DOI: 10.35191/medsoft_2025_1_37_vojtech

„OD UMĚNÍ K ALGORITMU" – FORMALIZACE A MODELOVÁNÍ LEGISLATIVNÍCH PROCESŮ SE ZAMĚŘENÍM NA ZDRAVOTNICTVÍ

Adam Vojtěch, Vojtěch Merunka, Jiří Kofránek

Abstrakt

Článek se zabývá formalizací a modelováním legislativních procesů ve zdravotnictví. Zdůrazňuje důležitost spolupráce mezi architekty, implementátory a uživateli informačních systémů od samého začátku jejich vývoje. Upozorňuje na úskalí tvorby informačních systémů pro veřejnou správu, která vyžadují zabudování přesného popisu procesů používaných v legislativě. Na příkladu vlastních zkušeností s řešením problematiky ePreskripce a tvorby legislativy pro digitalizaci záznamů o očkování ukazuje důležitost využití metod a nástrojů vizuální simulace pro usnadnění vzájemného porozumění mezi zdravotnickými pracovníky, zákonodárci a vývojáři softwaru, pro včasnou detekci chyb a pro výrazné zkrácení legislativního procesu. Článek se dále zabývá současnými metodami formalizace legislativní tvorby. Popisuje slibnou, mezinárodně standardizovanou objektově procesní metodologii (OPM) a naše zkušenosti s nástrojem OPCloud, který ji implementuje. Závěrem jsou formulována doporučení pro budoucí strategii a implementaci formalizačních přístupů, které usnadňují interdisciplinární spolupráci a vývoj hybridních přístupů kombinujících formální metody analýzy procesů s jazykovými modely umělé inteligence.

Klíčová slova

AI, formalizace, modelování, jazykové modely, legislativa, eHealth

1 Úvod – od katedrál k eHealth

1.1 Proč digitální světy stavíme jinak než Parléř chrám sv. Víta

Středověcí mistři, jako byl Petr Parléř při stavbě chrámu sv. Víta, ztělesňovali dvojjedinou profesi: byli zároveň návrháři i realizátory. Jejich komplexní a ‚esoterické' know-how se předávalo z generace na generaci a položilo základy modernímu inženýrství. Právě od bratrstev těchto kvalifikovaných stavitelů a kameníků, sdružených ve stavebních hutích – stavebních a kamenických dílnách, ve Francii nazývaných „loges" – se odvozuje název a původ zednářských lóží.

S rozvojem technických oborů se profese logicky rozdělila na **architekty** a **stavaře**. Architekti navrhují stavby z hlediska jejich ztvárnění a účelu, kterému mají sloužit. Řeší tedy především otázku **CO** chce uživatel od budoucí stavby. Ve svém návrhu spojují estetické i stavební znalosti. Plánují vnitřní i vnější uspořádání stavby tak, aby co nejlépe odpovídala potřebám budoucích uživatelů. Navrhují tvary a materiály s ohledem na to, aby stavba byla technicky proveditelná. Stavební inženýři pak řeší otázku **JAK** realizovat stavbu podle architektonického návrhu a jak zajistit její stabilitu a bezpečnost. Řídí postupnou realizaci a odpovídají za to, aby výsledná stavba byla z technického hlediska robustní a odpovídala původnímu záměru architekta.

Podobné rozdělení na **architekty** a **realizátory** se objevuje i u tvorby **informačních systémů**. Ta se totiž zdaleka netýká pouze naprogramování samotného softwaru, ale především návrhu řešení potřeb uživatelů a popisu začlenění systému do procesů organizace, pro kterou vzniká.

- Návrh informačního systému jako celku je úlohou **architektů informačního systému**. Ti odpovídají na otázku **CO**: jak má systém vypadat, jaké má mít funkce a jak má být členěn, aby naplnil svůj účel. Navrhují strukturu, komponenty, rozhraní a technologie s ohledem na funkčnost, bezpečnost, výkon a budoucí rozvoj. V neposlední řadě dávají návrhu technickou proveditelnost (obdobně jako architekt ve stavebnictví zohledňuje statiku).
- **Realizátoři informačního systému (programátoři)** jsou odpovědni za otázku JAK technicky implementovat (naprogramovat) architektonický návrh. Zajišťují správné a robustní sestavení jednotlivých částí systému (psaní kódu, testování, nasazení), aby byl výsledek stabilní, bezpečný a odpovídal záměru architekta.

Pro efektivní spolupráci využívají architekti a realizátoři srozumitelné komunikační nástroje. Patří sem nejrůznější standardizované diagramy a vizualizace, které definují strukturu a vztahy v systému na různých úrovních detailu. Postupem doby vznikla řada standardů (např. UML, SysML aj.), architektonických principů a osvědčených návrhových vzorů (design patterns) pro psaní kódu, které udržují konzistenci a snižují mentální námahu (tzv. kognitivní zátěž) programátorského týmu.

1.2 Úskalí spolupráce architektů a realizátorů informačních systémů

Na rozdíl od stavební praxe (kde se i malé domy obvykle staví podle určitého, byť typizovaného návrhu) vykazuje spolupráce mezi architekty a realizátory (programátory) při stavbě informačních systémů (IS) řadu úskalí:

- **Špatná nebo chybějící spolupráce s architektem**: Důsledkem je vznik systémů ‚lepených jako vlaštovčí hnízdo', které lze jen stěží modifikovat podle nových požadavků uživatelů.
- **Neúplný návrh architektury:** Podcenění spolupráce s budoucími uživateli vede k tomu, že systém neplní vše, co zákazník očekával. Nové požadavky se pak objevují až při testování v pilotním provozu, nebo dokonce až v ostré praxi.
- **Prolínání fází návrhu a realizace:** V praxi se tyto fáze často překrývají, což vede k neustálým změnám během výstavby. Programátoři provádějí **ad-hoc změny v architektuře** bez konzultace s architektem, obvykle pod tlakem na rychlé dodání funkčnosti. Systém se tak stává nekonzistentním a vzniká tzv. **technický dluh**, který snižuje jeho stabilitu a udržovatelnost.
- **Podcenění analýzy požadavků:** Při stavbě budov se nejdříve vybírá architektonický návrh a až poté dodavatel. V IT je analýza toho, **CO** uživatel potřebuje **(analýza požadavků)**, klíčovou součástí návrhu. Tato fáze se však často opomíjí a vypisuje se jedno výběrové řízení na návrh i realizaci současně, často s vágním zadáním přímo od uživatele. To vytváří prostor pro tzv. „systémové integrátory", kteří poskytují obě služby (jak tvorbu architektury, tak i její následnou realizaci). Tato konsolidace je sice efektivní, ale nese riziko: integrátor má sklon navrhovat řešení, která preferují jeho vlastní nástroje a technologie, jež nemusí být pro zákazníka optimální. Obrazně řečeno je to podobné, jako když stavební firma vyhraje soutěž na kostel a – protože vlastní panelárnu – postaví svatostánek z betonových panelů.
- **Vendor lock-in (neboli uzamčení na dodavatele):** Dodavatelé, zejména ti, kteří poskytují architekturu i softwarové řešení, mají sklon vytvářet závislost zákazníka na výhradním využívání jejich produktů a služeb. Zákazník je tak při jakékoli modifikaci nebo dalším rozvoji informačních služeb **„uzamčen"** na původního prodejce. Přechod k jinému dodavateli je poté velmi obtížný (či dokonce nemožný), aniž by hrozilo riziko značných **finančních** ztrát, technických komplikací nebo přerušení provozu. V krajním případě se může vyplatit začít budovat celý informační systém znovu od samého počátku.

DOI: 10.35191/medsoft_2025_1_37_vojtech

Názorným příkladem tohoto jevu je osud první verze informačního systému eReceptu, o kterém budeme mluvit později (viz kapitolu 2.1).

Při budování rozsáhlých IS, kam nepochybně patří i systémy ve zdravotnictví, je nutné se těmto úskalím vyhnout. Zásadní je zejména úzká spolupráce architektů s budoucími uživateli.

V návrhu nejde jen o vytvoření plánu pro programátory. Klíčové je nejprve jasně analyzovat, **CO je cílem informatizace celého systému** a jeho jednotlivých účastníků – zainteresovaných stran (v poangličtěné terminologii stakeholderů). Je nutné analyzovat procesy uvnitř těchto stran i mezi nimi, tedy provést tzv. **procesní analýzu**. Výsledkem je vytvoření tzv. „byznys vrstvy“ vícevrstevné architektury (obr. 1). Slovo „business“ zde není spojeno s obchodem; jde o terminologický úzus, který se v tomto kontextu týká spíše procesů a provozu. Srozumitelnější by v češtině bylo mluvit o **procesní (nebo provozní) vrstvě** architektury.

Právě procesní analýza a návrh provozní vrstvy vyžadují **nejužší součinnost** mezi **softwarovými architekty** a **budoucími uživateli**. Ve zdravotnictví však tato spolupráce bývá obzvláště náročná.

