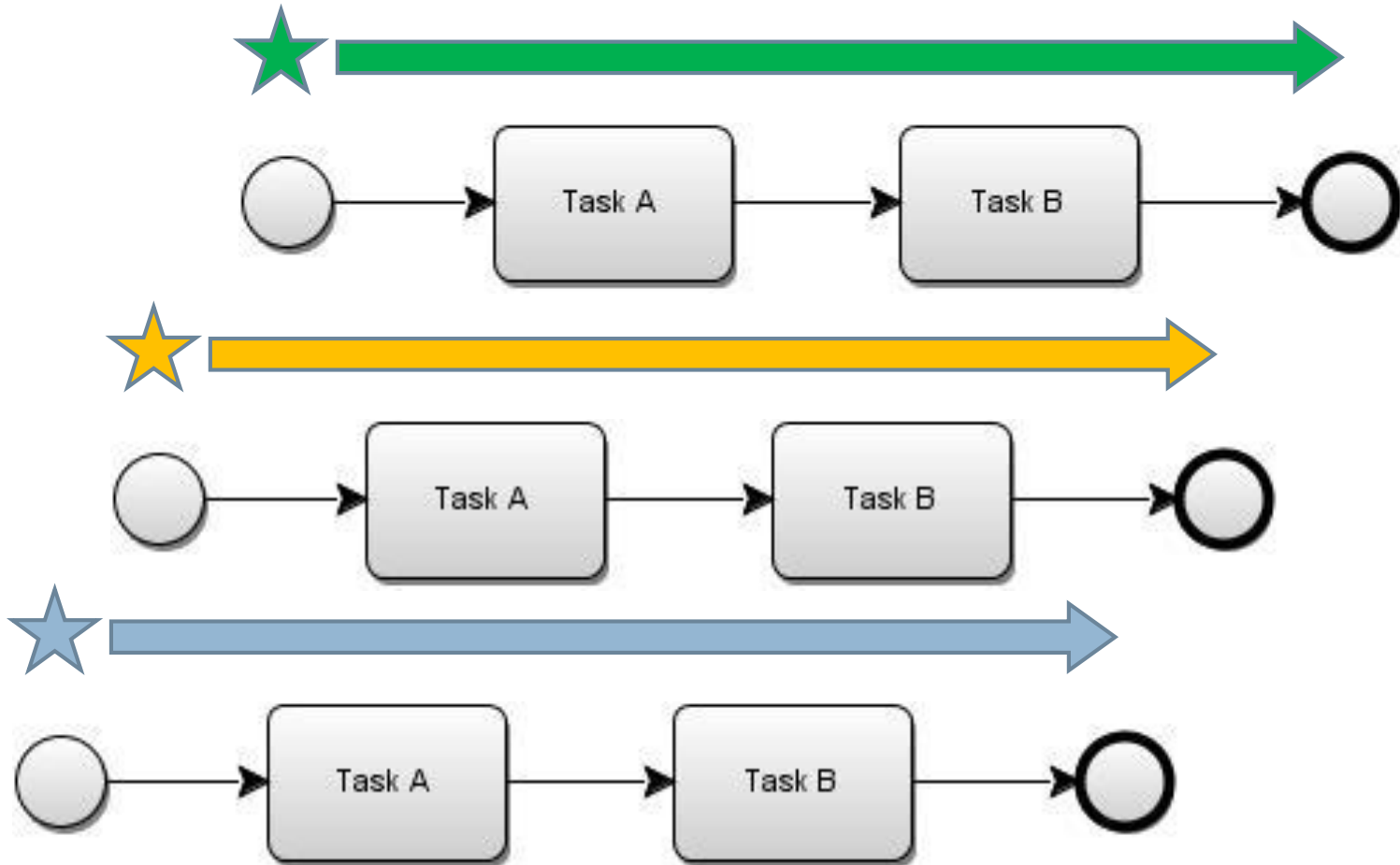
Postup tokenu diagramem

10



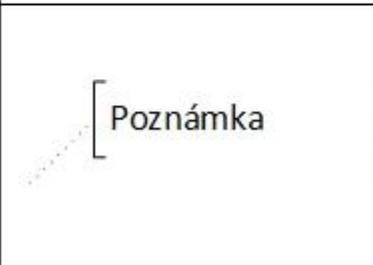
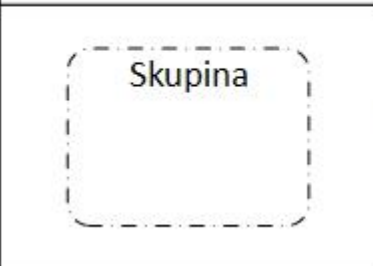
Instance procesů

