

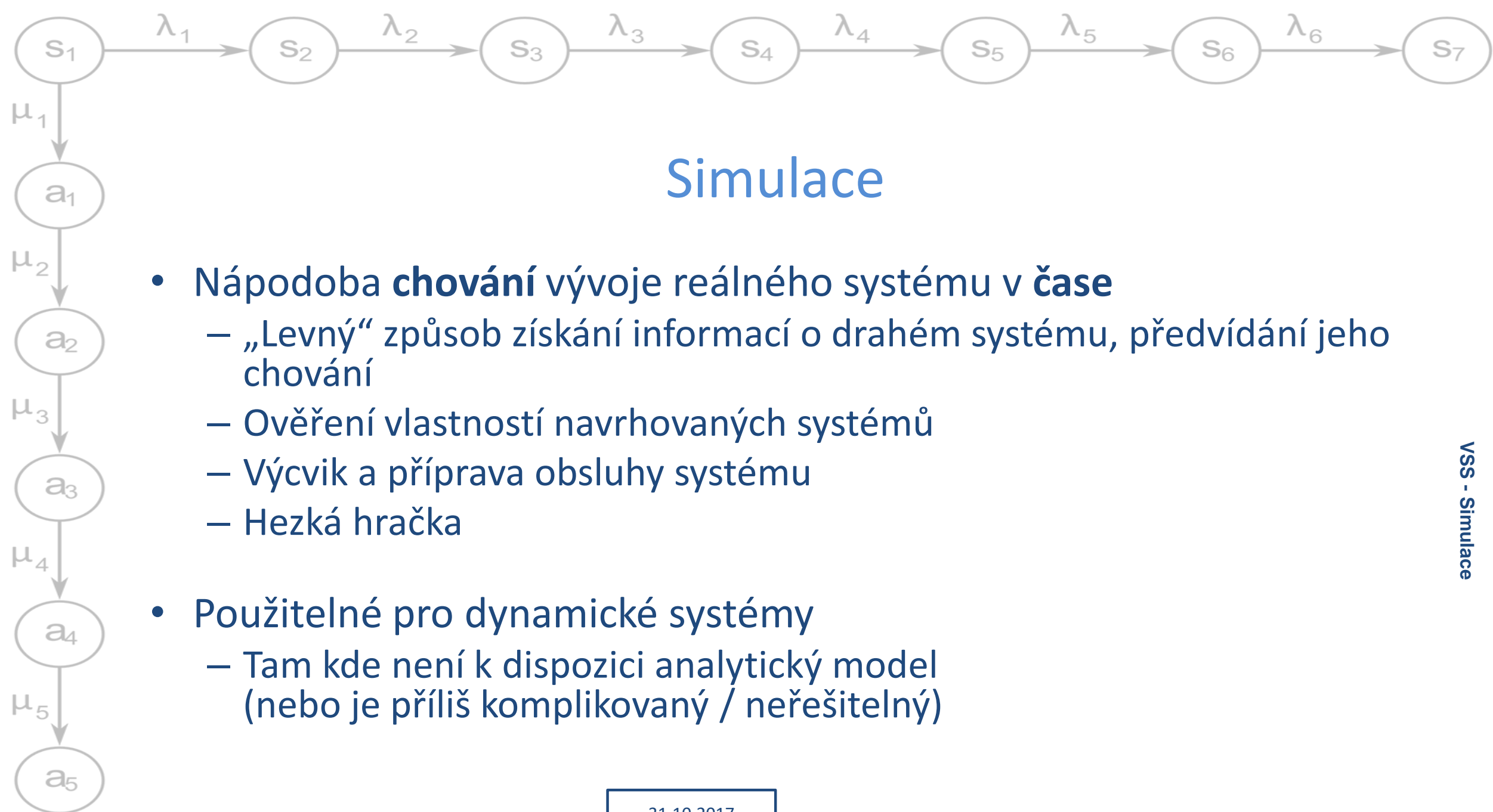
Simulace



Základy simulací

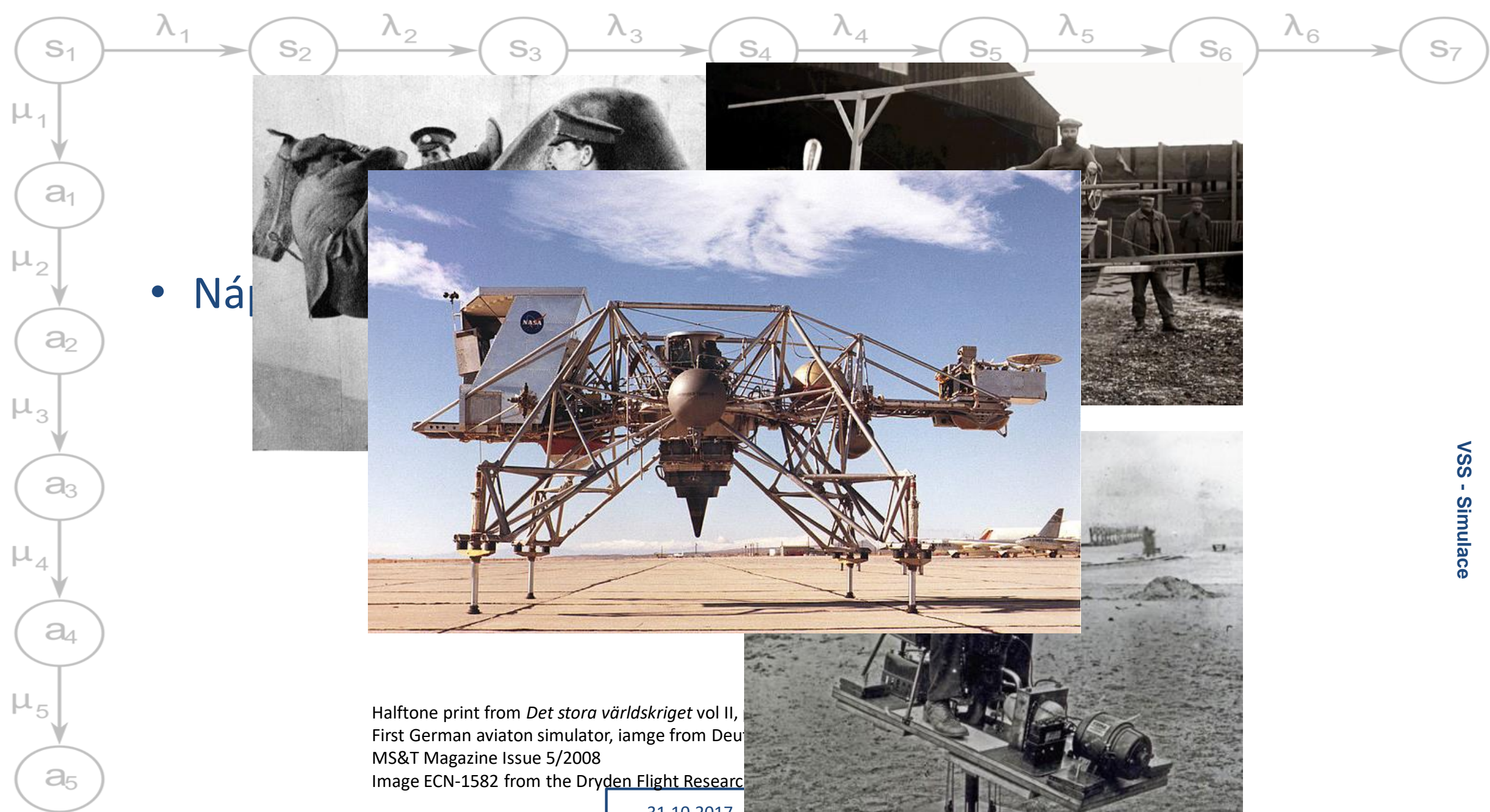
Richard Lipka

31.10. 2017

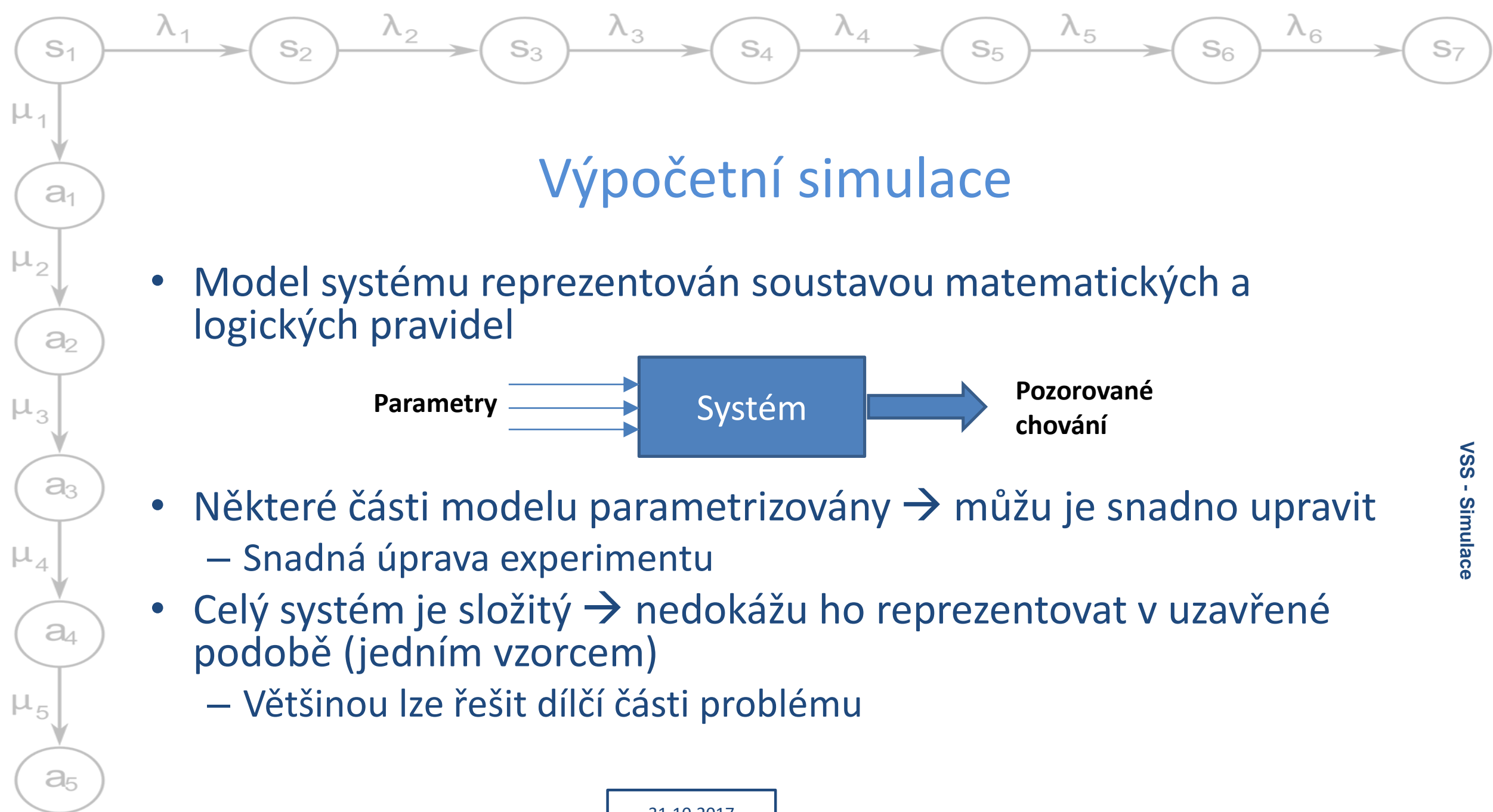


Simulace

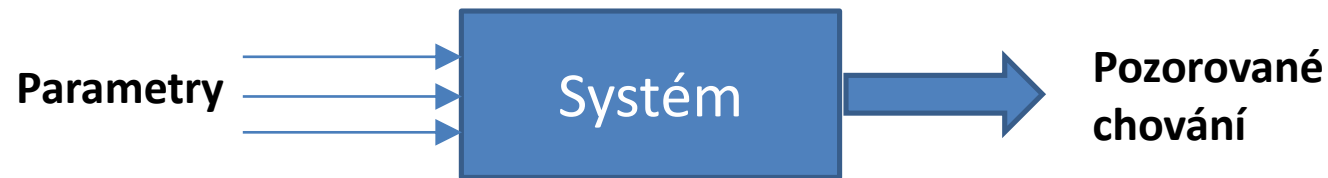
- Nápodoba **chování** vývoje reálného systému v **čase**
 - „Levný“ způsob získání informací o drahém systému, předvídání jeho chování
 - Ověření vlastností navrhovaných systémů
 - Výcvik a příprava obsluhy systému
 - Hezká hračka
- Použitelné pro dynamické systémy
 - Tam kde není k dispozici analytický model (nebo je příliš komplikovaný / neřešitelný)



31.10.2017



- Model systému reprezentován soustavou matematických a logických pravidel



- Některé části modelu parametrizovány \rightarrow můžu je snadno upravit
 - Snadná úprava experimentu
- Celý systém je složitý \rightarrow nedokážu ho reprezentovat v uzavřené podobě (jedním vzorcem)
 - Většinou lze řešit dílčí části problému