1.3 Spolupráce informatiků a zdravotníků při budování informačních systémů ve zdravotnictví

Procesy ve zdravotnictví jsou složité. Jejich tvorba klade velké nároky na multioborové porozumění mezi zdravotníky a informatiky.

Zdravotnictví a informační technologie jsou dva různé světy, což přináší rozdíly v myšlení i profesním zaměření jejich pracovníků. Obě odvětví se neustále specializují. Přední analytik nebo systémový architekt často postrádá hlubší znalost procesů ve zdravotnictví, a to i v oblastech úzce spojených s IT, jako je například elektronická preskripce, využívání analýzy obrazů a strojového učení pro diagnostiku a další oblasti eHealth. Stejně tak lékaři a zdravotnický personál mají blíž k přírodním vědám než ke světu algoritmů, databází a kódování.

Pro efektivní tvorbu informačních systémů je proto klíčové moderovat diskusi mezi všemi zainteresovanými stranami ve zdravotnictví (lékaři, farmaceuty, pojišťovnami, pacientskými asociacemi aj.). Výsledné shody a dohody je nezbytné transformovat do přesné a formalizované podoby, která poslouží jako:

1. **Architektonický základ** pro implementaci informačních systémů.
2. **Srozumitelný popis** pro všechny zúčastněné odborníky.
3. Primární podklad pro tvorbu **navazující legislativy**.

Tento přístup je zásadní pro zajištění souladu mezi potřebami praxe, technologickou architekturou a stávajícím právním rámcem.

Informační systémy ve zdravotnictví jsou navíc úzce propojeny se systémy státní a veřejné správy, což přináší další problémy spojené s nutností jejich legislativního zabezpečení.

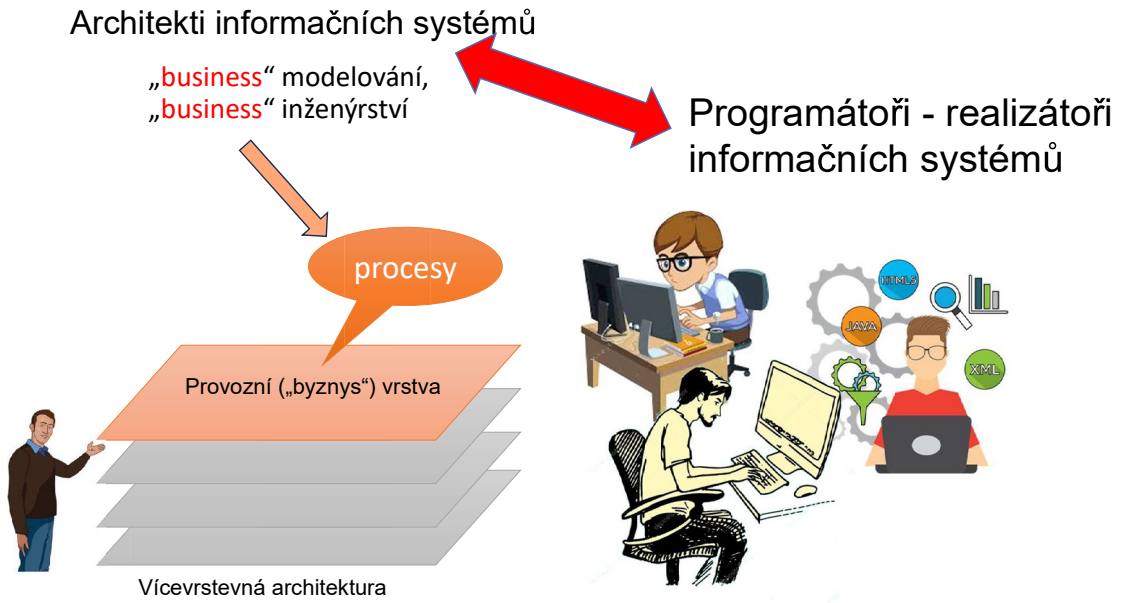
1.4 eHealth a legislativní smyčka – jak uhlídat citlivá data (a proč to musí být v zákoně)

Budování **státních informačních systémů** v demokratických zemích podléhá zásadním právním omezením. Zatímco občan může činit vše, co zákon nezakazuje, **stát smí vykonávat jen to, co mu zákon výslovně povoluje**.

To znamená, že každá činnost informačních systémů, kterou orgány veřejné moci využívají, musí být legislativně ukotvena. V zákoně **musí být přesně popsáno**, kde, kdy a jakým způsobem informační systém s citlivými údaji pracuje. Cílem je předejít úniku dat a zajistit, aby se stát prostřednictvím svých systémů nestal „Velkým bratrem“.

To se týká zejména zdravotnických informačních systémů. Elektronizace zdravotnictví umožňuje sdílení dat mezi poskytovateli péče, nahlížení do digitálních záznamů či vytváření klinických a administrativních registrů pro řízení zdravotnictví na základě velkých dat. Všechny tyto procesy musí být zajištěny na úrovni státu, pro který platí, že orgány veřejné moci mohou konat pouze v mezích zákona. Jelikož se ve zdravotnictví pracuje s velmi citlivými údaji, je nutné při jejich ukládání a sdílení chránit soukromí pacientů a eliminovat riziko, že k datům získá přístup neoprávněný subjekt.

Digitalizace zdravotnictví tak klade značné nároky na tvorbu **kvalitní legislativy**, zejména s ohledem na složitost procesů



Obrázek 1 – Spolupráce architektů informačních systémů a programátorů realizujících architektonický záměr. Úlohou architektů je především jasně analyzovat, CO je cílem, pro který je informační systém vytvářen, jaké jsou požadavky jeho budoucích uživatelů a jak bude IS začleněn do provozu a procesů celé organizace. Na základě této analýzy navrhují optimální strukturu těchto procesů. Klíčové je správně navrhnout tzv. provozní (procesní) vrstvu celé vícevrstvé architektury, aby vytvářený systém naplnil účel a potřeby svých uživatelů. Architekti definují komponenty, rozhraní a jejich vzájemné vztahy, čímž vytvářejí zřetelný a jasný plán, podle kterého pak programátoři realizují návrh softwarové architektury. Úlohou programátorů je pak nalézt způsob, JAK navrženou architekturu optimálně technicky realizovat.

propojujících odborníky z různých oblastí. Současné trendy vyžadují nejen technologické inovace, ale také důslednou komunikaci napříč profesemi, která zajistí účinnou implementaci nových systémů v praxi.

Legislativa pro eHealth má dva základní úkoly:

1. **Popis procesů:** Musí přesně a podrobně definovat způsob využití systémů, na kterém se shodli architekti i zdravotničtí profesionálové.
2. **Ochrana dat:** Musí striktně – organizačně i architekturou softwaru – zajistit ochranu osobních údajů a toto zabezpečení legislativně ukotvit.

Při tvorbě regulace spolu komunikují **tři skupiny účastníků**, jejichž vzdělání a přístup se mohou zdát na první pohled nekompatibilní: **zdravotníci** (uživatelé nástrojů digitalizace), **IT specialisté** (programátoři a architekti) a **legislativci** (tvůrci regulačního rámce) – viz obr. 2.

Obrázek 2 – Při tvorbě zdravotnických informačních systémů je klíčovým problémem mezioborové porozumění mezi zdravotníky, informatiky a tvůrci legislativy.

Aby jejich komunikace byla efektivní, je třeba využít **společný nástroj**, který od počátku slouží k **vizualizaci** procesů a objektů. Tento nástroj musí být **srozumitelný** pro všechny strany a zároveň musí prostřednictvím simulace umožnit odhalení chyb či slepých míst, která jsou v komplikovaném legislativním textu těžko detekovatelná. Jako ideální řešení pro vizualizaci, simulaci a optimalizaci kroků obsažených v právním předpisu se ukazuje **procesní modelování**.

2 Naše zkušenost s tvorbou systémů elektronické preskripce a elektronického očkovacího průkazu

2.1 Vznik elektronického předepisování v ČR

První snahy o elektronické předepisování léčivých přípravků v ČR sahají do roku 2006. V roce 2007 byl Státní ústav pro kontrolu léčiv (SÚKL) pověřen vývojem a správou systému eRecept na základě zákona č. 378/2007 Sb., o léčivech. Možnost předepisování léků elektronickou formou byla zavedena v roce 2011, zpočátku však lékaři ani lékárníci systém příliš nevyužívali. Kvůli nízkému zájmu bylo politicky rozhodnuto o **povinném zavedení eReceptu** pro všechny lékaře a lékárny od **ledna 2015.**

Dodavatelem softwaru (včetně systému eRecept) pro SÚKL byla firma Tronevia (a její předchůdci), což se v roce 2015 ukázalo jako problematické. Po vypršení servisních smluv a v době, kdy se SÚKL chystal vypsat nová výběrová řízení, začala Tronevia uplatňovat nároky na autorská práva k softwarovým aplikacím. Za autorské dílo a jeho užívání požadovala vysoké finanční částky, původně až 167 milionů korun. Na základě žaloby firmy Tronevia vydal Městský soud v Praze předběžné opatření, které SÚKL znemožnilo přístup k některým aplikacím včetně eReceptu (konkrétně k tehdejší verzi Centrálního úložiště elektronických receptů). Ministr zdravotnictví (Svatopluk Němeček) a vedení SÚKL tehdy požadavky firmy ostře odmítli jako nepřiměřené finanční ‚vydírání‘. SÚKL trval na tom, že za dodané služby řádně zaplatil, a **termín povinného používání eReceptu** byl posunut na **leden 2018**.