11



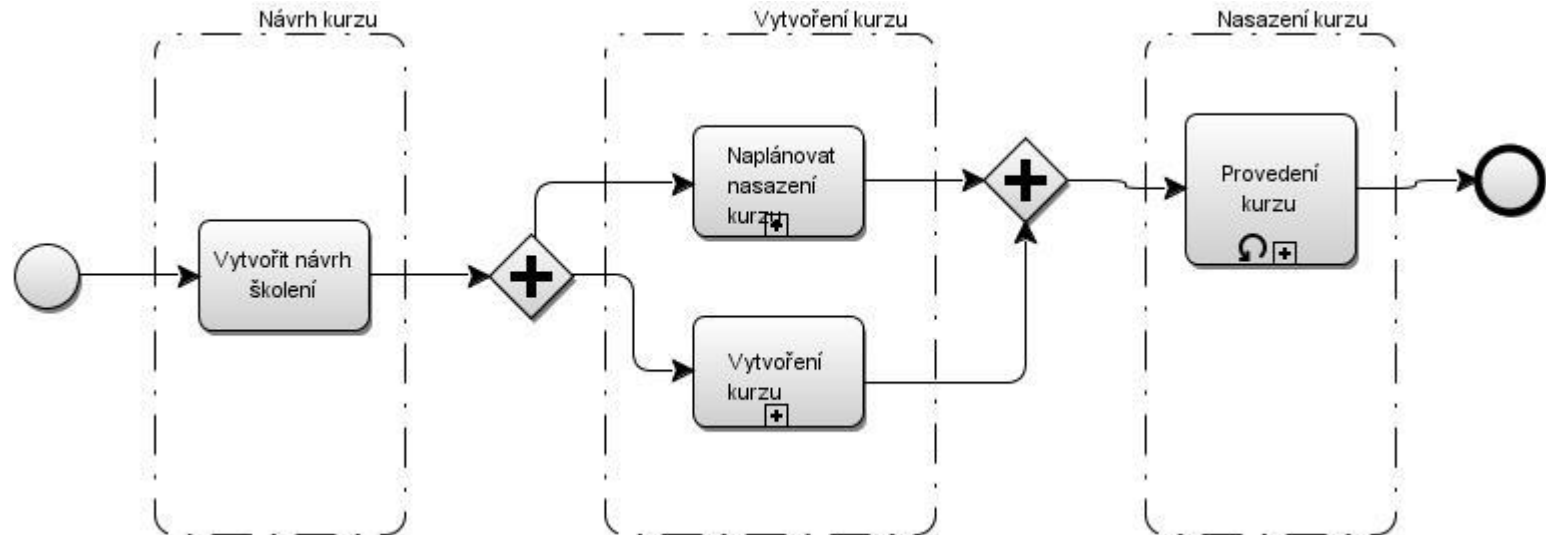
Komponenty diagramu - artefakty

12

ARTEFAKTY		
Syntax	Název	Sémantika
	Textová anotace (poznámka)	Zvyšují srozumitelnost diagramu, umožňují vkládat doplňující text. Pozn.: využívá se asociační vazba k propojení s objekty sekvenčního toku.
	Skupina	Dovoluje seskupovat dohromady elementy diagramu, slouží ke znázornění jejich vzájemného propojení

Použití artefaktů

13



BPMN diagram spolupráce

14

- Collaboration diagram – více než jeden účastník
- Pokud je pouze jeden kontext a žádné oddíly kontextu, jedná se o procesní diagram a rámeček kontextu může být vynechán

Knihovna	Správa katalogu knihovny	V tomto oddílu kontextu se zakresluje ta část procesu, která přísluší Správě katalogu knihovny
	Správa výpůjček	V tomto oddílu kontextu se zakresluje ta část procesu, která přísluší Správě výpůjček

Události

15

- Výskyt něčeho, co nastane při vykonávání procesu a je pro organizaci důležité z procesní perspektivy
- Mohou mít trigger (spouštěč) nebo result (výsledek)
 - indikuje symbol uvnitř události

Typy událostí

16

TYPY UDÁLOSTÍ			
Syntax	Ohraničení	Název	Sémantika
	Tenká čára	Startovací / vstupní událost	Zahajuje proces vysláním tokenu.
	Dvojitá čára	Vnitřní událost	Nastává během procesu.
		Vnitřní hraniční událost	Vnitřní události, které jsou napojeny na okraj aktivity (tasku či sub-procesu).
	Tlustá čára	Koncová / výstupní událost	Ukončuje procesní tok konzumací tokenu.

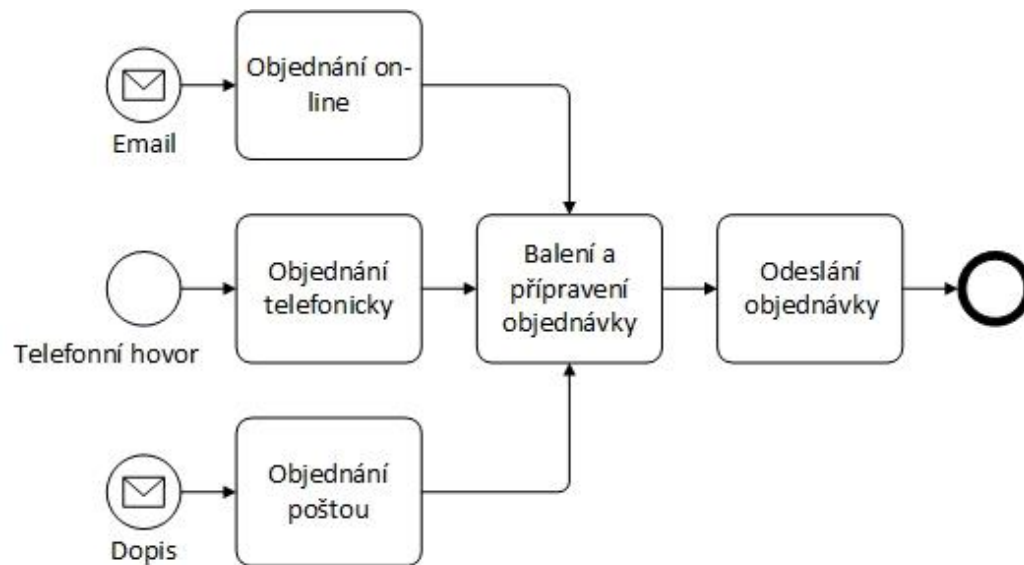
Základní události

17

- Tzv. „none events“ – nspecifikované události.
- Nemají upřesněný spouštěč
- Používají se tehdy, jestliže není známo co událost spouští (vyvolává)



- V případě více vstupních událostí – pouze jedna může být nespecifikovaná.