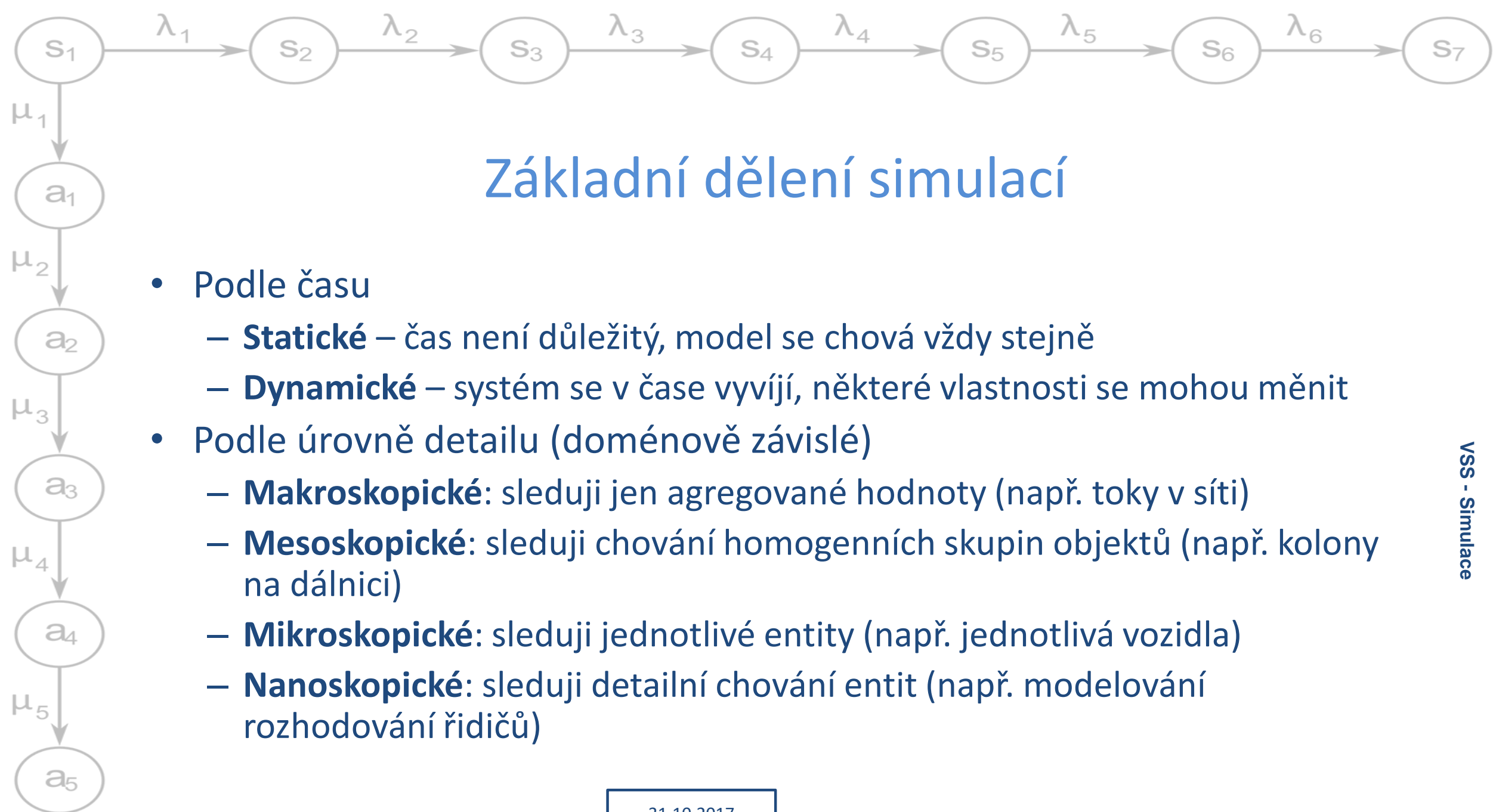
laci

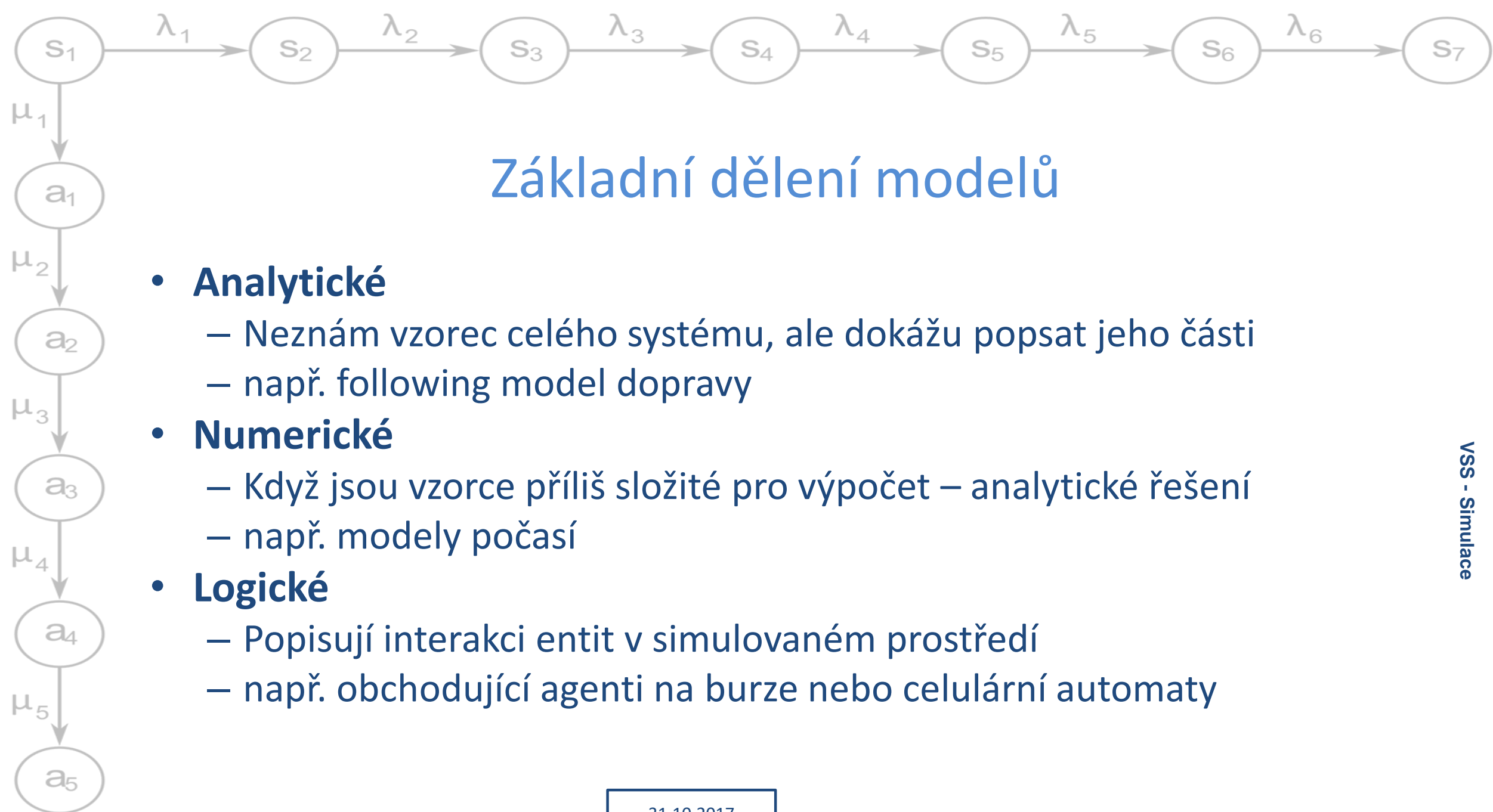
- Podle chov
 - **Deterministický** chování je
 - **Stochastický** (pseudo)
- Podle reprezentace času
 - **Spojitý**: stav systému
 - **Diskrétní**: stav systému definován jen v některých bodech v case



VSS - Simulace

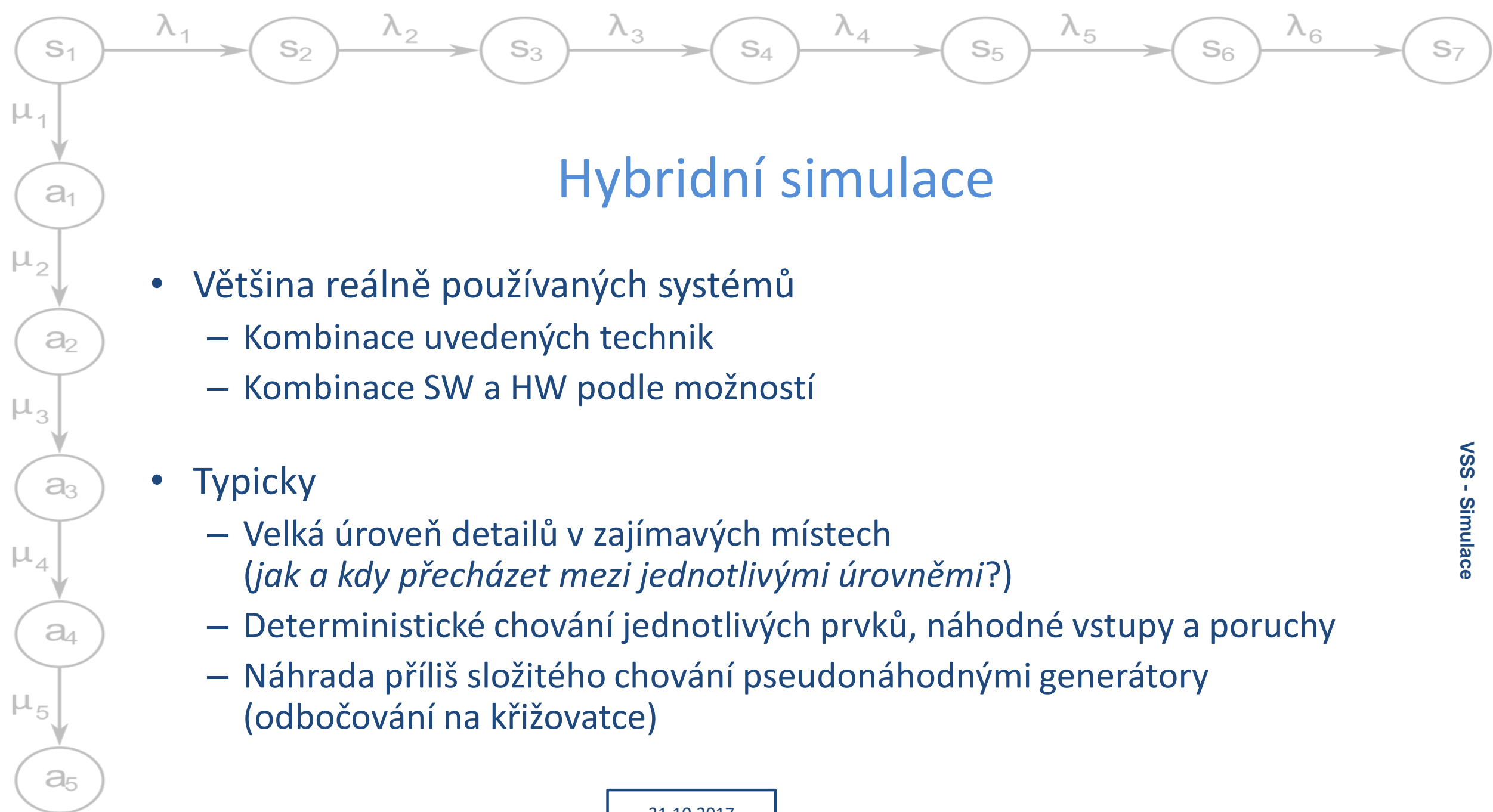
Source: <http://www.cgtronic.com/inspiration/next-please-by-zhipeng-song/>





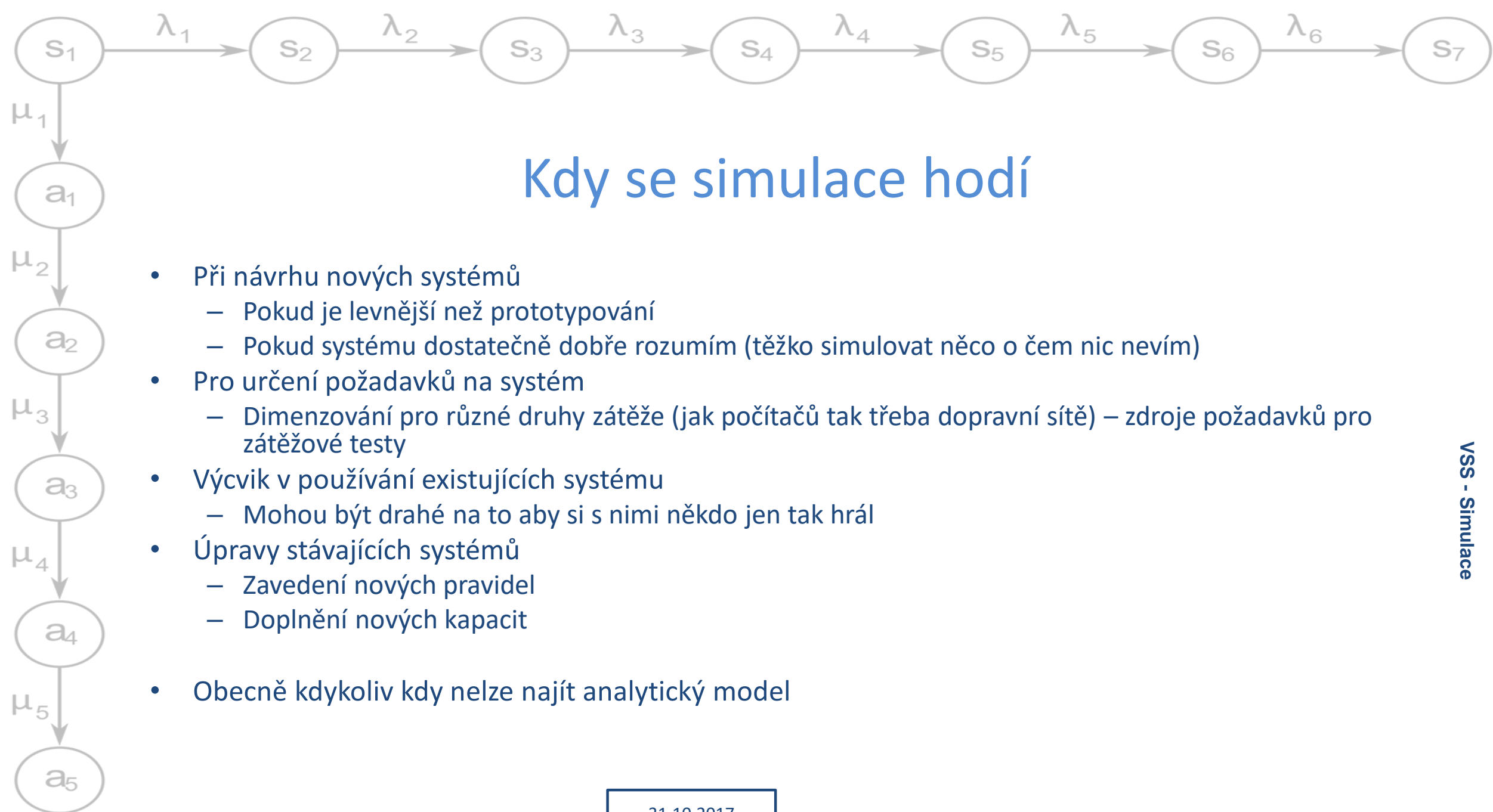
Základní dělení modelů

- **Analytické**
 - Neznám vzorec celého systému, ale dokážu popsat jeho části
 - např. following model dopravy
- **Numerické**
 - Když jsou vzorce příliš složité pro výpočet – analytické řešení
 - např. modely počasí
- **Logické**
 - Popisují interakci entit v simulovaném prostředí
 - např. obchodující agenti na burze nebo celulární automaty