Kvůli soudní blokaci musel SÚKL urychleně vyvinout a nasadit nový, náhradní systém elektronické preskripce, aby zajistil funkčnost od března 2016. Tím se definitivně odstřihl od softwaru společnosti Tronevia. Spor s firmou byl nakonec uzavřen soudním smírem až v roce 2022. SÚKL na základě smíru zaplatil firmě Tronevia vyrovnání ve výši téměř 14,5 milionu korun, aniž by uznal původní požadavek na omluvu nebo plnou výši autorských nároků. Systémy, kterých se spor týkal, již SÚKL v té době dávno nepoužíval.

2.2 Rozšíření funkčnosti eReceptu

V prosinci 2017, kdy byl autor tohoto příspěvku Adam Vojtěch jmenován ministrem zdravotnictví, zůstával počet lékařů využívajících elektronickou preskripci stále nízký, v řádu jednotek procent. Od **ledna 2018** navíc lékařům hrozily finanční sankce za nepoužívání eReceptu, což ve zdravotnické veřejnosti vzbuzovalo značné napětí. Hlavním důvodem odporu bylo, že lékaři ani lékárníci neviděli v elektronizaci žádnou zásadní výhodu oproti tradičním receptům. Většina lékařů v té době navíc nedisponovala kvalifikovaným elektronickým podpisem, který byl nezbytný pro ověření identity.

Aby elektronická preskripce dávala skutečný zdravotní smysl (umožnila posouzení lékových interakcí či výběr nejvhodnější farmakoterapie) a nebyla jen pouhou náhradou papíru elektronickým kódem, bylo nutné, aby lékař i lékárník měli přehled **o všech lécích předepsaných a vydaných danému pacientovi.** Systém eReceptu bylo proto nezbytné rozšířit o další funkcionality. To vyžadovalo jak změny v softwarové architektuře, tak vytvoření navazujících změn v legislativě (obr. 3).

Obrázek 3 – V lednu 2018 bylo podle tehdejší legislativy nutné začít povinně používat eRecept. Většina lékařů však v té době neměla kvalifikovaný elektronický podpis nezbytný pro elektronickou preskripci. Sankce za nepoužívání eReceptu byly odloženy o rok a systém bylo nutné rozšířit o další funkcionality (zejména pacientský lékový záznam), které dodaly elektronické preskripci skutečný zdravotní smysl.

Během přípravy nové funkčnosti bylo zapotřebí procesy elektronické preskripce **důkladně prodiskutovat** se všemi zainteresovanými stranami – Českou lékařskou komorou, zástupci lékárníků, pojišťovnami, odbornou veřejností, tvůrci softwarové architektury i správci základních registrů eGovernmentu.

Na základě výsledků těchto diskusí bylo možné **upravit stávající software** a zároveň vypracovat novou legislativu, která podrobně popsala všechny procesy související s rozšířením eReceptu (tzv. lékový záznam). Následně byl vypracován návrh novely zákona, který prošel schvalovacím procesem v Parlamentu ČR.

Za standardních okolností trvá tento proces přibližně dva roky. My jsme se jej rozhodli urychlit využitím technologie vizuální reprezentace struktury propojených objektů a procesů. Tato metoda výrazně napomohla mezioborovému porozumění mezi programátory, zdravotníky a autory legislativy.

DOI: 10.35191/medsoft_2025_1_37_vojtech

2.3 Vizualizace návrhu nové softwarové i legislativní struktury v metodologii BORM

Jako základ pro vizualizaci jsme zvolili metodiku **Business Object Relationship Modelling (BORM)** [1–7], která se objevila na počátku 90. let. Vznikla v rámci česko-britského projektu VAPPIENS financovaného Britskou radou, což jí poskytlo solidní teoretický základ a ranou konceptualizaci. Klíčovou fází bylo spojení s poradenskou firmou Deloitte v roce 1996, která začala metodiku ve spolupráci s akademickým sektorem aktivně podporovat a využívat v rozsáhlých projektech.

Hlavní silnou stránkou BORM je využití **velmi jednoduché grafické notace**, která je snadno srozumitelná i pro netechnické uživatele. Tento princip řeší běžný problém při vývoji systémů: obtížnost, s níž se při návrhu **procesní architektury** potýkají zúčastněné strany (zejména budoucí uživatelé), které nemusí vždy rozumět složitým technickým aspektům. Cílem je umožnit jim pochopit, jak bude systém fungovat, a na základě diskuse dospět k jeho optimálnímu návrhu.

Praktická aplikace a šíření metodiky BORM jsou úzce spojeny se softwarovým nástrojem **Craft.CASE** [8], který byl speciálně vyvinut pro podporu této metodiky. Za jeho vývojem a patentovanou technologií [9] stojí **Jiří Berger.** Craft.CASE je, obrazně řečeno, jakási pomyslná „tužka" na navrhování a modelování procesní architektury systému.

Craft.CASE není pouze nástroj pro zobrazování architektury, ale slouží také jako modelovací prostředí, které umožňuje testovat chování navrhovaného systému a včas odhalovat chyby v návrhu. To má mimořádný význam u netechnických systémů, kde je největším úskalím mezioborové porozumění mezi architekty a uživateli. Výhoda názorného zobrazení procesů se zde spojuje s možností testovat jejich chování pomocí simulačního modelu.

Jako příklad můžeme uvést použití Craft.CASE při návrhu legislativy občanského zákoníku (viz obr. 4). Procesní model je zde zobrazen pomocí komunikujících objektů, které jsou v daném čase charakterizovány určitým stavem. Přechod objektu z jednoho stavu do druhého je výsledkem příslušné aktivity (procesu) a může být podmíněn splněním zadaných podmínek. Spuštění procesu může být iniciováno také aktivitou v jiném objektu. Tímto způsobem si objekty mezi sebou předávají data a komunikují. To umožnilo srozumitelným způsobem formálně popsat procesy, které navrhovaná legislativa řešila.

Podstatné je, že nástroj Craft.CASE umožnil pomocí simulace otestovat chování všech prvků – například zjistit, zda se některý proces nedostane do situace, kdy marně čeká na informaci z jiného objektu. Dochází tak k odhalení tzv. uváznutí (deadlock), kdy se posloupnost přechodů v objektu zastaví. Při návrhu občanského zákoníku tak procesní modelování umožnilo odhalit řadu legislativních chyb již v průběhu jeho přípravy

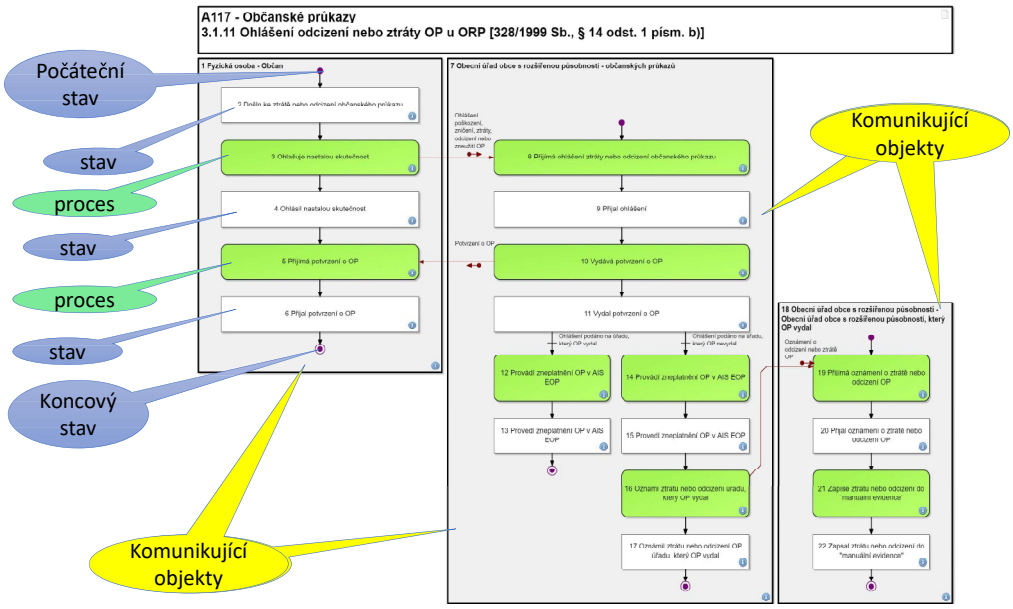
2.4 Využití nástroje Craft.CASE a technologie BORM pro návrh softwarové i legislativní architektury pro zlepšení funkčnosti eReceptu

Vývojový nástroj Craft.CASE:

- je interaktivním nástrojem pro postupné vytváření popisu objektů, procesů a jejich propojení;
- umožňuje přesný popis procesní architektury;
- slouží jako vizualizační nástroj, který usnadňuje mezioborové porozumění;
- umožňuje pomocí simulace zobrazit chování propojených objektů a procesů;
- vytváří exaktní podklady pro přípravu legislativy i pro návrh procesní architektury softwarového řešení.