Nespecifikované vnitřní události

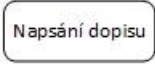
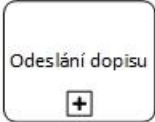
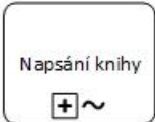
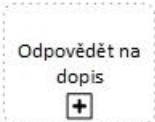
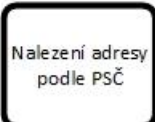

19

- Nemají trigger, spouštějí se okamžitě po získání tokenu
- Užívají se jako indikátor toho, že aktivita/proces dosáhly důležitého stavu nebo milníku
- Neměly by se užívat často!



Aktivity

20

TYPY AKTIVIT		
Syntax	Název	Sémantika
	Task	Aktivita, která je v rámci tohoto procesu považována za atomickou.
	Sub-proces (collapsed)	Aktivita, která může být hierarchicky dekomponována na další aktivity. (a.k.a. embedded sub-process)
	Ad-hoc sub-process (collapsed)	Sekvence aktivit je volena vykonavatelem procesu.
	Event sub-process (collapsed)	Sub-process, který je spouštěn zprávou, chybou, eskalací, kompenzací, podmínkou, signálem, časovou událostí nebo vícenásobnou událostí.
	Call activity	Vyvolává globální proces nebo task.
	Transaction sub-process (collapsed)	Reprezentuje sub-process, který má transakční vlastnosti.

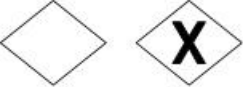




Aktivity – vícenásobné toky

21

- Aktivita může mít žádné nebo více vstupních vláken a žádné nebo více výstupní vlákno.
- Pokud má aktivita více vstupních vláken, spouští se při **každém** příchodu tokenu.
- Po dokončení vysílá aktivita tokeny **na všechna** odchozí vlákna.

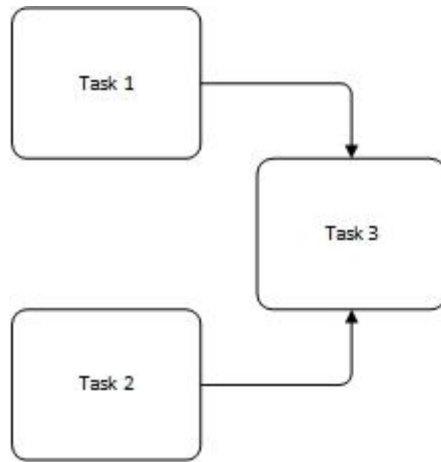
Rozhodovací uzly

22

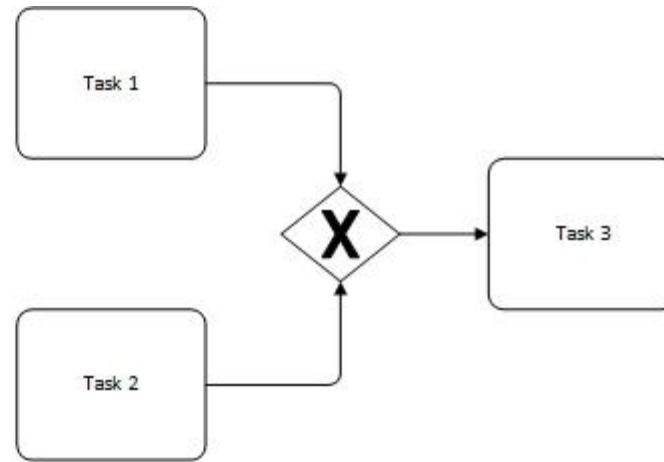
TYPY ROZHODOVACÍCH UZLŮ			
Syntax	Název	Konvergenční pozice (řízení vstupů)	Divergenční pozice (řízení výstupů)
	Exclusive gateway	Konzumuje příchozí token.	Vysílá jeden token na odchozí vlákno, jehož podmínka je splněna (True).
	Parallel gateway	Čeká na příchod tokenů na všech příchozích vláknech.	Vysílá token na všechna odchozí vlákna.
	Event based gateway	Konzumuje příchozí token.	Vysílá tokeny na všechny odchozí vlákna, která jsou napojena na vnitřní události. Událost spuštěná jako první předává token dále.
	Inclusive gateway	Spojuje tokeny z dřívějšího rozdělení inkusivním rozhodovacím uzlem.	Vysílá tokeny na všechna odchozí vlákna, jejichž podmínky jsou splněny (True).
	Complex gateway	Chování je determinováno podmínkou, která je specifikována v atributu activationCondition .	Determinováno podmínkami specifikovanými na odchozích sekvenčních tocích.