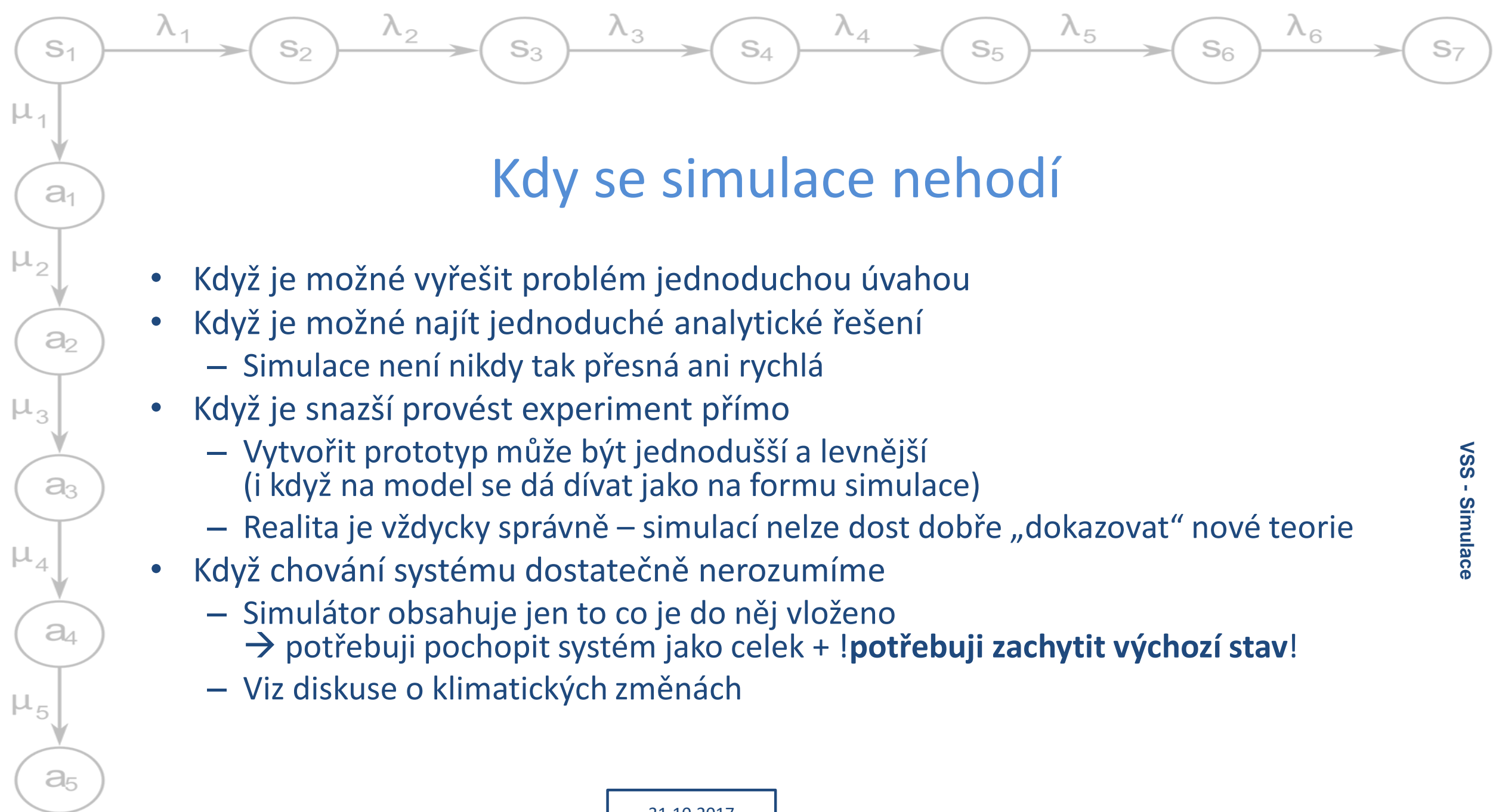
Hybridní simulace

- Většina reálně používaných systémů
 - Kombinace uvedených technik
 - Kombinace SW a HW podle možností
- Typicky
 - Velká úroveň detailů v zajímavých místech
(*jak a kdy přecházet mezi jednotlivými úrovněmi?*)
 - Deterministické chování jednotlivých prvků, náhodné vstupy a poruchy
 - Náhrada příliš složitého chování pseudonáhodnými generátory
(odbočování na křižovatce)



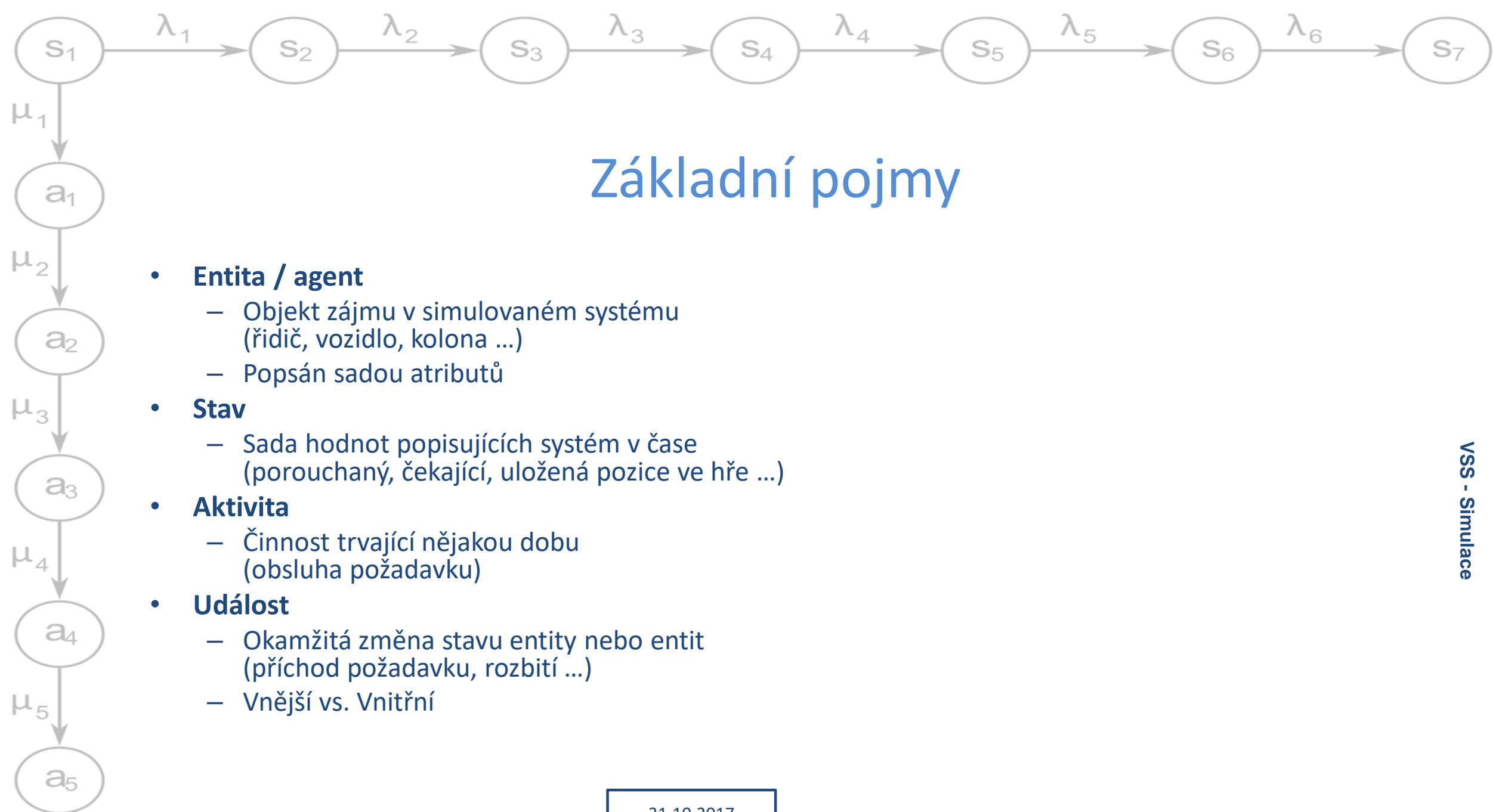
Kdy se simulace hodí

- Při návrhu nových systémů
 - Pokud je levnější než prototypování
 - Pokud systému dostatečně dobře rozumím (těžko simulovat něco o čem nic nevím)
- Pro určení požadavků na systém
 - Dimenzování pro různé druhy zátěže (jak počítačů tak třeba dopravní sítě) – zdroje požadavků pro zátěžové testy
- Výcvik v používání existujících systému
 - Mohou být drahé na to aby si s nimi někdo jen tak hrál
- Úpravy stávajících systémů
 - Zavedení nových pravidel
 - Doplnění nových kapacit
- Obecně kdykoliv kdy nelze najít analytický model