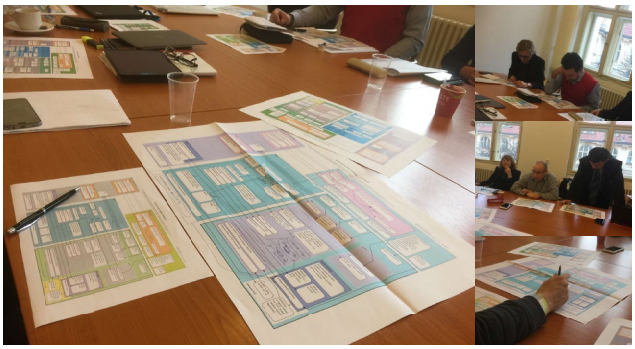
Proto byl Craft.CASE zvolen jako základní vývojový nástroj pro návrh nové softwarové i legislativní verze eReceptu. Jeho autor, Jiří Berger, byl klíčovým členem vývojového týmu.

Pro vytvoření nové architektury eReceptu s rozšířenou funkcionalitou lékového záznamu jsme organizovali řadu intenzivních workshopů. Těch se zúčastnily všechny zainteresované strany: od softwarových vývojářů přes zástupce lékáren, pojišťoven a legislativců až po Českou lékařskou komoru a další odborníky. Na základě těchto diskusí jsme v systému Craft.CASE postupně upravovali vizualizovaný návrh procesní architektury (viz obr. 5).



Obrázek 4 – Softwarový nástroj Craft.CASE realizující metodu BORM byl prakticky vyzkoušen při tvorbě občanského zákoníku. Procesní model je vizualizován pomocí komunikujících objektů, které jsou v daném čase charakterizovány určitým stavem. Přechod z jednoho stavu do druhého je výsledkem působení příslušného procesu a může být podmíněn splněním zadaných podmínek. Proces může být také iniciován aktivitou – procesem probíhajícím v jiném objektu – tímto způsobem spolu objekty komunikují a předávají si data. Stavy v objektech by měly postupně přecházet od počátečního stavu ke koncovému stavu. Craft.CASE umožňuje tuto komunikaci a přechody stavů simulovat. Při chybném návrhu se může stát, že objekt marným čekáním na informaci z jiného objektu uvázne; tato chyba se díky simulaci odhalí dříve, než je text zákona finalizován. Tímto způsobem bylo možné identifikovat řadu legislativních nedostatků již v průběhu přípravy návrhu.

DOI: 10.35191/medsoft_2025_1_37_vojtech



Obrázek 5 – Počátkem roku 2018 jsme organizovali řadu intenzivních pracovních workshopů, kterých se zúčastnily všechny zainteresované strany (od softwarových vývojářů přes představitele lékáren, pojišťoven a legislativců až po zástupce lékařské komory a další odborníky). Společně jsme diskutovali a upravovali vizualizovaný návrh procesní architektury eReceptu, vytvářený pomocí nástroje Craft.CASE podle metodologie BORM.

Na souhrnném grafickém schématu, vygenerovaném systémem Craft.CASE, jsou přesně popsány procesy účastníků i jejich vzájemné ovlivňování. Na pozadí těchto diagramů jsou uloženy podrobné scénáře, které umožnily pomocí simulace ověřit, zda navržená struktura neobsahuje logické rozpory a chová se podle původních záměrů.

Během workshopů se do architektury postupně vnášely změny, které se promítly do další verze návrhu. Ten se tak vyvíjel od původní podoby (obr. 6) až po finální, desátou verzi (obr. 7), kterou odsouhlasily všechny strany. Tato verze se stala odrazovým můstkem pro úpravu softwaru i pro vytvoření legislativy popisující zvolenou architekturu. Diagram byl rovněž součástí důvodové zprávy k návrhu novely zákona o léčivech.

Součástí řešení byla i sada procesních diagramů detailně popisujících jednotlivé scénáře. Například na obrázku 8 je zachycen proces předpisu léčivého přípravku, včetně stavů a procesů v komunikujících objektech. Tento diagram není jen pasivním obrázkem, ale interaktivně simuluje souvislosti, čímž testuje správnost celého modelu.

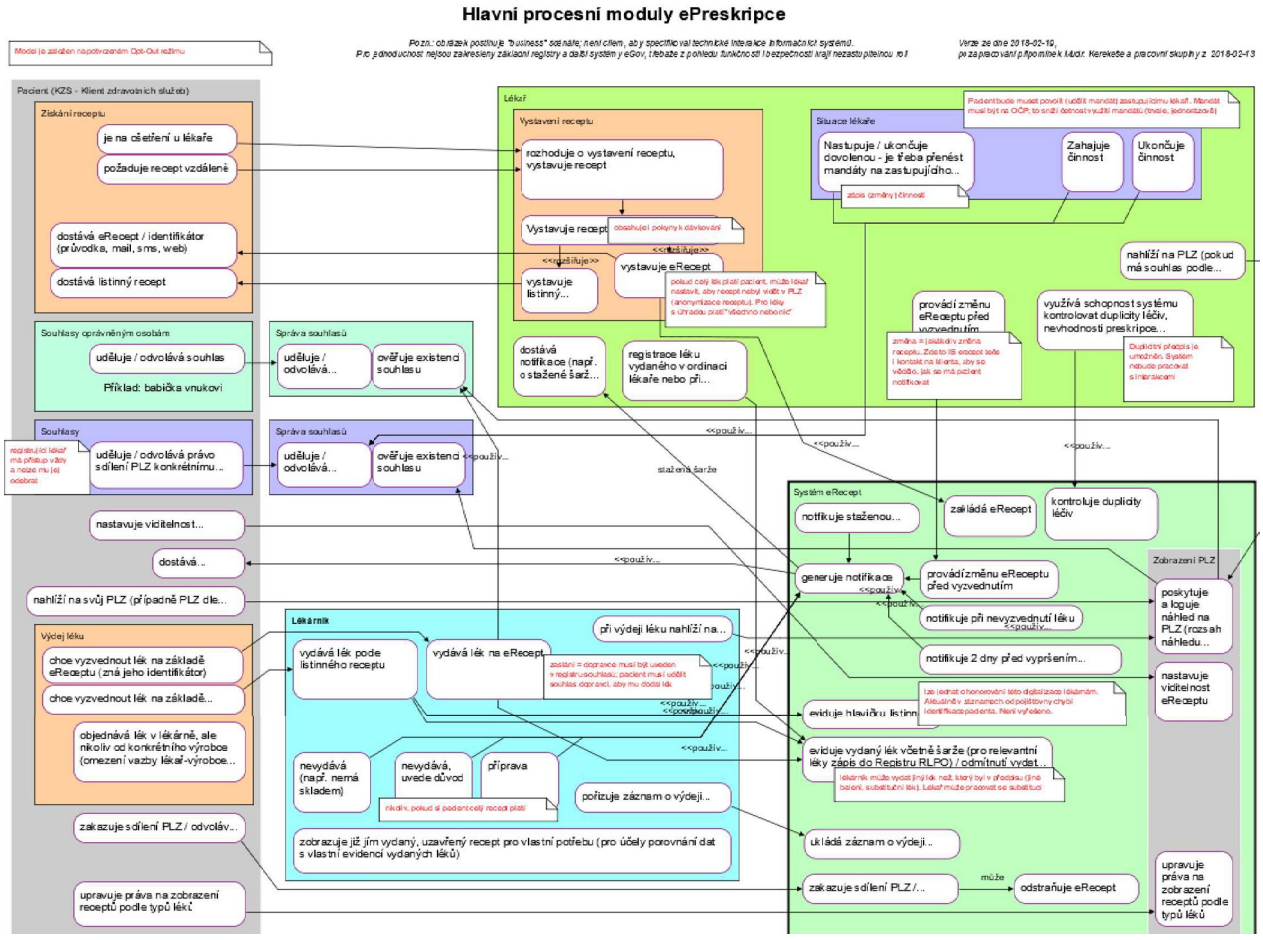
Díky tomuto postupu se podařilo v rekordně krátké době připravit legislativu i modifikovat software pro zavedení pacientského lékového záznamu [10–13]. V prosinci 2019 byl zákon schválen Parlamentem ČR a systém se stal plně funkčním.

Zavedení eReceptu v roce 2018 bylo zlomovým bodem. Graf na obrázku 9 ukazuje, jak kombinace zákonné povinnosti a nových výhod katapultovala využívání systému. Zatímco koncem roku 2017 tvořily elektronické recepty jednotky procent, v roce 2018 podíl vzrostl na 80 % a v roce 2019 již překročil 95 %.

Počátkem roku 2020 vypukla pandemie covidu-19 – díky funkčnímu eReceptu se chroničtí pacienti mohli dostat ke svým lékům distančně, což v kritické době zásadně pomohlo českému zdravotnictví.