Explicitní a implicitní sloučení

23



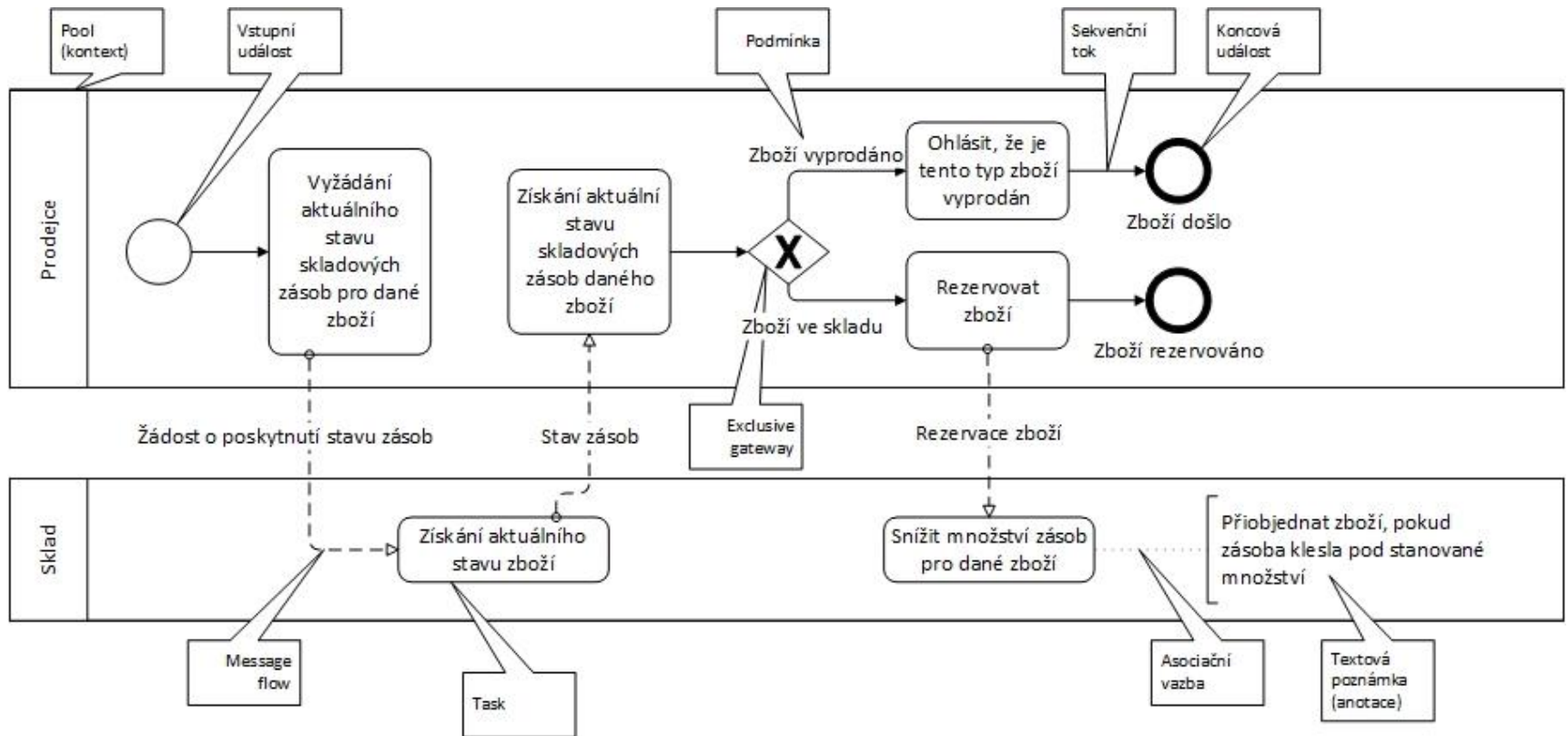
Implicitní sloučení



Explicitní sloučení

Příklad BPD

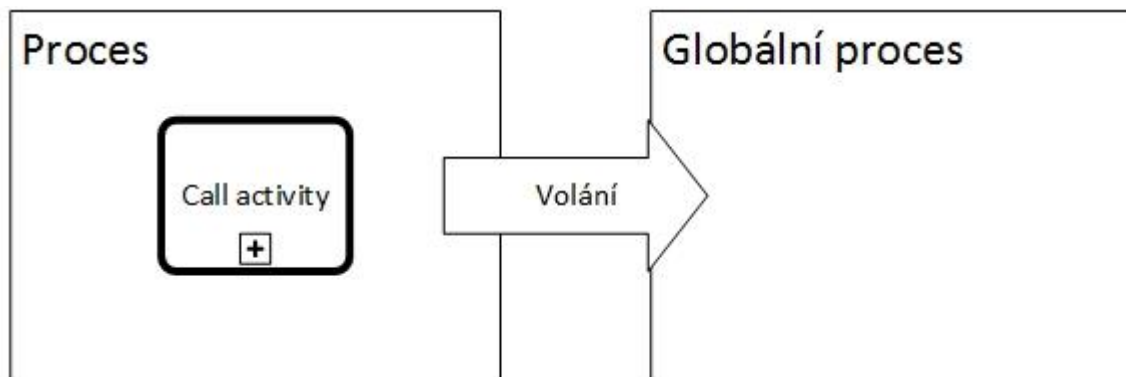
24



Aktivity – volání (Call Activities)







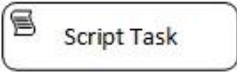

25

- Globální task/proces je znovupoužitelný task/proces, který může být volán jiným procesem využitím call aktivity.
- Call aktivity mají silné ohraničení.



Typy tasků

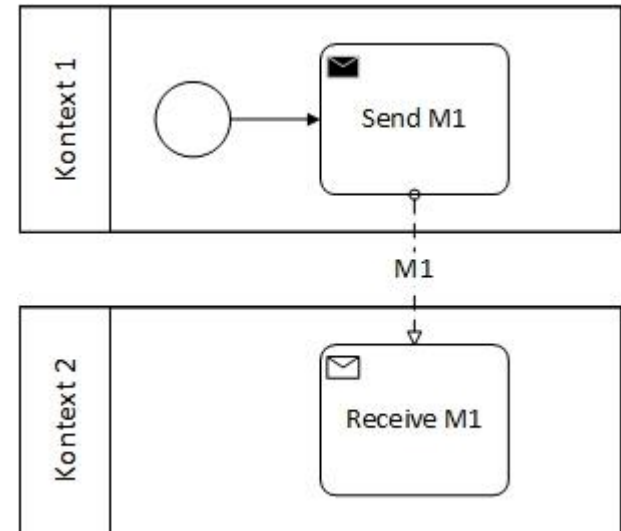
26



TYPY ÚKOLŮ (TASKS)		
Syntax	Sémantika	Globální?
 Task	Aktivita, která je v rámci konkrétního procesu považována za atomickou.	N
 Send Task	Po dokončení zasílá externímu účastníkovi zprávu.	N
 Receive Task	Čeká na zprávu od externího účastníka.	N
 Manual Task	Task, který je prováděn výlučně manuálně.	Y
 User Task	Task, který je prováděn člověkem s pomocí automatizace.	Y
 Service Task	Služba jako jsou webové služby nebo automatizované aplikace.	Y
 Script Task	Skript spuštěný Business Process Engine.	Y
 Business Rule Task	Task provedený Business Process Engine.	Y

Send a Receive

27

- Send task – po dokončení odešle zprávu po všech odchozích message flow šipkách.
- Receive task – čeká na zprávu, po skončení vysílá token na všech odchozích vláknech.