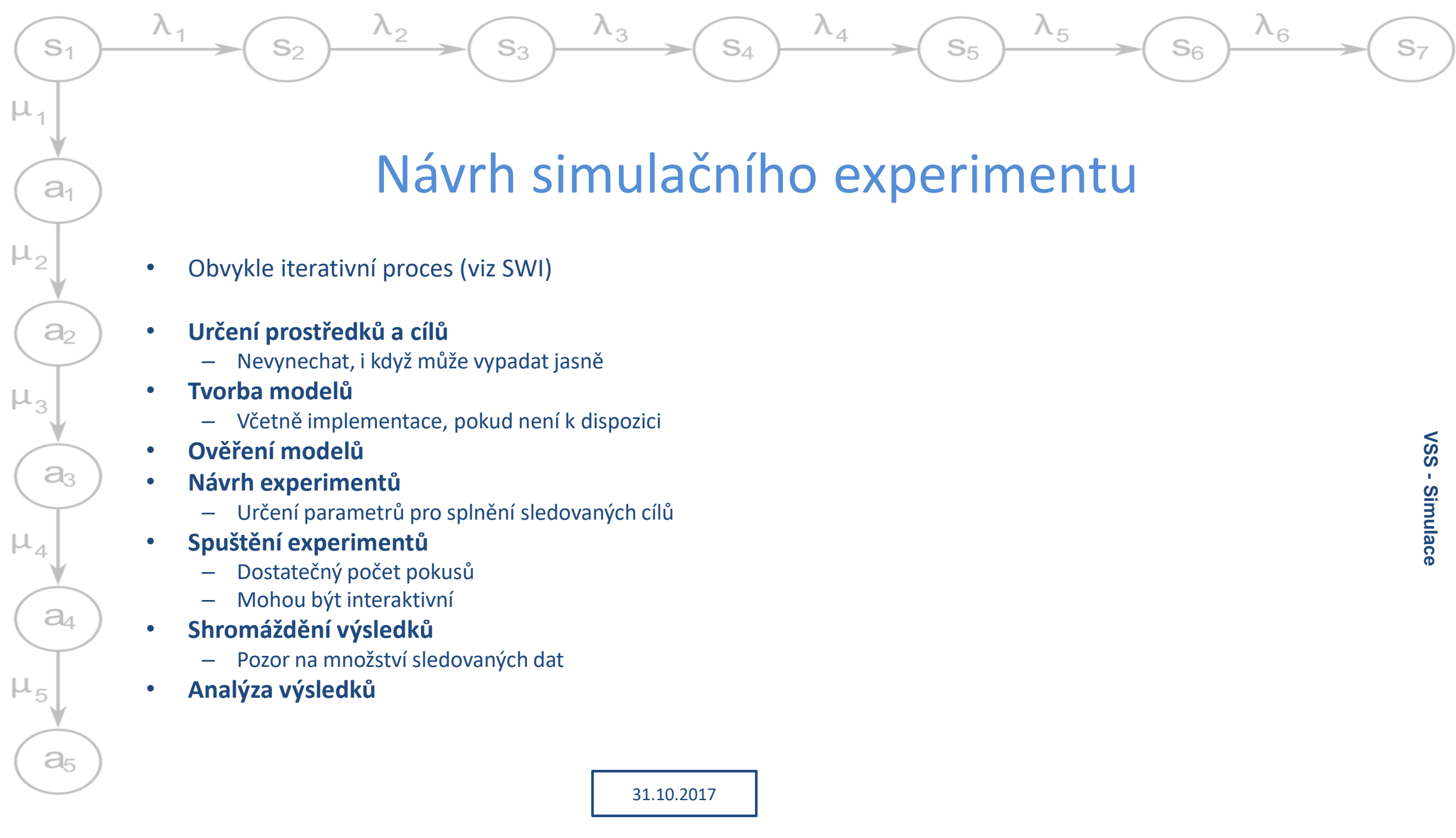
Kdy se simulace nehodí

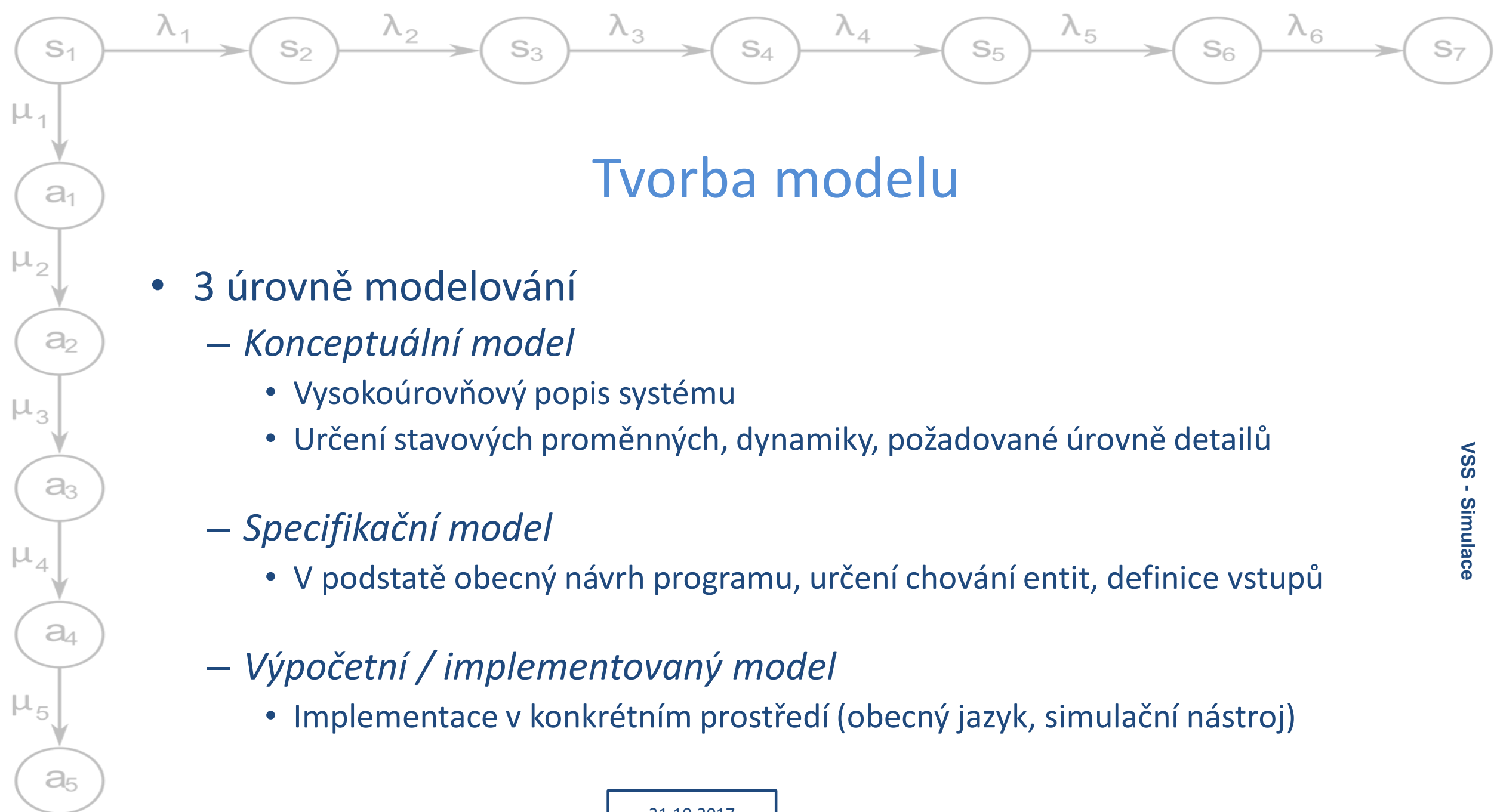
- Když je možné vyřešit problém jednoduchou úvahou
- Když je možné najít jednoduché analytické řešení
 - Simulace není nikdy tak přesná ani rychlá
- Když je snazší provést experiment přímo
 - Vytvořit prototyp může být jednodušší a levnější (i když na model se dá dívat jako na formu simulace)
 - Realita je vždycky správně – simulací nelze dost dobře „dokazovat“ nové teorie
- Když chování systému dostatečně nerozumíme
 - Simulátor obsahuje jen to co je do něj vloženo
→ potřebuji pochopit systém jako celek + **!potřebuji zachytit výchozí stav!**
 - Viz diskuse o klimatických změnách

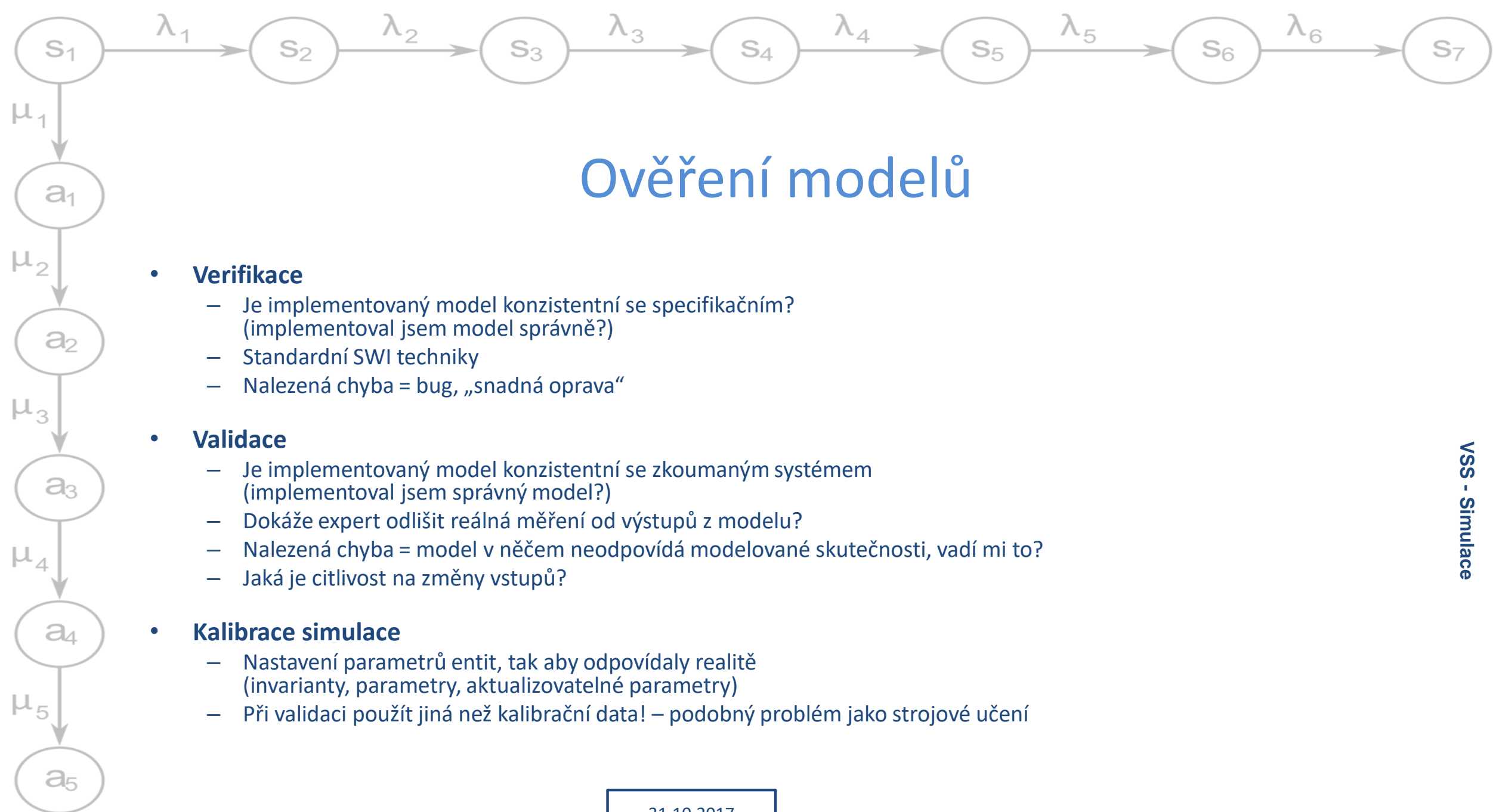


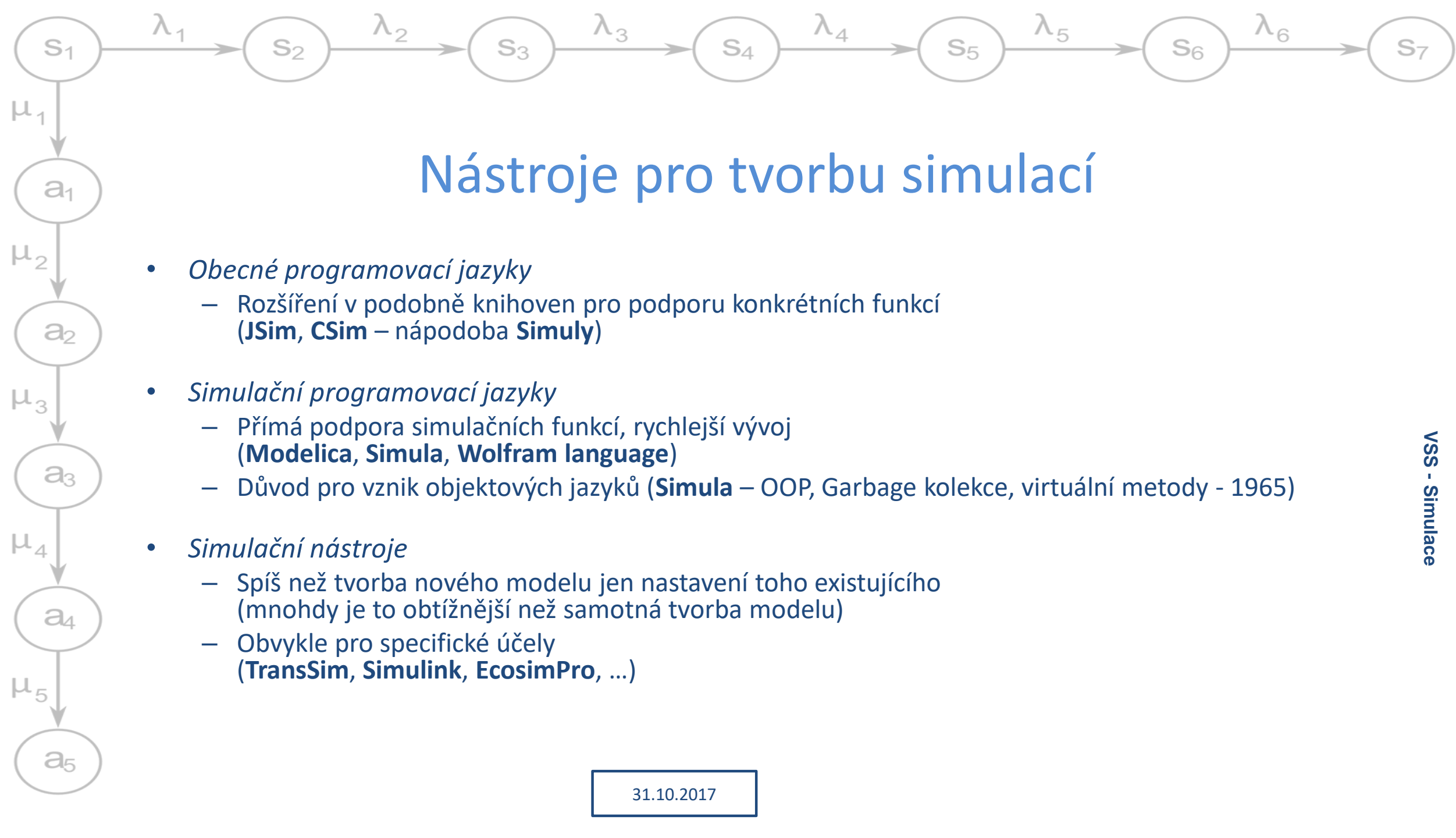
Základní pojmy

- **Entita / agent**
 - Objekt zájmu v simulovaném systému (řidič, vozidlo, kolona ...)
 - Popsán sadou atributů
- **Stav**
 - Sada hodnot popisujících systém v čase (porouchaný, čekající, uložená pozice ve hře ...)
- **Aktivita**
 - Činnost trvající nějakou dobu (obsluha požadavku)
- **Událost**
 - Okamžitá změna stavu entity nebo entit (příchod požadavku, rozbití ...)
 - Vnější vs. Vnitřní



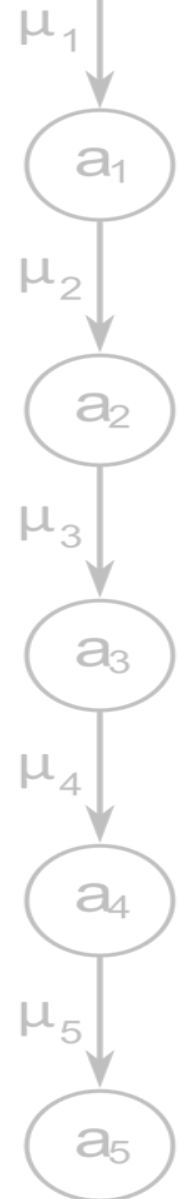






Nástroje pro tvorbu simulací

- *Obecné programovací jazyky*
 - Rozšíření v podobě knihoven pro podporu konkrétních funkcí (**JSim**, **CSim** – nápodoba **Simuly**)
- *Simulační programovací jazyky*
 - Přímá podpora simulačních funkcí, rychlejší vývoj (**Modelica**, **Simula**, **Wolfram language**)
 - Důvod pro vznik objektových jazyků (**Simula** – OOP, Garbage kolekce, virtuální metody - 1965)
- *Simulační nástroje*
 - Spíš než tvorba nového modelu jen nastavení toho existujícího (mnohdy je to obtížnější než samotná tvorba modelu)
 - Obvykle pro specifické účely (**TransSim**, **Simulink**, **EcosimPro**, ...)