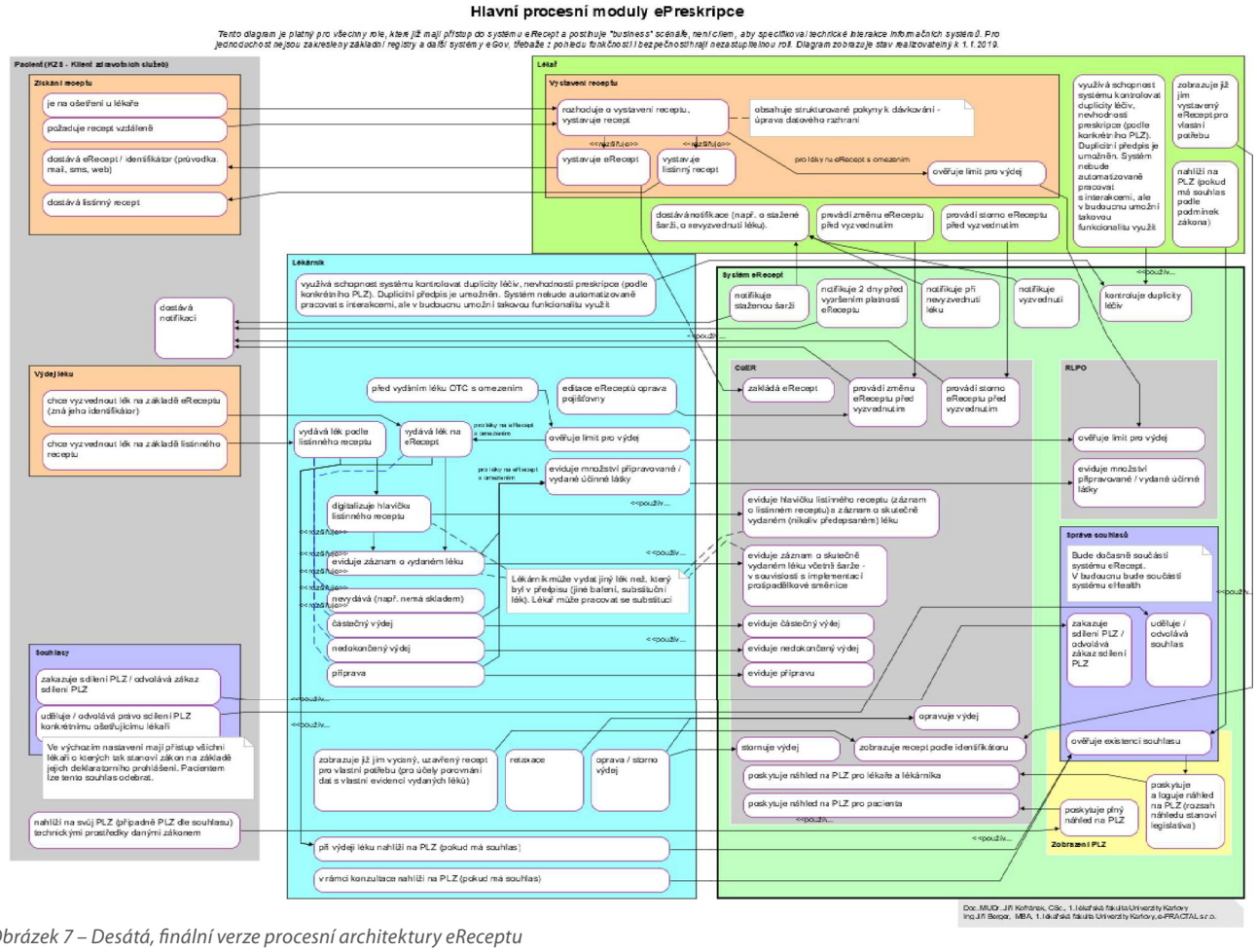
2.5 Vytvoření legislativy pro elektronický očkovací průkaz

Klíčovou součástí elektronické preskripce je lékový záznam, který obsahuje přehled všech pacientovi předepsaných a vyzvednutých léků. Díky němu mohou lékaři i lékárníci posoudit vzájemné interakce a inkompatibility léčiv. Tento systém navíc umožňuje propojení s dalšími aplikacemi, například pro personalizovanou preskripci.

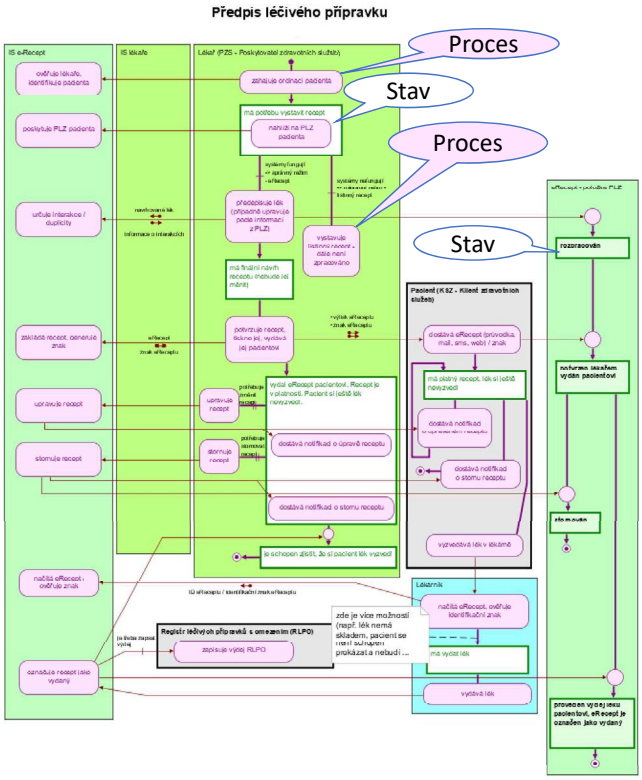


Obrázek 6 – První verze procesní architektury eReceptu

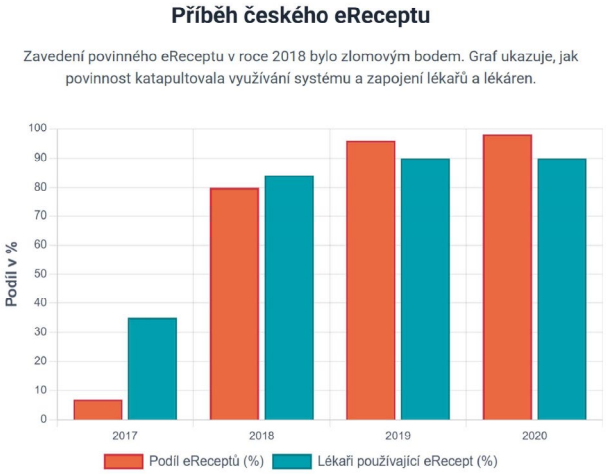
DOI: 10.35191/medsoft_2025_1_37_vojtech



Obrázek 7 – Desátá, finální verze procesní architektury eReceptu



Obrázek 8 – Procesní diagram popisující předpis léčivého přípravku, jednotlivé stavy a procesy v komunikujících objektech. Simulace v nástroji Craft.CASE umožnila otestovat a ověřit dynamické chování tohoto modelu.



Obrázek 9 – Ještě na konci roku 2017 se podíl eReceptů pohyboval v jednotkách procent. Zásadní zlom nastal v roce 2018 se zavedením povinnosti a nových funkcionalit; o rok později již podíl elektronických receptů přesáhl 95 %.

Zatímco s příchodem pandemie covidu-19 se narychlo zavedla elektronická evidence očkování proti tomuto onemocnění, ostatní očkování digitalizována nebyla. Záznamy o nich zůstávaly roztříštěné v papírové dokumentaci u jednotlivých lékařů a pacienti k nim neměli elektronický přístup.

Rozhodli jsme se proto systém eRecept a navazující lékový záznam rozšířit o evidenci veškerých absolvovaných očkování a vytvořit pro její zavedení potřebné legislativní podmínky. Elektronický očkovací průkaz může fungovat na stejném principu jako lékový záznam. Při návrhu legislativy i procesní archi-

DOI: 10.35191/medsoft_2025_1_37_vojtech

tektury bylo možné vyjít z osvědčené struktury elektronického receptu, kterou stačilo doplnit o specifika spojená s očkováním. Díky tomu byl procesní model očkovacího záznamu vypracován za pouhé tři týdny.

Na obrázku 10 je zachycena struktura návrhu procesní architektury eReceptu, rozšířená o legislativní podporu elektronického očkovacího průkazu. Podle tohoto modelu byl následně vytvořen legislativní návrh, který poslanec Adam Vojtěch (v té době již nebyl ministrem zdravotnictví) předložil dne 13. 4. 2021 jako poslanecký pozměňovací návrh k novele zákona o návykových látkách. Dne 2. 6. 2021 jej Poslanecká sněmovna ve třetím čtení schválila.