Syntax	Sémantika	Globální?
 Send Task	Po dokončení zasílá externímu účastníkovi zprávu.	N
 Receive Task	Čeká na zprávu od externího účastníka.	N

Manual a User



28

- Manual task – vykonává jej účastník bez jakékoliv formy strojového zpracování nebo automatizace
- User task – člověk realizuje task s pomocí nějakého SW systému

Service

29



- Service task – účastní se jej účastníci **zvnějšku** aktuálního procesu
 - Service – externí systém
 - User – externí uživatel
- Tento typ tasku **musí** být spouštěn vstupní událostí typu message!
- Po skončení je často odeslána zpráva zpět do procesu, který task vyvolal.

Syntax	Sémantika	Globální?
 User Task	Task, který je prováděn člověkem s pomocí automatizace.	Y
 Service Task	Služba jako jsou webové služby nebo automatizované aplikace.	Y

Script a Business Rule

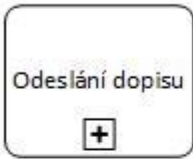
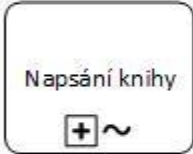


30

- Script – reprezentuje skript, který je spouštěn Business Process Engine
- Business Rule – poskytuje vstup nebo získává výstup z Business Rule Engine
 - ▣ BR – výrok, který definuje nějakou formu omezení pro chod organizace a vždy má hodnotu True/False

Syntax	Sémantika	Globální?
 Script Task	Skript spuštěný Business Process Engine.	Y
 Business Rule Task	Task provedený Business Rule Engine.	Y

Typy sub-procesů

31

TYPY SUB-PROCESŮ (SUB-PROCESS TYPES)		
Syntax	Název	Sémantika
	Sub-proces (collapsed)	Normální typ sub-procesu. Je umístěný v rodičovském procesu, je jím spouštěn, a může přistupovat ke všem jeho globálním datům.
	Ad-hoc sub-process (collapsed)	Sekvence aktivit je volena vykonavatelem procesu.
	Event sub-process (collapsed)	Sub-process, který je spouštěn zprávou, chybou, eskalací, kompenzací, podmínkou, signálem, časovou událostí nebo vícenásobnou událostí.
	Transaction sub-process (collapsed)	Reprezentuje sub-process, který má transakční vlastnosti. Realizuje se celý nebo vůbec; v případě selhání zanechává systém v konzistentním stavu.

Ad-hoc

32

- Pořadí vnořených aktivit určují vykonavatelé procesu
- Každá aktivita může být provedena 0-krát až n-krát

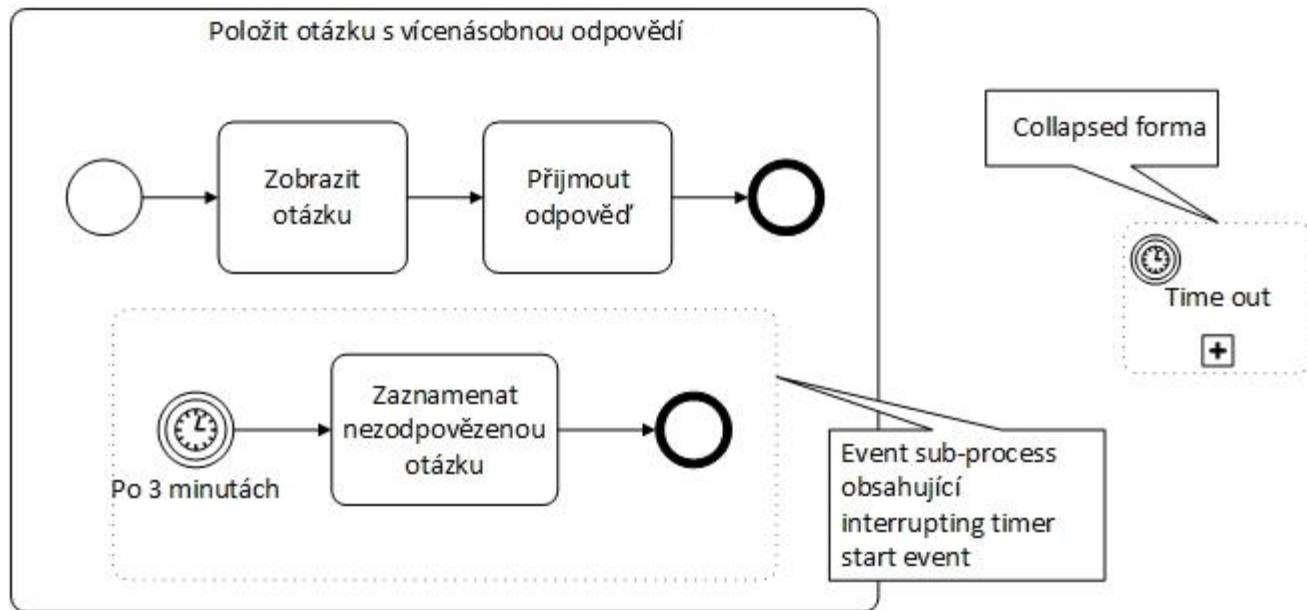
Event sub-proces

33

- Nejsou součástí standardního procesního toku
- Nemají vstupní/výstupní vlákna, nejsou částí rodičovského procesního toku!
- Jsou vždy spouštěny vstupní událostí s triggerem - message, error, escalation, compensation, conditional, signal, timer nebo multiple
- Pokud jsou spouštěny přerušující vstupní událostí, přerušují běh procesu, jinak běží souběžně
- **Musí** mít okraje zakresleny přerušovanou čarou