Obecné simulační jazyky

- Diskrétní událostní simulace
 - Simula (základ OOP), SimPy
- Spojité simulace
 - VisSim (grafický návrh, generování C)
- Hybridní
 - Simulink (integrovaný s Matlabem, dataflow ...)
 - Modelica (open nástroj i placená, fyzikální systémy)
 - SciLab (numerické simulace, dynamika kapalin ...)
 - NetLogo (založené na Logu, agentní modely a interakce s prostředím)



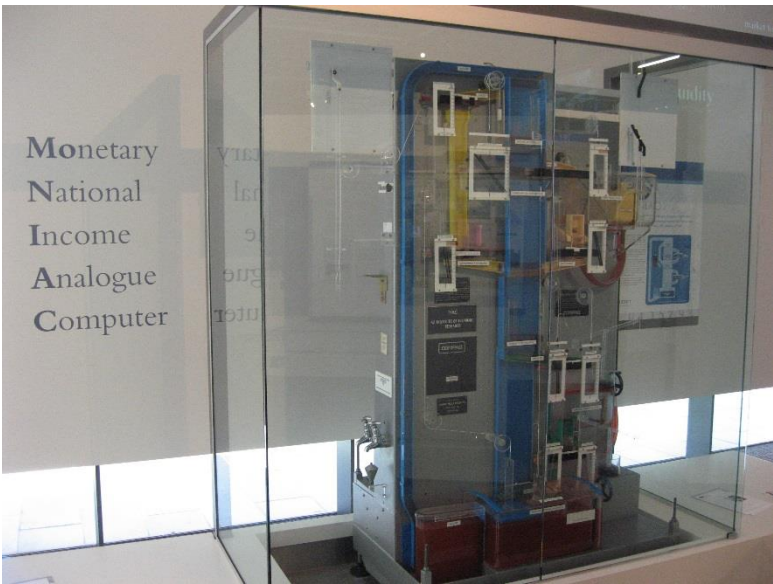
Spojité simulace

- Obvykle založené na soustavách diferenciálních rovnic
 - Balistické křivky, let rakety, proudění kapaliny ...
→ fyzikální modely spojitých dějů
 - Obvykle spojitý čas i prostor
- V každém okamžiku lze určit stav systému
- Původně řešeny na analogových počítačích
- V současné době numerické výpočty (→ výpočet převeden na diskrétní s dostatečně krátkým krokem)
- Obvykle deterministické modely



Spojité simulace - příklad

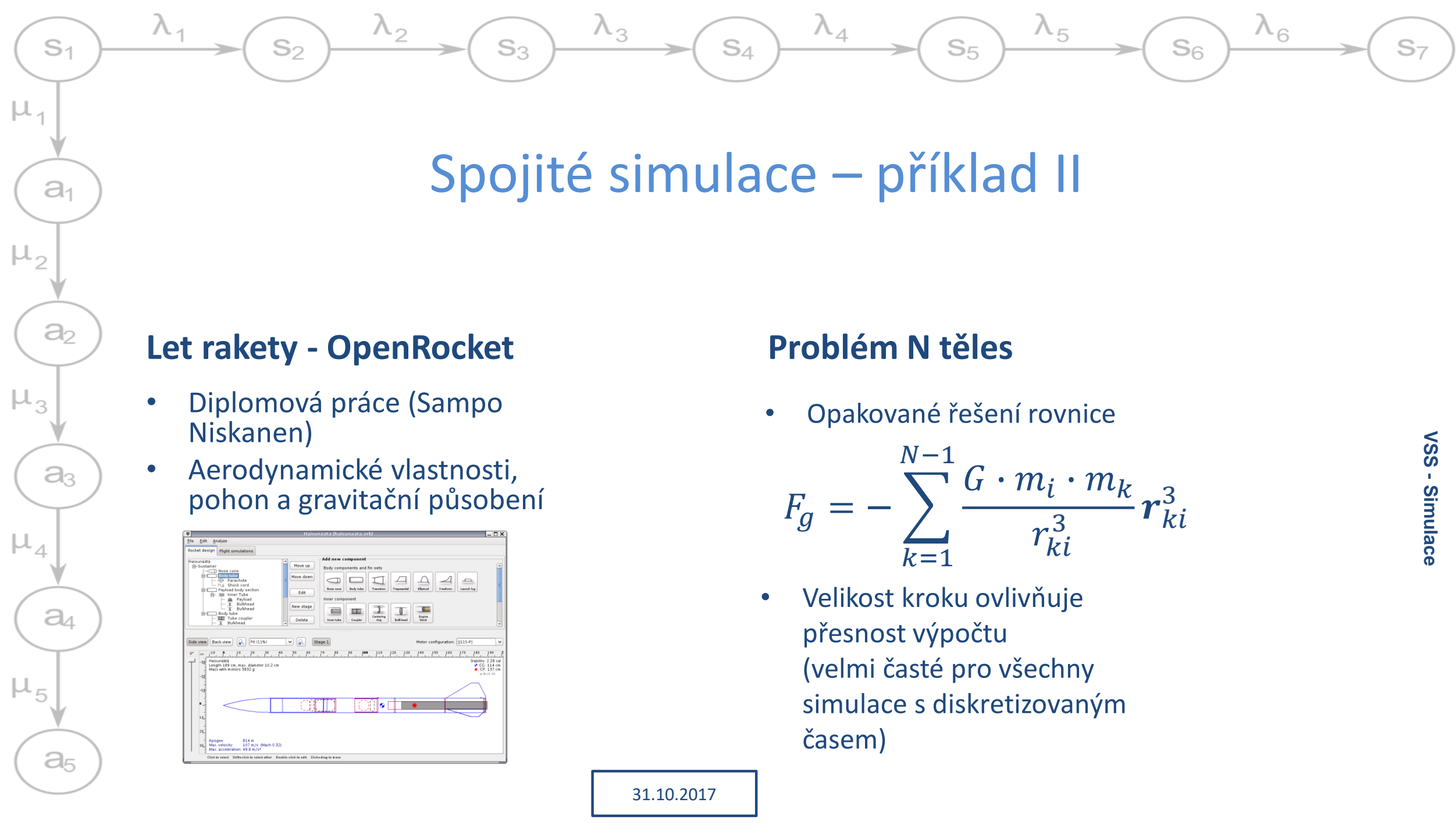
Moniac - modelování ekonomických toků proudem kapaliny (1949)



Globus IMP – mechanický výpočet polohy lodi Vostok I (1961 - 2002)



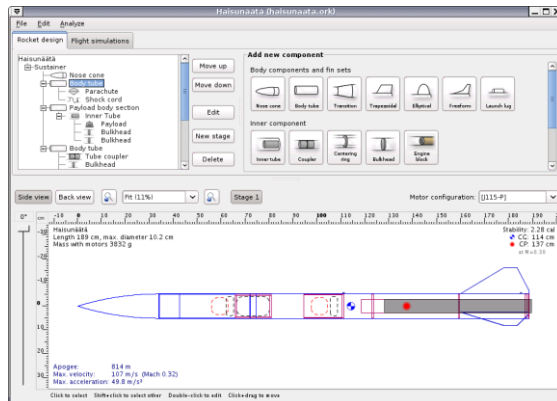
VSS - Simulace



Spojité simulace – příklad II

Let rakety - OpenRocket

- Diplomová práce (Sampo Niskanen)
- Aerodynamické vlastnosti, pohon a gravitační působení

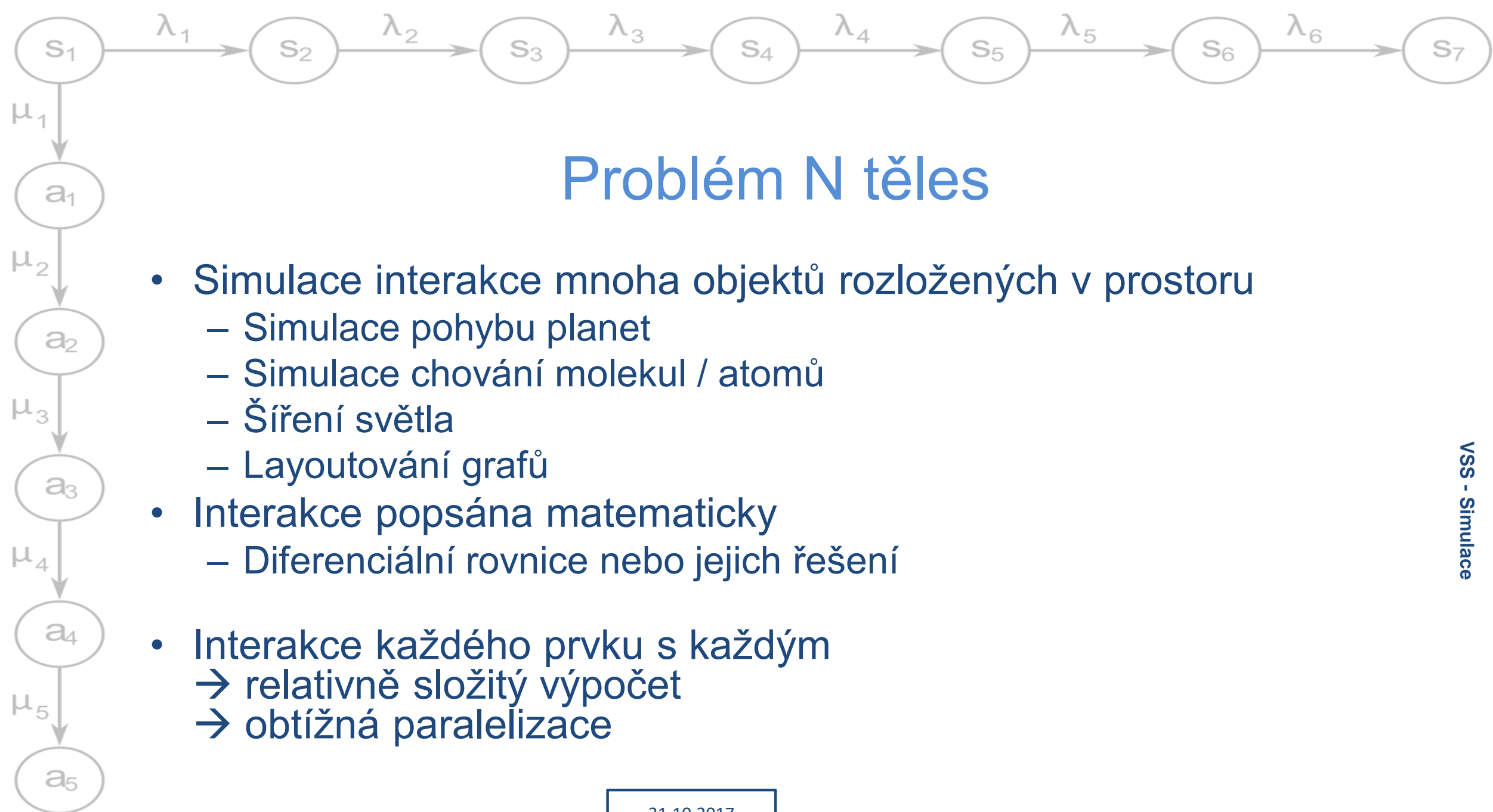


Problém N těles

- Opakované řešení rovnice

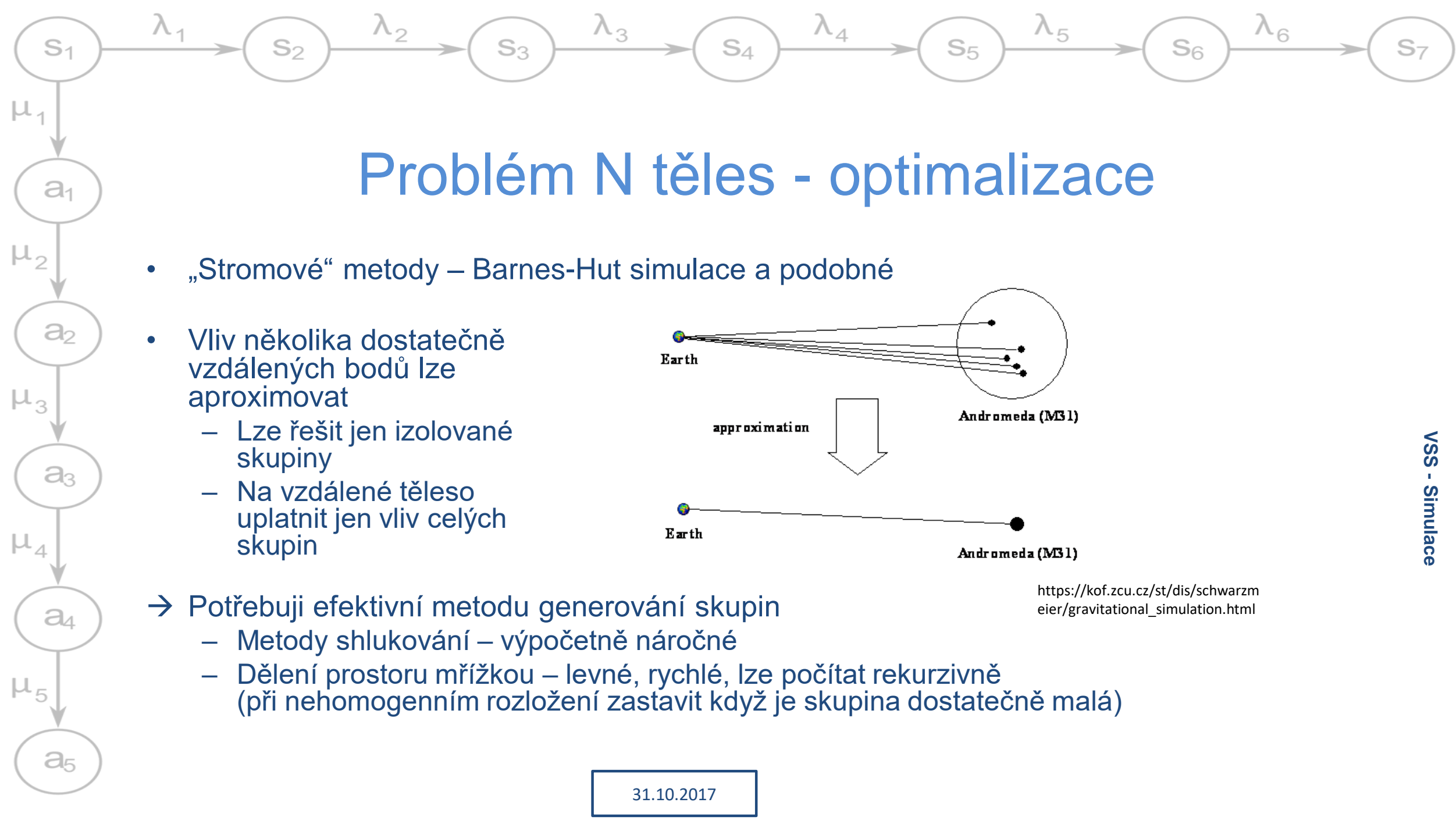
$$F_g = - \sum_{k=1}^{N-1} \frac{G \cdot m_i \cdot m_k}{r_{ki}^3} \mathbf{r}_{ki}^3$$