Vlastní příprava zákona, od vytvoření procesního modelu až po finální legislativní text, potom trvala zhruba čtyři týdny a již za tři měsíce byl zákon schválen. To ukazuje, jak velkým přínosem může být procesní modelování a s ním spojená formalizace popisu procesů pro efektivní přípravu legislativy [14].

3 Formalizace tvorby legislativy

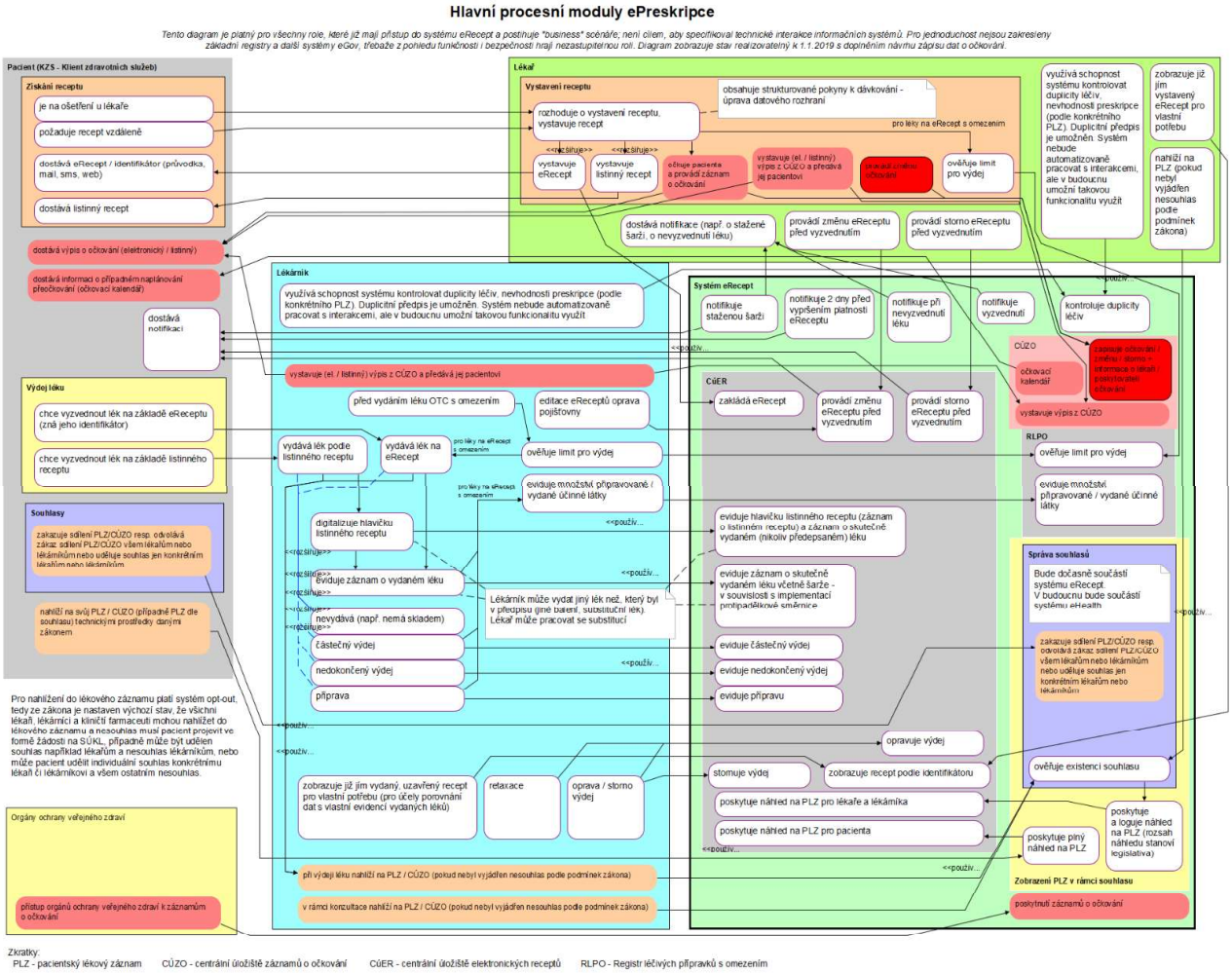
3.1 Cíle a paradigma digitální formalizace legislativy

Formalizace je, stručně řečeno, proces, při kterém nahrazujeme vágní a mnohoznačná slova či popisy (jako je běžná řeč) přesnými symboly a vzorci podle definovaného systému pravidel. V tomto kontextu znamená formalizace přepis vět z běžné češtiny do symbolického zápisu, se kterým lze následně nakládat podle daných pravidel. Cílem je odstranit dvojsmysly: zatímco v běžné řeči může mít jedno slovo více významů, ve formálním systému má každý symbol význam pouze jeden. Formalizace je klíčová zejména při tvorbě softwaru, kde se požadavky na chování programu formalizují do kódu podle pravidel programovacího jazyka.

Digitální formalizace legislativy, často označovaná v širším kontextu jako **computational law** (výpočetní právo), posouvá tradiční pojetí z roviny teoretické do roviny technické a aplikační. Nejde již o pouhou digitalizaci, tedy o prostý převod textu zákona do elektronického formátu (např. PDF), ale o hlubokou a komplexní transformaci právních textů, pravidel a procesů do strukturovaných, strojově čitelných a výpočetně zpracovatelných formátů [15].

Tato transformace využívá formální modely práva, které umožňují reprezentovat nejen text, ale i jeho vnitřní logickou strukturu, hierarchii a explicitní vztahy mezi jednotlivými ustanoveními. Zatímco právní teorie 20. století z velké části opustila rigidní formalismus (protože uznala, že právo je plné nejednoznačností, principů a diskrece, které nelze redukovat na čistou logiku [16]), nástup informatiky si formální modely vynucuje. Digitální formalizace tak představuje pragmatické a inženýrské „znovusetkání“ s formalismem. Cílem není naivně tvrdit, že veškeré právo je algoritmus, ale spíše identifikovat a přesně modelovat ty jeho části, které algoritmizovatelné jsou – v českém kontextu označované jako „plně programovatelné případy“ [17].

Primárním cílem digitální formalizace není jen zpřístupnění práva, ale jeho integrace do digitálních systémů státu a společnosti. To vede k fundamentálnímu paradigmatickému posunu:



Obrázek 10 – Struktura návrhu procesní architektury eReceptu, rozšířená o legislativní podporu elektronického očkovacího průkazu. Ten v principu představuje rozšíření pacientského lékového záznamu o údaje o všech očkováních.

od práva jako pasivního informačního zdroje k právu jako aktivní systémové komponentě. Tradiční právní informační systémy byly navrženy pro vyhledávání (legal search) – jejich cílem bylo pomoci uživateli najít relevantní paragraf [18]. Moderní formalizované systémy směřují k provádění (legal execution) – jejich cílem je spustit logiku obsaženou v paragrafu jako algoritmus (např. „pokud příjem X a status Y, pak nárok Z"), jak naznačují koncepty jako **Rules as Code** [19].

Klíčové cíle tohoto nového paradigmatu jsou:

- Interoperabilita: Schopnost různých národních i mezinárodních systémů (registry, parlamenty, soudy) bezproblémově sdílet a správně interpretovat legislativní data. To je základní předpoklad fungování jednotného digitálního trhu EU [20].
- Automatizace: Možnost automatizovat širokou škálu úkonů, od rutinní kontroly konzistence návrhů zákonů [21] přes automatizované ověřování shody (compliance checking) [22] až po přímé digitální vymáhání některých norem (např. výpočet daní, správa environmentálních zón) [23].
- Podpora rozhodování (Decision Support): Poskytování pokročilých nástrojů pro tvůrce legislativy, které umožňují modelovat dopady navrhovaných změn, a zároveň poskytování srozumitelnějších nástrojů pro adresáty norem [21].
- Transparentnost: Zpřehlednění celého legislativního procesu, od prvotního návrhu až po finální publikaci, a jeho zpřístupnění veřejnosti ve strojově zpracovatelné podobě, např. prostřednictvím portálů eSbírka nebo eLegislativa.

3.2 Nejpoužívanější standardy a sémantika digitální legislativy

Digitální formalizace legislativy není jednotný proces; je to spíše „vrstvená architektura" (layered architecture), která se skládá z několika technologických úrovní. Tyto vrstvy na sebe logicky navazují – od základní struktury dokumentu přes jeho sémantický význam až po jeho spustitelnou logiku.

1. Strukturální formalizace: XML a standard Akoma Ntoso (AKN)

Základní stavební vrstvou je standardizace struktury dokumentu. Bez jednotného formátu pro popis toho, co je „zákon", „paragraf" nebo „odkaz", je jakákoli interoperabilita nemožná. Klíčovým globálním standardem v této oblasti je **Akoma Ntoso (AKN)** [20]. Jedná se o standard mezinárodní organizace OASIS, který je součástí širší iniciativy LegalDocumentML (LegalDocML). AKN je založen na jazyce **XML (eXtensible Markup Language)** [24] a poskytuje podrobný slovník (schéma) pro sémantické a strukturální značkování právních, parlamentních a soudních dokumentů.