Event sub-proces

34



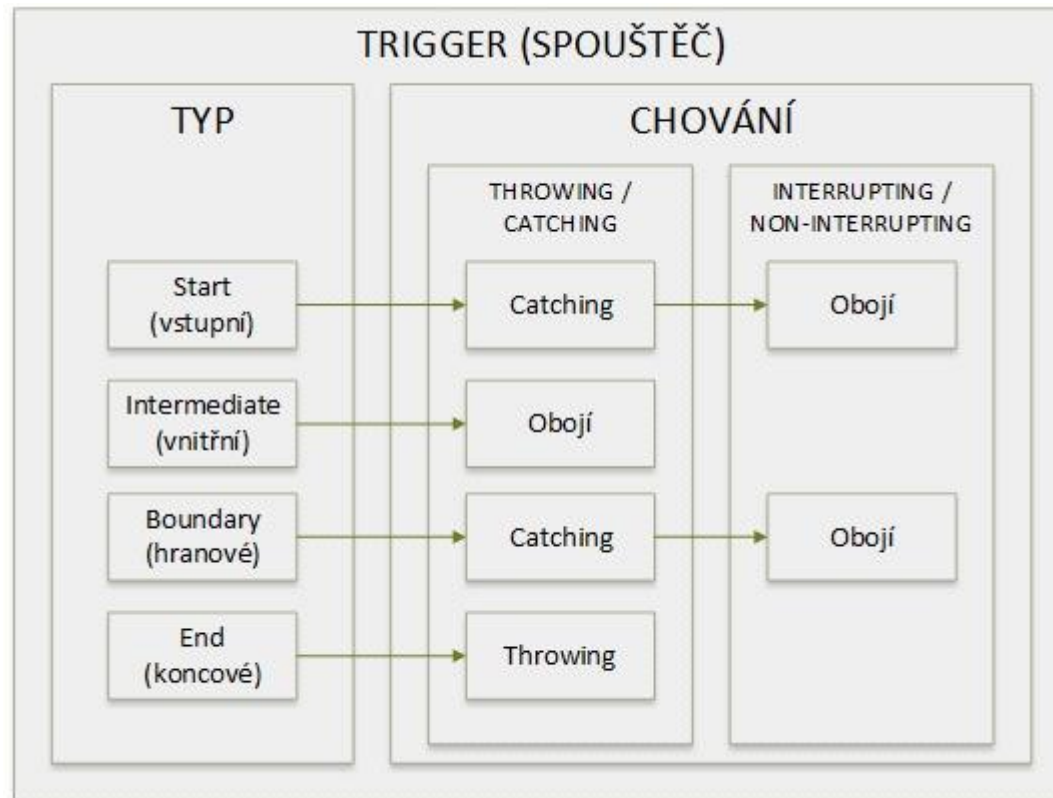
Události – dělení

35

- Trigger – určuje co způsobilo výskyt události
- Behavior – Throw nebo Catch – zda přerušují či nepřerušují obsaženou aktivitu
- Type – kde mohou být použity v rámci procesu

Události – dělení

36



Spouštěč události (Trigger)

37

- Popisuje, co událost ve skutečnosti dělá, např. zda zachytává či vysílá zprávu
- Typ triggeru ovlivňuje catch/throw sémantiku

Throw a Catch

38

- Throwing – čeká na token, pak vysílá trigger.
- Catching – událost čeká na trigger, pak vysílá token.
- Vstupní – pouze zachycují (catch)
- Koncové – pouze háží (throw)
- Intermediate – mohou obojí, záleží na triggeru
- Boundary – pouze zachycují (catch)



Interrupting a Non-interrupting

39

- Přerušující - interrupting – spuštění aktivitu ukončuje, procesní tok pokračuje dále z události, která zachycuje (catching) – plná čára
- Nepřerušující – non-interrupting – při spuštění aktivita pokračuje, procesní tok se větví u zachytávající události (catching) – přerušovaná čára