- Velikost kroku ovlivňuje přesnost výpočtu (velmi časté pro všechny simulace s diskretizovaným časem)



Problém N těles

- Simulace interakce mnoha objektů rozložených v prostoru
 - Simulace pohybu planet
 - Simulace chování molekul / atomů
 - Šíření světla
 - Layoutování grafů
- Interakce popsána matematicky
 - Diferenciální rovnice nebo jejich řešení
- Interakce každého prvku s každým
 - relativně složitý výpočet
 - obtížná paralelizace

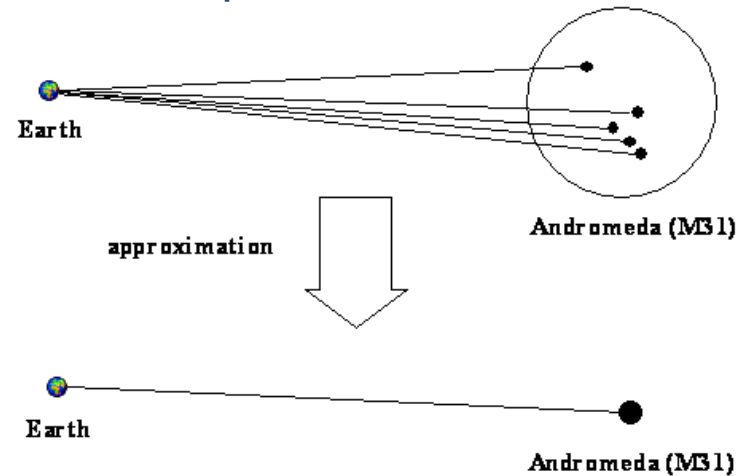


Problém N těles - optimalizace

- „Stromové“ metody – Barnes-Hut simulace a podobné

- Vliv několika dostatečně vzdálených bodů lze aproximovat

- Lze řešit jen izolované skupiny
- Na vzdálené těleso uplatnit jen vliv celých skupin

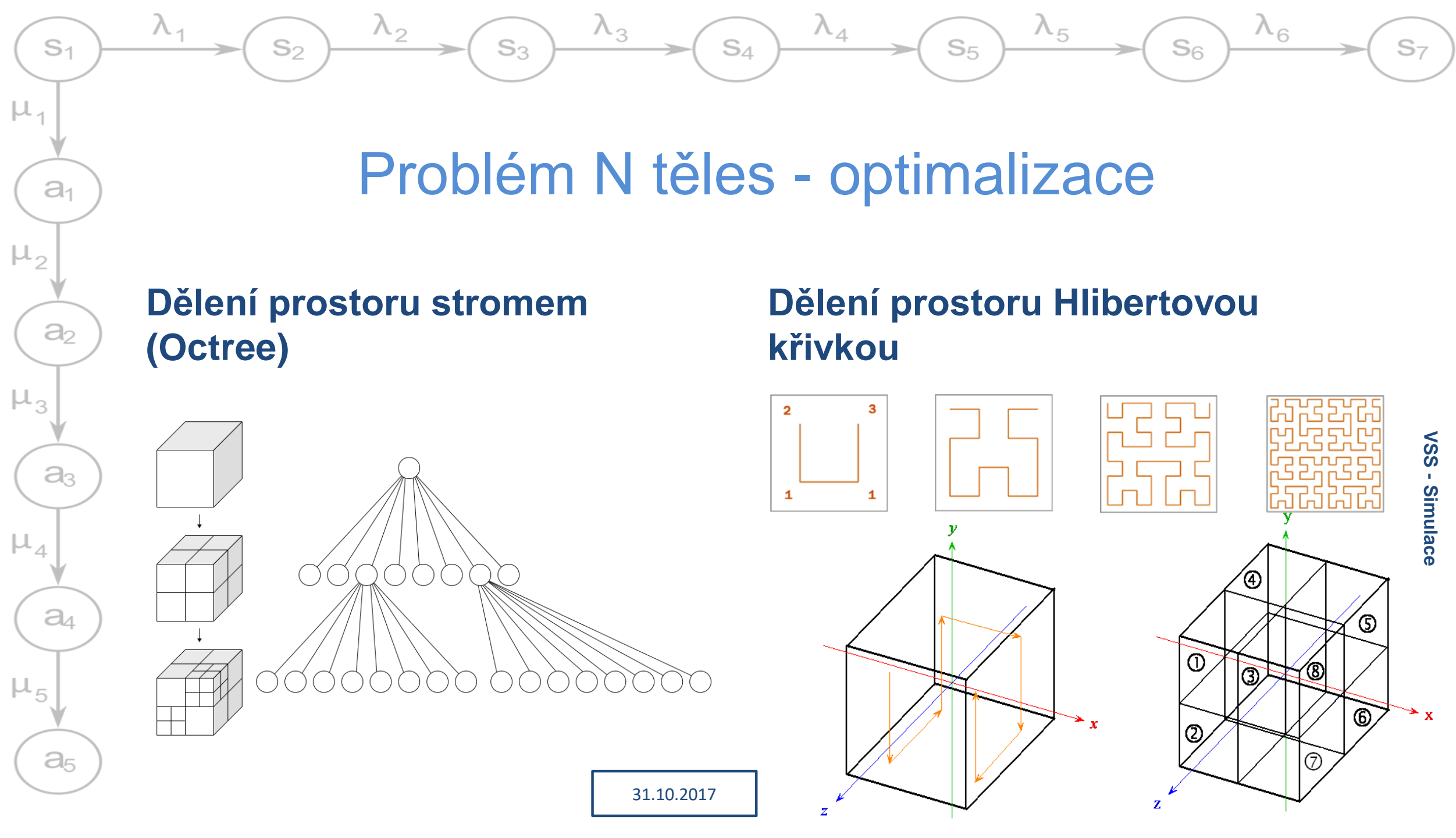


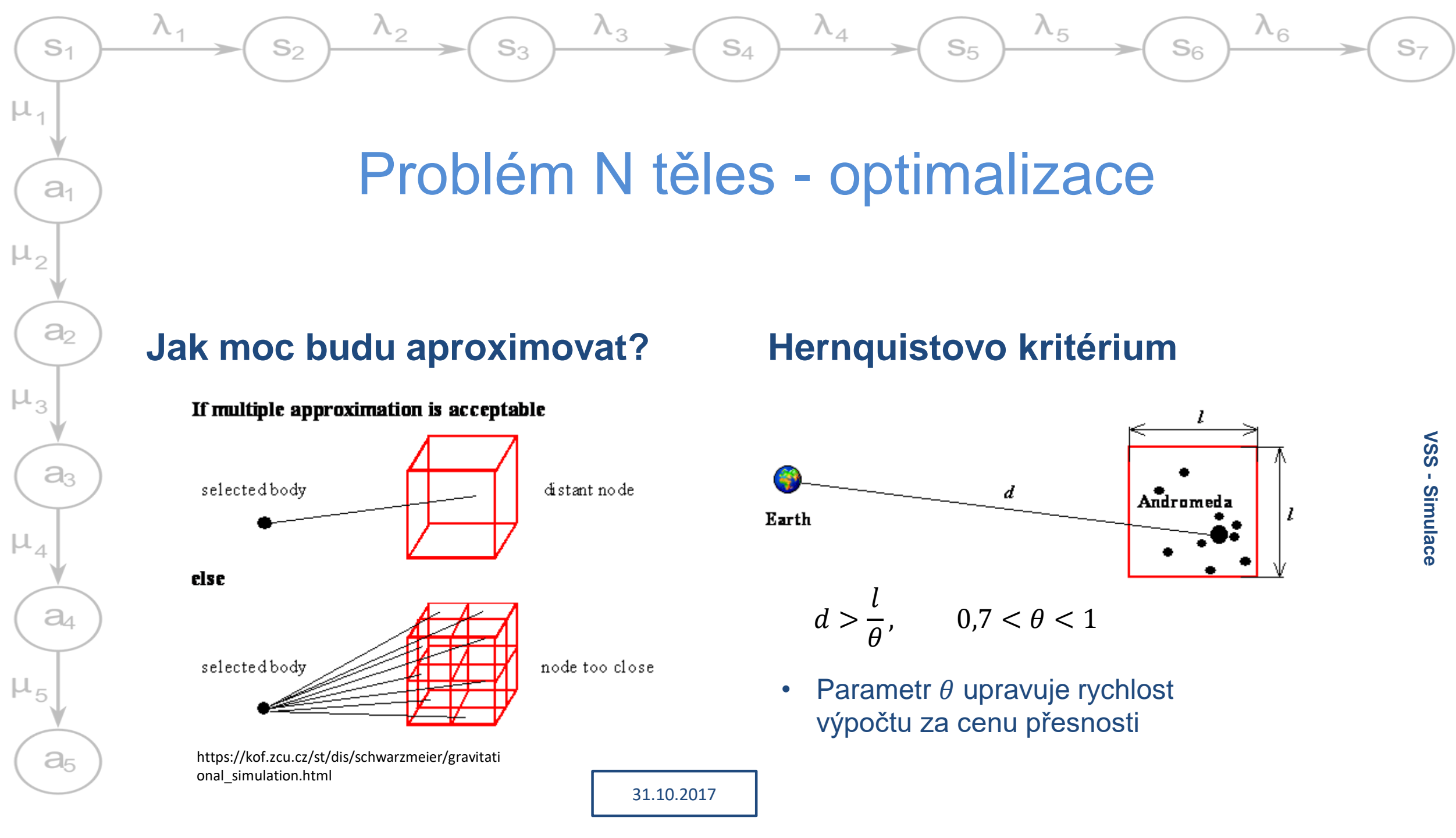
https://kof.zcu.cz/st/dis/schwarzmeier/gravitational_simulation.html

VSS - Simulace

→ Potřebuji efektivní metodu generování skupin

- Metody shlukování – výpočetně náročné
- Dělení prostoru mřížkou – levné, rychlé, lze počítat rekurzivně (při nehomogenním rozložení zastavit když je skupina dostatečně malá)

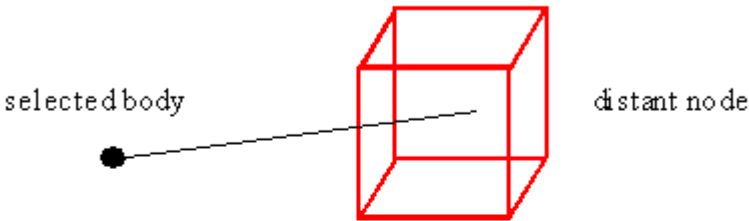




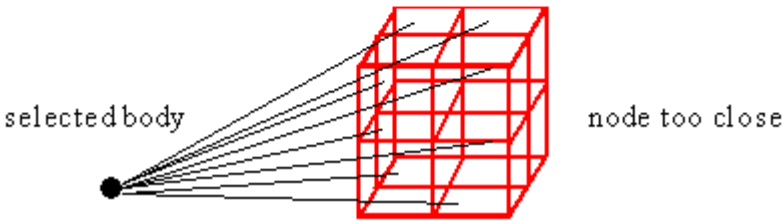
Problém N těles - optimalizace

Jak moc budu aproximovat?