Účel AKN není pouze technický, ale především strategický. Jak uvádí specifikace, jeho cílem je definovat společný formát a datový model pro výměnu dokumentů mezi institucemi kdekoli na světě [25]. Pokrývá celý životní cyklus dokumentu – od parlamentních záznamů přes návrhy zákonů a schválenou legislativu až po soudní rozhodnutí. Umožňuje tak budovat moderní, komponentově založené systémy namísto zastaralých monolitů. V kontextu EU je AKN (ve své specifické variantě „AKN4EU") základem pro klíčové nástroje, jako je editor LEOS [26].

2. Sémantická formalizace: Právní ontologie (RDF a OWL)

Samotná struktura XML (Akoma Ntoso) však nestačí. AKN dokáže popsat, že určitý text je paragraf (<paragraph>) nebo odkaz (<ref>), ale nedokáže definovat význam právních vztahů a konceptů v něm obsažených. K tomu slouží druhá, sémantická vrstva, postavená na technologiích sémantického webu [27]:

- **RDF (Resource Description Framework)[28]:** Představuje základní datový model. Místo popisu dokumentů popisuje vztahy pomocí jednoduchých trojic (subjekt–predikát–objekt). Tím umožňuje vytvářet propojené grafy znalostí.
- **OWL (Web Ontology Language)** [29]: Je jazykem pro definování ontologií, tedy formálních, strojově čitelných modelů znalostí v určité doméně. OWL (postavené nad RDF) umožňuje definovat třídy (např. „Smlouva", „Závazek"), jejich vlastnosti a vzájemné logické vztahy (např. „Kupní smlouva je typem smlouvy").

V právním kontextu umožňují ontologie modelovat komplexní právní koncepty, hierarchie norem a inferenční pravidla. To je nezbytný předpoklad pro pokročilé systémy právní analýzy, inteligentní vyhledávání a aplikace umělé inteligence, které potřebují „rozumět" významu textu, nejen jeho struktuře [30].

Práce na právních ontologiích má v České republice tradici, jak dokládá například projekt „Právnický elektronický slovník (PES)" [31]. Jedná se o dlouhodobý interdisciplinární projekt (spolupráce právníků, lingvistů, matematiků a informatiků) zaměřený na analýzu právní terminologie. Z hlediska formalizace je klíčový přístup, který projekt zvolil: místo snahy o vytvoření jedné univerzální právní ontologie a priori (shora dolů), která by byla nevyhnutelně subjektivní a obtížně aplikovatelná, se tvůrci vydali cestou budování ontologií „zdola" (bottom-up). To znamená, že struktura termínů a jejich vztahů je odvozována přímo z analýzy konkrétních textů (např. učebnic práva). Tento přístup lépe respektuje kontextuální povahu právního jazyka.

Nejčastěji současně používané technologie ukazuje tabulka 1.

Atribut	XML / Akoma Ntoso	RDF / OWL	Rules as Code (RaC)
Hlavní účel	Struktura a interoperabilita	Reprezentace a odvozování	Automatizace a exekuce
Základní jednotka	XML element (např. <article>)	Trojice (Subjekt--Predikát-Objekt)	Logický výrok (IF-THEN-ELSE)
Příklad v právu	Označkovaný Termin xt zákona (eSbírka)	Právní ontologie (vztahy mezi normami)	Kód pro výpočet nároku na dávku
Zdroje	[20]	[28,29]	[19]

Tabulka 1 – Srovnání používaných technologií pro digitální formalizaci legislativy

3.3 Procesní analýza a procesní modelování – nástroj mezioborové komunikace i podklad pro tvorbu legislativy

Digitální formalizace se netýká pouze finálního produktu (zákona), ale v rostoucí míře i samotného procesu jeho vzniku. Tvorba legislativního rámce není jen záležitostí zákonodárců, ale všech zájmových skupin, kterých se daná norma týká. Při tvorbě nových předpisů je proto vhodné využívat komunikační mechanismy usnadňující vzájemné pochopení.

Obzvláště důležité je to u norem týkajících se digitalizace veřejné správy. Protože orgány veřejné moci mohou při digitalizaci správních agend činit pouze to, co jim zákon výslovně dovoluje (tzv. princip enumerativnosti veřejnoprávních pretenzí), musí legislativa procesy informatizace popisovat zcela jednoznačně.

Tento princip je naprosto klíčový při informatizaci zdravotnictví, kde se pracuje s velmi citlivými osobními údaji. Při ukládání a sdílení dat je nutné vyhnout se ‚efektu Velkého bratra', tedy chránit soukromí pacientů a eliminovat riziko, že k datům získá přístup neoprávněný subjekt. Aby bylo možné tyto procesy v legislativě přesně definovat, musí být nejprve popsány formou,

které budou rozumět uživatelé systému, IT specialisté i tvůrci legislativy. Formalizace zde není prostředkem pro zjednodušení psaní norem, ale především nástrojem komunikace a mezioborového porozumění.

Pro efektivní komunikaci je třeba od počátku využívat nástroje pro vizualizaci procesů a objektů. Tento popis musí být srozumitelný pro všechny zúčastněné. K tomu slouží standardizované grafické notace, původně odvozené z **UML** (Unified Modeling Language) [32]. Patří mezi ně především **SysML** (Systems Modeling Language) [33], upravený pro popis komplexních systémů zahrnujících hardware, software, data i lidi. Pro procesní analýzu se pak široce využívá **BPMN** (Business Process Model and Notation) [7], jehož hlavní předností je srozumitelnost pro analytiky, vývojáře i koncové uživatele.

Pro práci se SysML a BPMN byla vyvinuta řada nástrojů – od komerčních systémů, jako je **Enterprise Architect** [34], až po otevřené nástroje typu **Archi** (pro **jazyk ArchiMate**) [35]. Výsledky procesního modelování agend veřejné správy jsou v ČR veřejně dostupné na stránkách **Digitální a informační agentury (DIA)** [36].

Metodika Business Object Relationship Modelling **(BORM)** [1–7] představuje strukturovaný přístup k analýze se zvláštním důrazem na propojené procesy a objekty. Kombinuje objektově orientovaný přístup (stavy a vztahy) s procesním modelováním (časová posloupnost a dynamické chování). Hlavní silnou stránkou BORM je velmi jednoduchá notace, která pomáhá i netechnicky zaměřeným stranám pochopit složitá schémata, jež se běžně používají pro zápis architektury.

BORM staví na přesných matematických základech (konečné automaty a Petriho sítě). Díky tomuto robustnímu formálnímu zázemí lze modely nejen vytvářet, ale také simulovat, verifikovat a validovat. Model funguje na principu komunikujících objektů. Změnu stavu spouští aktivita (proces), která může být podmíněna splněním kritérií nebo iniciována jiným objektem. Vzájemnou komunikací a předáváním dat je dosaženo názorného popisu složitých systémů.

Softwarový nástroj **Craft.CASE** [8] umožňuje tyto diagramy nejen vytvářet, ale také pomocí simulace testovat jejich chování. Lze tak včas zjistit, zda někde nedochází k tzv. **uváznutí** (deadlock), kdy proces marně čeká na signál z jiného objektu a zastaví se.

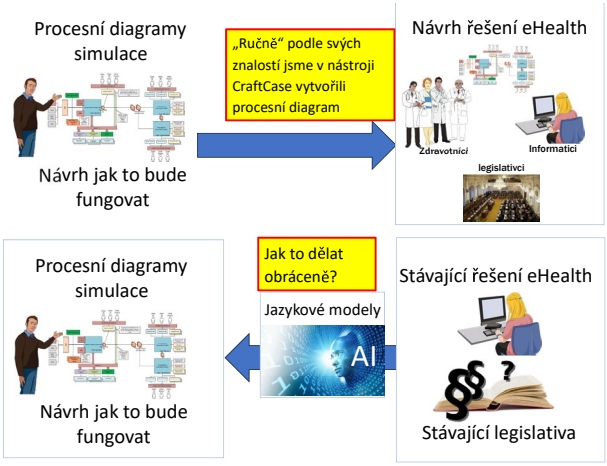
Na řešení problematiky ePreskripce jsme si v praxi ověřili, že metoda BORM a nástroj Craft.CASE zásadně usnadnily porozumění mezi zdravotníky, legislativci a vývojáři. To nám umožnilo v několika iteracích dospět ke konečnému řešení, podle něhož byla vypracována legislativa, aktualizován software i metodika práce uživatelů [10–14].

4 Objektově procesní metodologie (OPM)

Technologie BORM a vývojový nástroj Craft.CASE nám umožnily na základě procesní analýzy postupně vytvářet procesní model, zobrazovat jej v podobě diagramů a pomocí simulace ověřovat jeho chování.

Potřebný je však i opačný přístup: na základě stávajícího stavu – tedy textového popisu legislativy a normalizovaného popisu architektury softwarového systému – vytvořit, například s využitím generativních schopností velkých jazykových modelů (LLM) umělé inteligence (AI), procesní model (obr. 11), který by bylo možné dále modifikovat.