Typy událostí

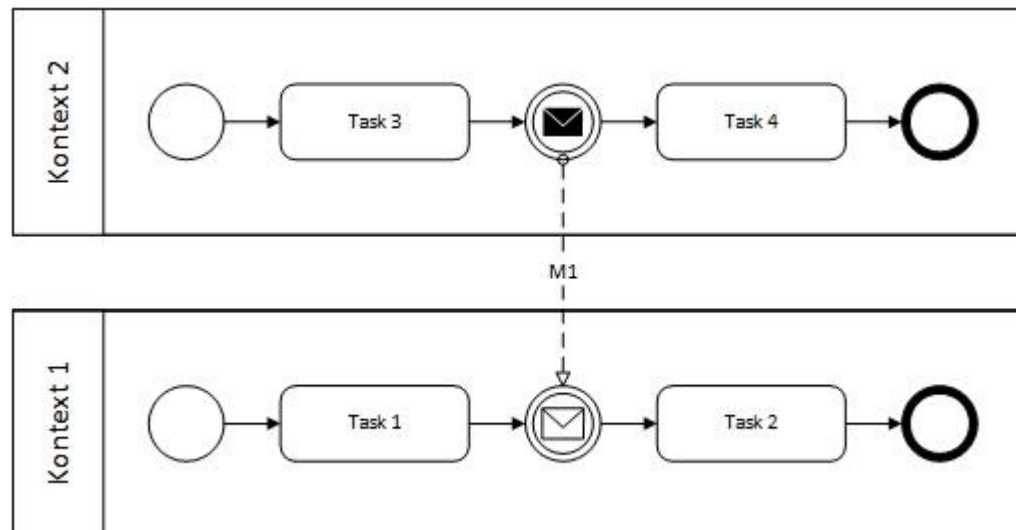
40

- Vstupní
- Vnitřní
- Hraniční (boundary)
- Koncové

Vnitřní události

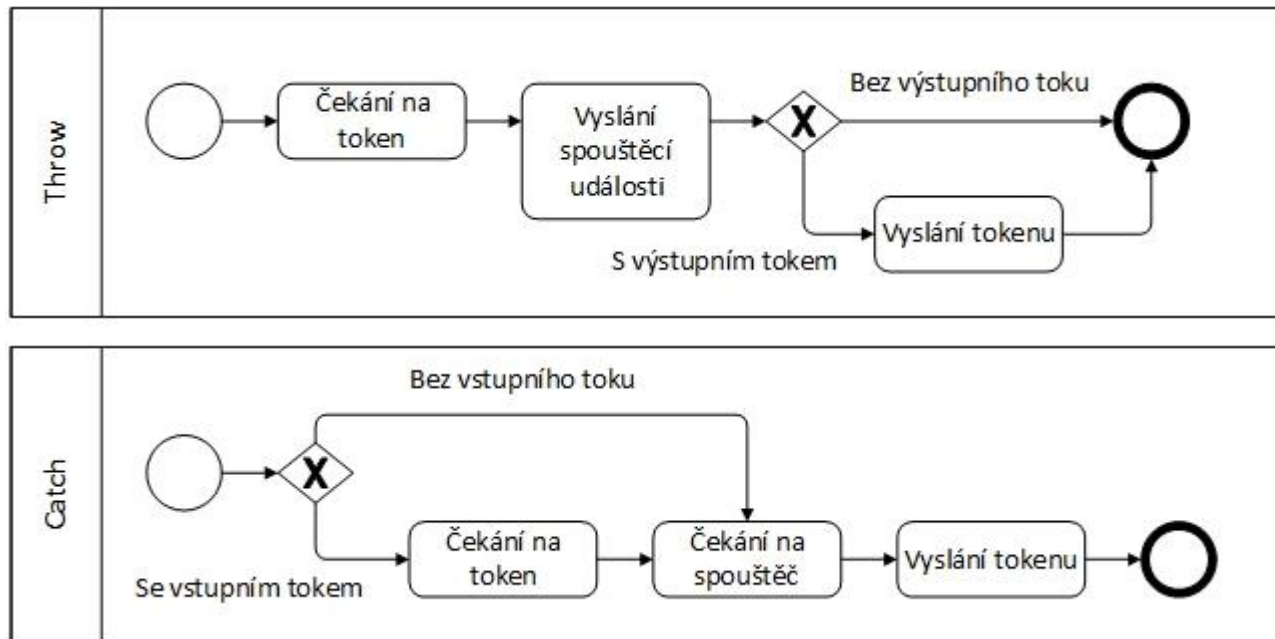
41

- Nevyvolávají novou instanci procesu! Vznikají až v jeho průběhu.
- Mohou chytat i házet (catch/throw)
 - ▣ Catch – čeká na trigger události, pak vysílá token
 - ▣ Throw – vysílá trigger události, pak vysílá token



Vnitřní události - metamodel

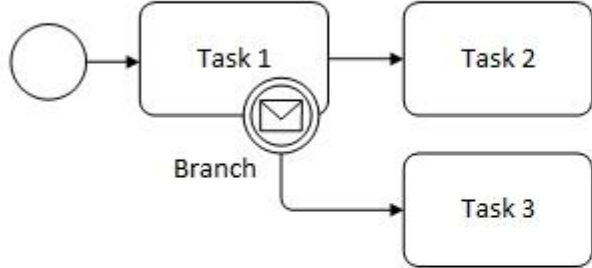
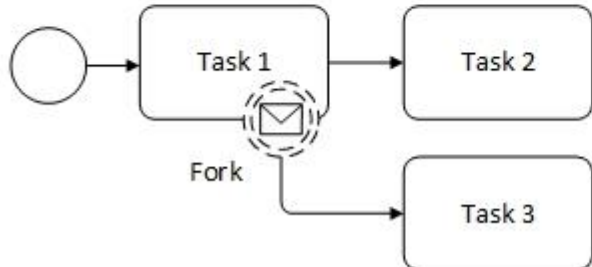
42



Boundary event














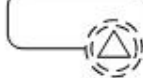


























43

- Interrupting – zahýbají procesní tok (branch)
- Non-interrupting – větví procesní tok (fork)

HRANIČNÍ UDÁLOSTI (BOUNDARY EVENTS)		
Syntax	Chování	Sémantika
	Přerušující (interrupting)	IF Branch triggered before Task 1 finishes terminate Task 1 execute Task 3 Else execute Task 2
	Nepřerušující (non-interrupting)	IF Fork triggered before Task 1 finishes continue executing Task 1 AND for to execute Task 3 concurrently Else execute Task 2

Události

44

Trigger	Start			Intermediate				End
Název	Top	Event sub-process interrupting	Event sub-process non-interrupting	Catching	Boundary interrupting	Boundary non-interrupting	Throwing	
Message								
Signal								
Timer								
Conditional								
Terminate								