If multiple approximation is acceptable

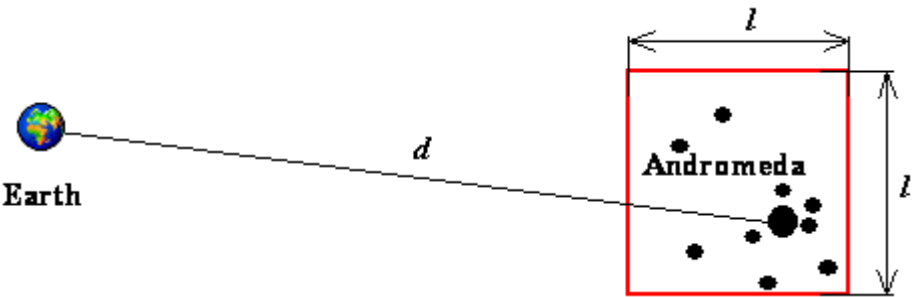


else



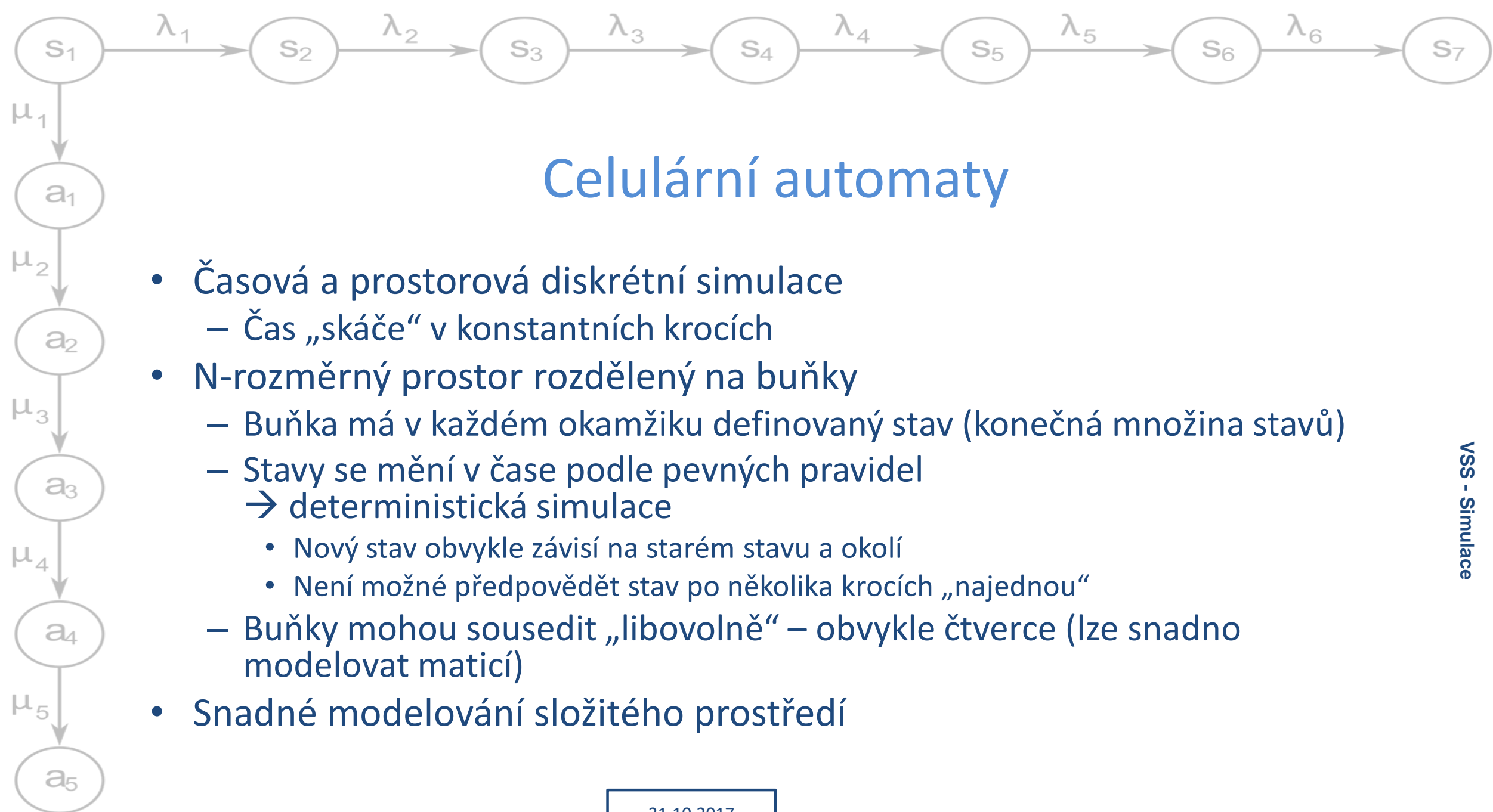
https://kof.zcu.cz/st/dis/schwarzmeier/gravitational_simulation.html

Hernquistovo kritérium



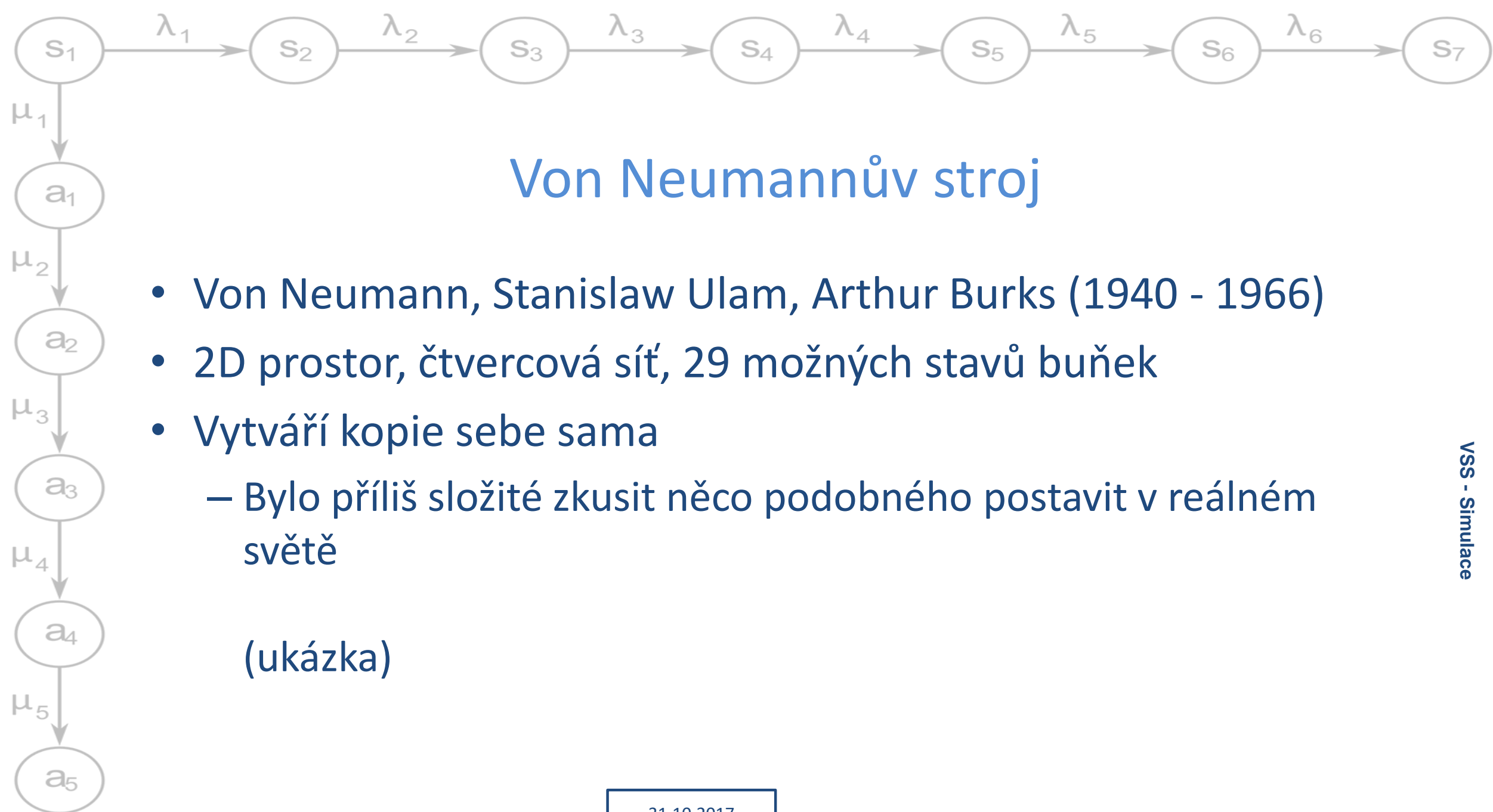
$$d > \frac{l}{\theta}, \quad 0,7 < \theta < 1$$

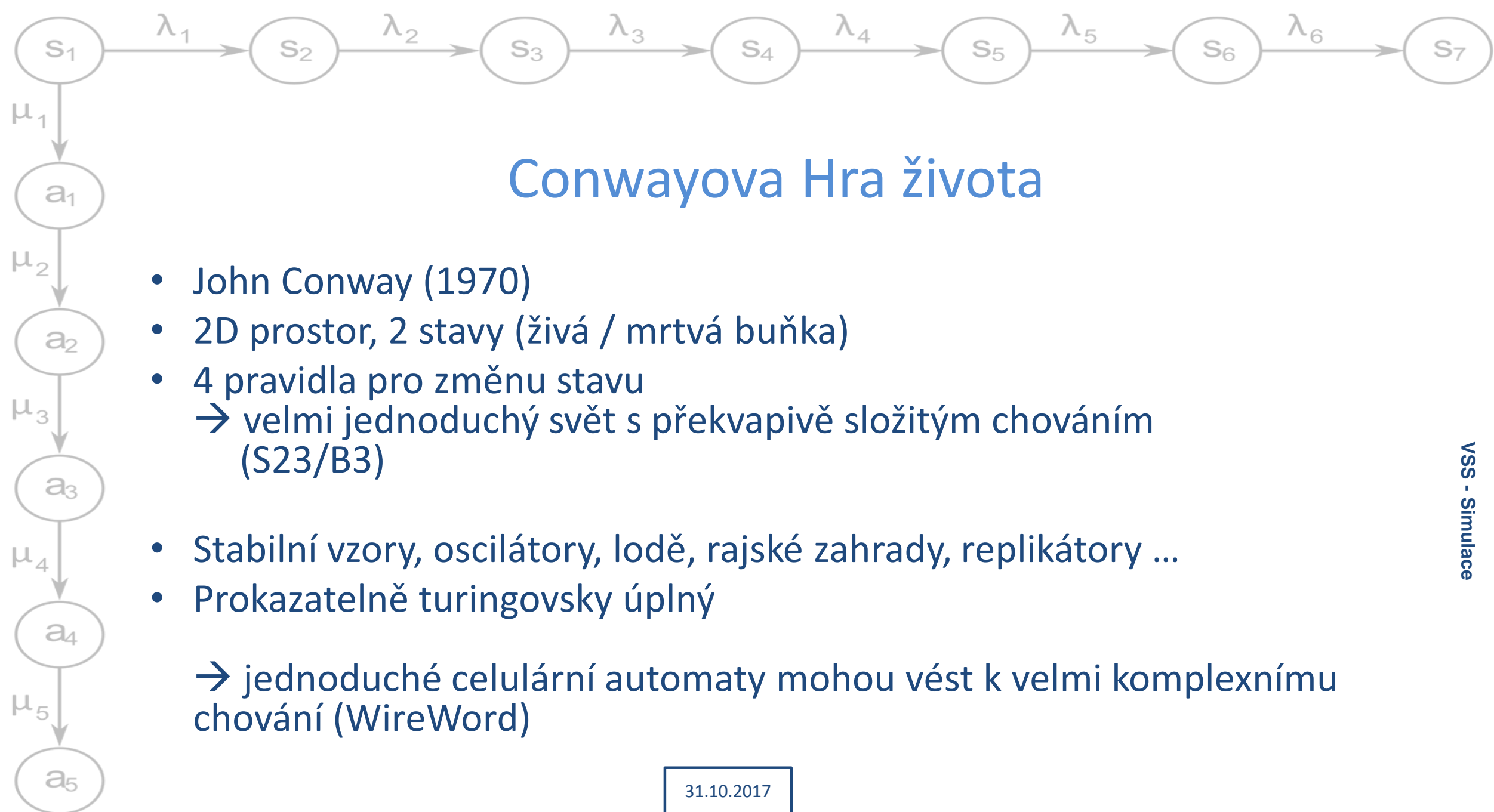
- Parametr θ upravuje rychlost výpočtu za cenu přesnosti

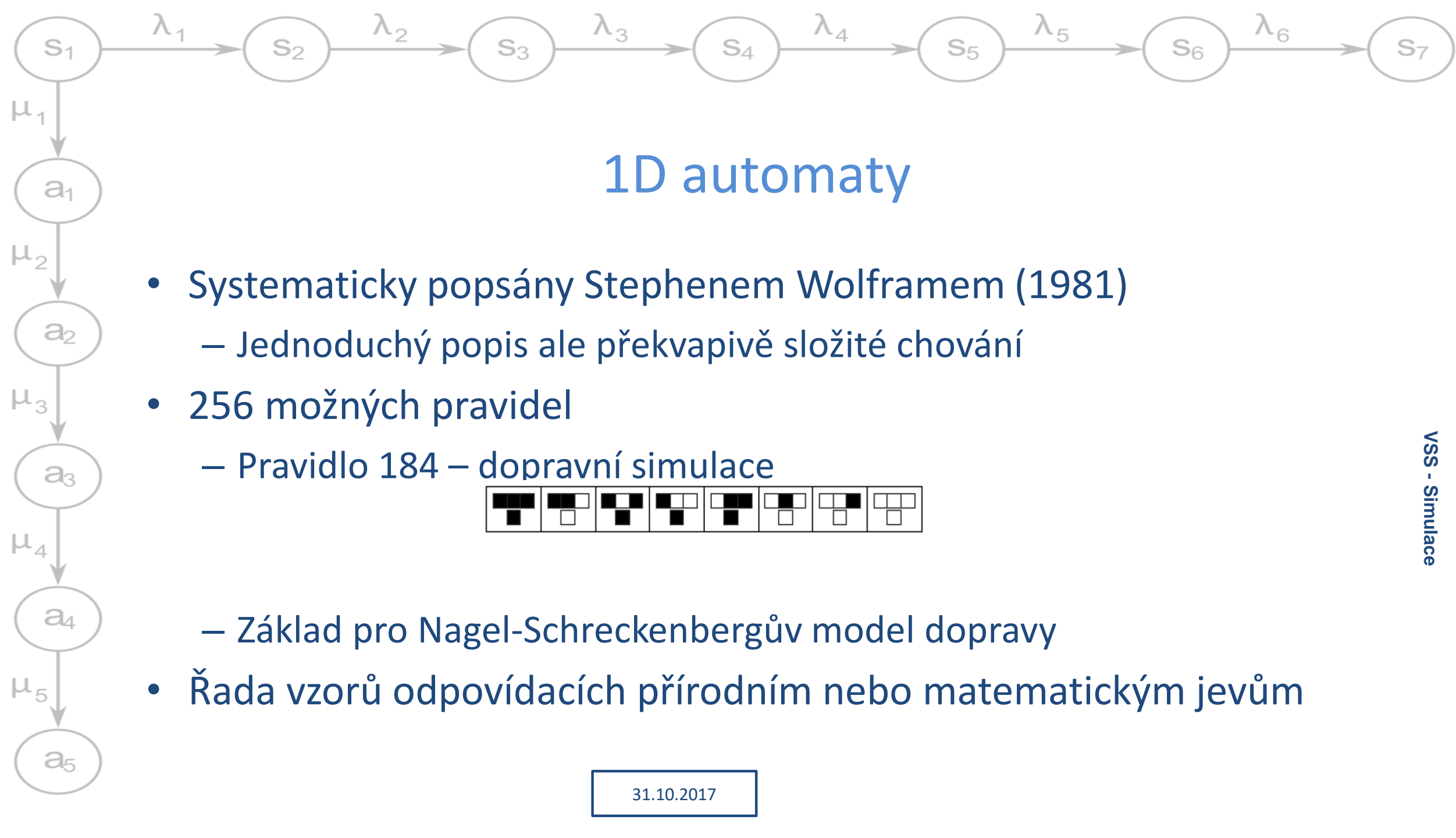


Celulární automaty

- Časová a prostorová diskrétní simulace
 - Čas „skáče“ v konstantních krocích
- N-rozměrný prostor rozdělený na buňky
 - Buňka má v každém okamžiku definovaný stav (konečná množina stavů)
 - Stavy se mění v čase podle pevných pravidel
→ deterministická simulace
 - Nový stav obvykle závisí na starém stavu a okolí
 - Není možné předpovědět stav po několika krocích „najednou“
 - Buňky mohou sousedit „libovolně“ – obvykle čtverce (lze snadno modelovat maticí)
- Snadné modelování složitého prostředí

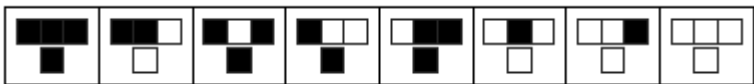


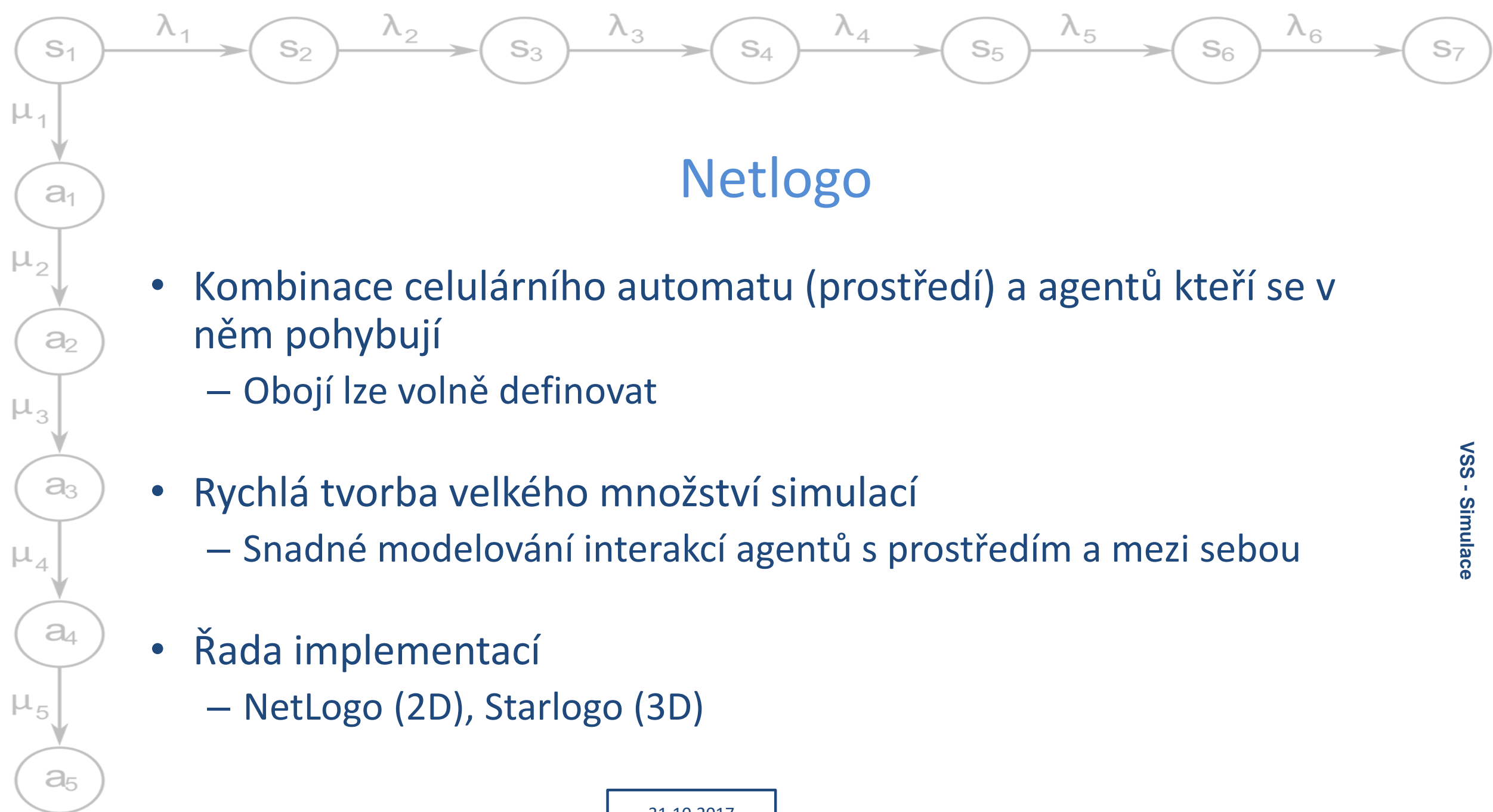


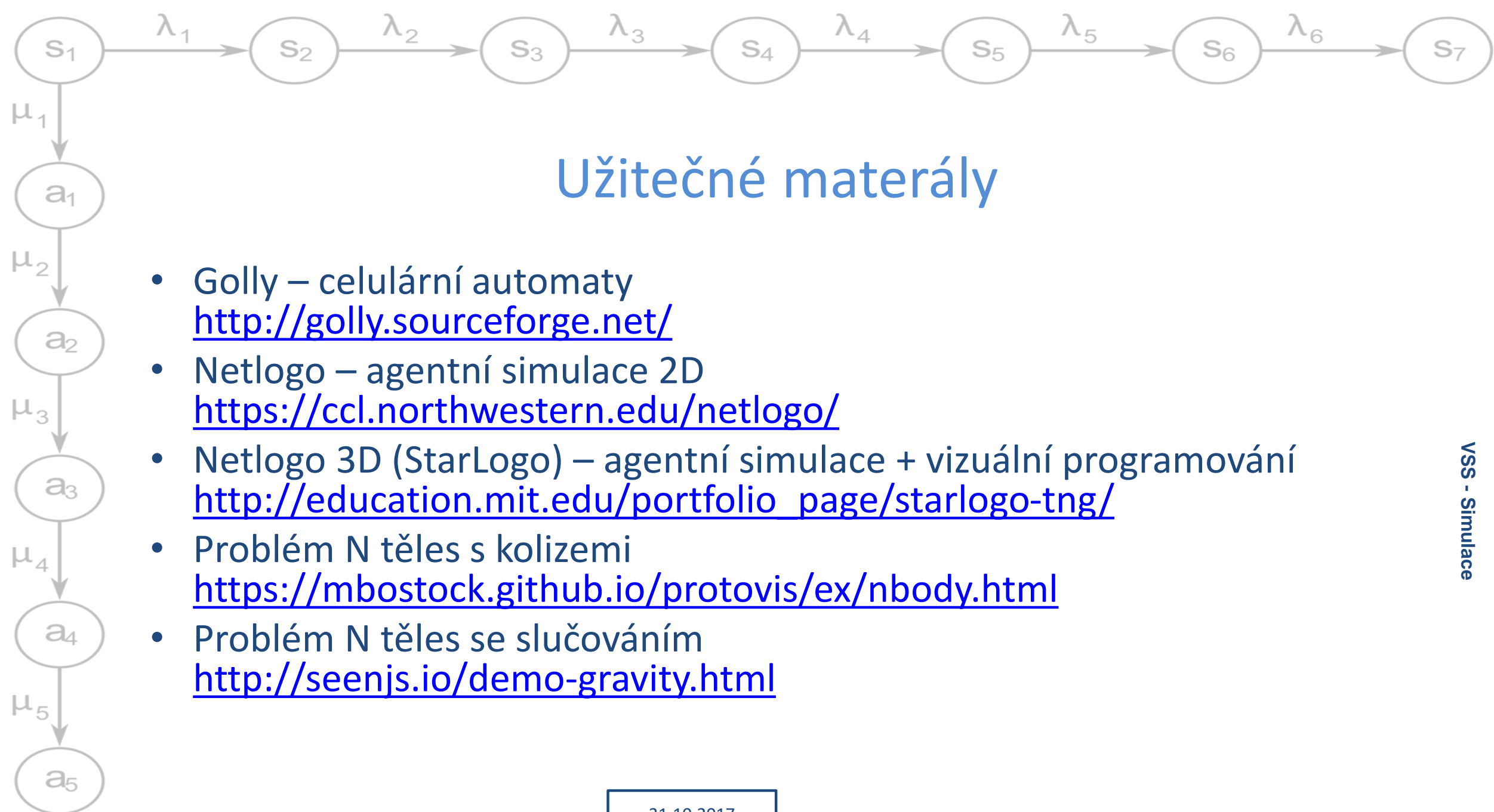


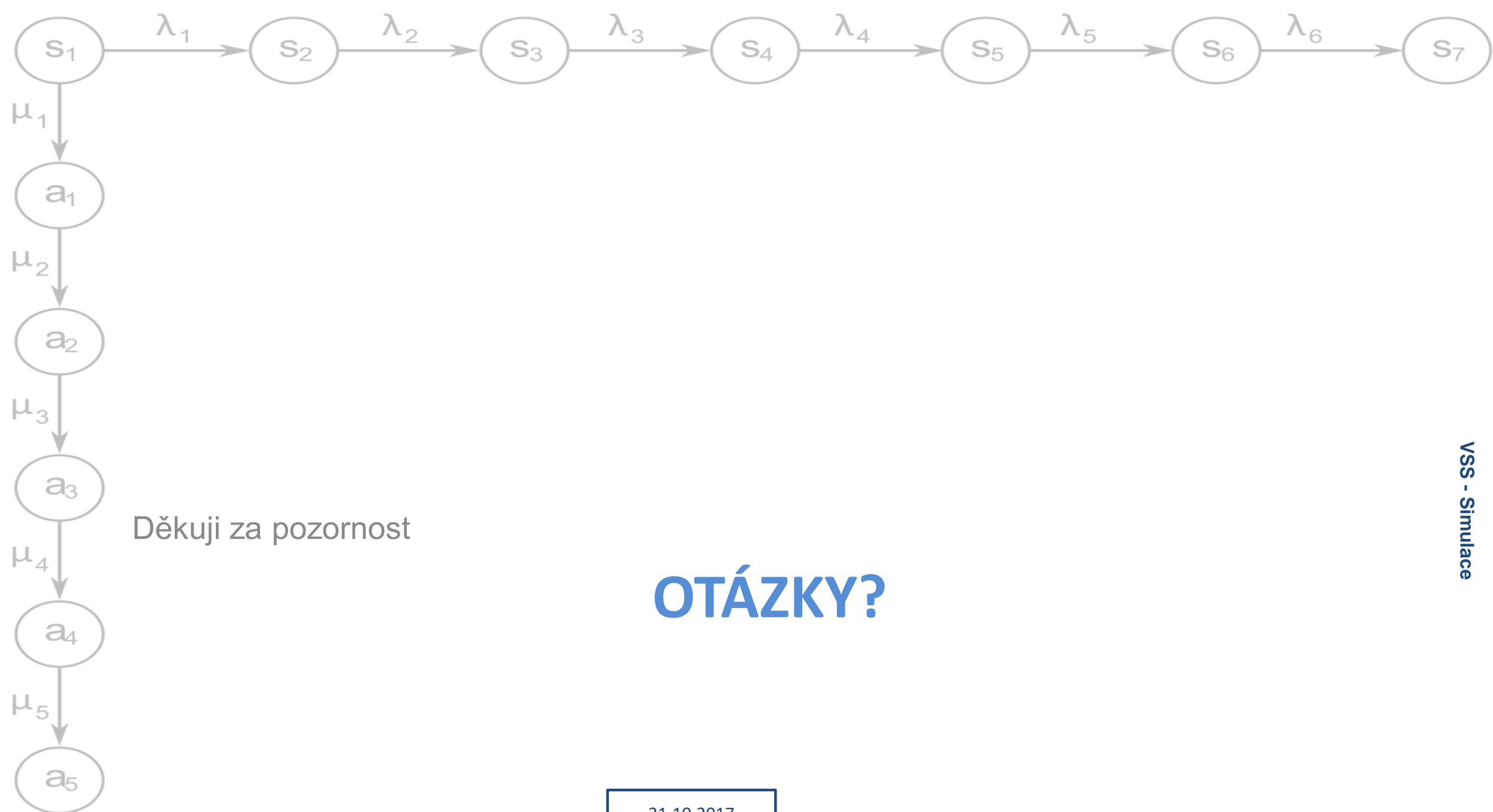
1D automaty

- Systematicky popsány Stephenem Wolframem (1981)
 - Jednoduchý popis ale překvapivě složité chování
- 256 možných pravidel
 - Pravidlo 184 – dopravní simulace
- Řada vzorů odpovídacích přírodním nebo matematickým jevům









Děkuji za pozornost

OTÁZKY?