Message a Signal









45

- Obojí jsou balíčky nějakých dat
- Message – zasíláno 1:1, obsahuje informace od odesílateli i příjemci
- Signal – zasíláno 1:N, odesílatel ani příjemce nejsou specifikováni

Události Message

46

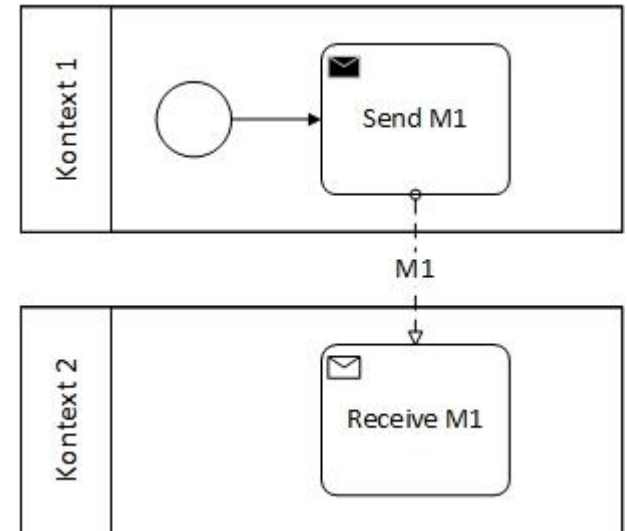
- Catching Message – spouštěny přijetím zprávy z jiného kontextu
- Throwing Message – hází zprávu (výsledek) do jiného kontextu

Trigger	Start			Intermediate				End
Název	Top	Event sub-process interrupting	Event sub-process non-interrupting	Catching	Boundary interrupting	Boundary non-interrupting	Throwing	
Message								
Catching sémantika – čeká se na zprávu z jiného kontextu, pak se vysílá token							Throwing sémantika - při přijetí tokenu odeslání zprávy do jiného kontextu	

Send a Receive tasky

47






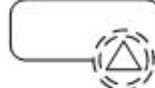


- Lze používat i ve spojení se Send a Receive tasky (speciální druh tasků)
- Fungují zcela analogicky
 - ▣ Send task – po skončení zasílá zprávu přes všechny odchozí message flows
 - ▣ Receive task – čeká na zprávu a po skončení vysílá tokeny na všechny odchozí vlákna



Události Signal

48









- Catching události typu Signal jsou spouštěny přijetím vysílaného signálu ze současného kontextu nebo z externího kontextu

Trigger	Start			Intermediate				End
	Název	Top	Event sub-process interrupting	Event sub-process non-interrupting	Catching	Boundary interrupting	Boundary non-interrupting	
Signal								
Catching sémantika – čeká se na obdržení signálu, pak se vysílá token							Throwing sémantika - čeká se na token, pak se hází signál	

Události Timer

49









- Reaguje na časový okamžik – proto je používán pouze v režimu catch. Spuštění – jakmile je splněna podmínka
- 3 způsoby použití:
 - ▣ Vypršení času
 - ▣ Časové rozvětvení – pro non-interrupting variantu
 - ▣ Oddálení – v rámci toku pozdrží pokračování, dokud není splněna podmínka

Trigger	Start			Intermediate				End
	Název	Top	Event sub-process interrupting	Event sub-process non-interrupting	Catching	Boundary interrupting	Boundary non-interrupting	
Timer								
Catching sémantika – Start - vysílá se token, jakmile je podmínka Timeru TRUE Intermediate - čeká se na token, pak se vysílá token v okamžiku, kdy je podmínka TRUE							Throwing sémantika - nepoužívá se	

Události Conditional

50


- Spouštěny splněním Booleovské podmínky z FALSE na TRUE

Trigger	Start			Intermediate				End
Název	Top	Event sub-process interrupting	Event sub-process non-interrupting	Catching	Boundary interrupting	Boundary non-interrupting	Throwing	End
Conditional								
Catching sémantika – čeká se na změnu podmínky FALSE -> TRUE, pak se vysílá token							Throwing sémantika - nepoužívá se	

Události Terminate

51

- Událost zkonsumuje token a ukončí všechny aktivity v procesu (všechny vlákna)

Trigger	Start			Intermediate				End
Název	Top	Event sub-process interrupting	Event sub-process non-interrupting	Catching	Boundary interrupting	Boundary non-interrupting	Throwing	End
Terminate								
Catching sémantika – nepoužívá se							Throwing sémantika - po přijetí tokenu je zkonsumován a všechny aktivity v procesu ukončeny	




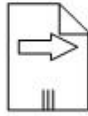




Datové objekty

52

- Propojovány asociační vazbou, ne sekvenčním tokem!
- Neovlivňují přímo sekvenční tok
- Mohou být v 0 či více stavech (UML 2 specifikace: *„A condition or situation during the life of an object during which it satisfies some condition, performs some activity or waits for some event.“*)

Typy datových objektů

53

TYPY DATOVÝCH OBJEKTŮ (DATA OBJECT TYPES)		
Syntax	Syntax kolekce	Sémantika
 Data Object	 Data Object Collection	Znázorňuje datový tok v rámci procesu. Nejčastěji užívaná forma.
 Data Object	 Data Object Collection	Ukazuje datové vstupy do top-level procesu, globalního procesu nebo tasku volaného call aktivitou. Atribut optional specifikuje, zda se jedná o volitelný či mandatorní vstup.
 Data Object	 Data Object Collection	Ukazuje datové výstupy z top-level procesu.
 Data Store		Odkaz na externí datové úložiště jako např. relační databázi, která může uchovávat a poskytovat data. Má atribut kapacita , který je normálně nastaven na neomezeno .

DĚKUJI ZA POZORNOST