Zde se nabízí možnost využít **Objektově procesní metodologii (OPM – Object-Process Methodology)**, která slouží jako komplexní jazyk a metodika pro koncepční modelování systémů. OPM v mnoha ohledech pokrývá filozofii BORM a zároveň ji posouvá dál. OPM byla vyvinuta profesorem Dovem Dorim, přičemž počáteční myšlenky byly publikovány v roce 1995 v článku zabývajícím se problematikou konverze strojírenských



Obrázek 11 – Pomocí nástroje Craft.CASE jsme mohli pohodlně „ručně“ vytvářet procesní diagramy a simulací si ověřovat jejich chování. Nabízí se otázka, jak využít jazykové modely umělé inteligence k opačnému kroku – tedy jak ze stávající legislativy a popisu architektury automatizovaně vygenerovat procesní model.

výkresů do podoby 3D CAD modelů [37]. Počáteční aplikace OPM prokázaly její všestrannost v různých oblastech včetně počítačově integrované výroby, obrazové analýzy, transakcí v elektronickém obchodu a prostředí výzkumu a vývoje [38, 39].

První kniha o OPM, *Object-Process Methodology: A Holistic Systems Paradigm*, byla vydána nakladatelstvím Springer v roce 2002 [40]. O čtyři roky později následovala další učebnice [41], ve které Dov Dori teoreticky i prakticky na mnoha příkladech uvádí uplatnění této metodologie při návrhu a modelování komplexních systémů

Tyto publikace znamenaly významný krok ve formalizaci a šíření principů OPM. Původ této metodiky v řešení reálných inženýrských problémů, spíše než v čistě teoretické abstrakci, jí dává pragmatický a robustní základ. Rychlé uplatnění v různých oblastech ještě před její formální standardizací potvrzuje její inherentní užitečnost a přizpůsobivost. Vydání obou knih poskytlo ucelený a přístupný základ, který byl klíčový pro budování odborné komunity a usnadnil širší přijetí metodiky, jejíž mezinárodní ISO standard byl dokončen v roce 2024.

4.1 Holistický princip, správa složitosti a možnost simulace

OPM nabízí jednotný holistický (z řeckého holos – celek) pohled na systém jako celek – integruje strukturu, funkce a chování do jednoho soudržného modelu. Dokáže v jediném uceleném modelu zachytit jak strukturu (objekty), tak i chování (procesy) systému a integruje tyto dva pohledy do jednoho uceleného **objektově-procesního diagramu** (**OPD** – Object-Process Diagram).

Pomocí tohoto diagramu můžeme v nástroji implementujícím OPM ozřejmit chování modelovaného systému pomocí simulace. Složitost je řízena pomocí hierarchické abstrakce a zpřesňování. Pro zachycení komplexity celého systému se používají hierarchicky provázané objektově-procesní diagramy (OPD) – od jednoho kořenového diagramu až po detailní úrovně na nižších hierarchických úrovních.

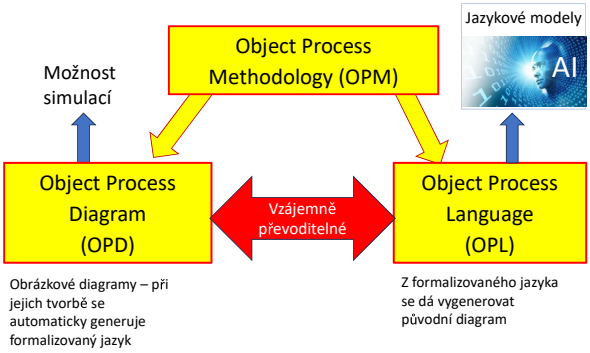
Namísto několika specializovaných typů diagramů, jako v jiných modelovacích jazycích (např. 9 až 14 typů v UML a SysML), se zde používá pouze jeden typ diagramu, který je hierarchicky provázán. Na rozdíl od jiných přístupů tak OPM poskytuje jediný, plnohodnotný pohled, což zlepšuje komunikaci mezi zúčastněnými stranami a poskytuje komplexní obraz o závislostech.

OPM umožňuje automatizaci a simulaci a udržuje vyvážené zobrazení objektů i procesů, čímž se odlišuje od jiných modelovacích jazyků a rozšiřuje jejich možnosti.

4.2 Bimodální, grafická i textová, reprezentace

Podstatné je, že při vytváření **objektově-procesního diagramu (OPD)** se automaticky generuje textový popis v jednoduchém normalizovaném **objektově-procesním jazyce (OPL – Object--Process Language).** Oba popisy jsou vzájemně zaměnitelné: z textového popisu v jazyce OPL lze vygenerovat diagram OPD a naopak. Modelovaný systém je tak reprezentován dvěma ekvivalentními způsoby – vizuálně a textově. Tato dualita zajišťuje jak formální přesnost, tak intuitivní pochopení.

Zde se do budoucna nabízí možnost propojit textový popis OPL s velkými jazykovými modely (LLM). Díky oboustranné převoditelnosti mezi OPL a OPD bude možné metody AI propojit s exaktními simulacemi v grafickém modelu [42] (obr. 12).



Obrázek 12 – Objektově-procesní metodologie (OPM) definuje grafické znázornění složitého systému pomocí hierarchicky organizovaných objektově-procesních diagramů (OPD). Nástroj implementující tuto metodologii (např. OPCloud) umožňuje prostřednictvím diagramů OPD simulovat chování modelovaného systému. Každý objektově-procesní diagram má svou textovou reprezentaci v tzv. objektově-procesním jazyce (OPL). Oba popisy systému (grafický OPD i textový OPL) jsou vzájemně zaměnitelné. Tato dualita otevírá do budoucna možnost propojit textovou podobu OPL s velkými jazykovými modely AI a zároveň algoritmicky simulovat chování systému pomocí grafického modelu OPD.

OPM v sobě spojuje aspekty ‚jazyka‘ a ‚metodiky‘. Jako jazyk poskytuje formální syntaxi a sémantiku pro modelování, což zajišťuje přesnost a jednoznačnost popisu systému. Jako metodika nabízí systematický způsob, jak tento jazyk aplikovat při vývoji a analýze. Tato kombinace řeší otázku **CO** (systém řeší) i **JAK** (jak to systém dělá), čímž se OPM stává komplexním nástrojem pro praxi. Schopnost integrovat různé disciplíny a reprezentovat veškeré důležité interakce činí tuto metodiku široce použitelnou v mnoha různých oblastech [43–46].

4.3 Mezinárodní standardizace

Význam OPM podtrhuje nedávná standardizace v podobě mezinárodní normy ISO 19450:2024 [47,48]. Česká agentura pro standardizaci již připravuje českou verzi této mezinárodní normy, která má být hotová a publikovaná v únoru 2026 a na jejíž české lokalizaci se osobně podílíme.

Standardizace OPM normou ISO zvyšuje její důvěryhodnost a podporuje širší přijetí, zejména v legislativně regulovaných odvětvích, kde jsou formální specifikace prvořadé. Pro návrh architektury systémů to přináší následující výhody:

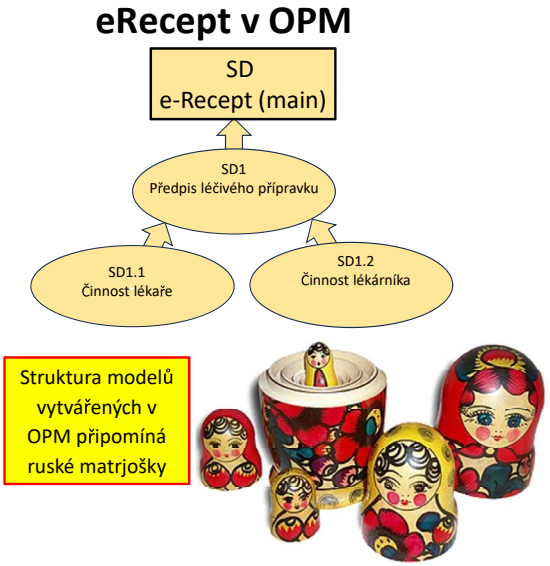
- Zlepšení porozumění a komunikace: Graficko-textová (bimodální) reprezentace zpřístupňuje komplexní modely technickým i netechnickým zúčastněným stranám, neboť vyhovuje různým způsobům myšlení (kognitivním stylům). Výsledkem je včasné zapojení aktérů, rychlejší sběr požadavků a snazší dosažení shody nad vytvářeným návrhem.
- Snížení nejednoznačnosti a zvýšení přesnosti: Formální syntaxe OPM, minimální ontologie a přísné definice minimalizují nejednoznačnost, což vede k jasnějším a přesnějším specifikacím systému.
- Přesná analýza a návrh: Integrovaný pohled na funkci, strukturu a chování v jediném modelu poskytuje koherentní referenční rámec, čímž zvyšuje přesnost analýzy i návrhu.
- Řízení složitosti: Mechanismy zpřesnění a abstrakce OPM umožňují modelářům řídit velké množství detailů, předcházet přeplněnosti diagramů a udržovat jasnost napříč různými úrovněmi abstrakce.

Vytvoření mezinárodního standardu OPM umožňuje sjednotit nástroje pro tvorbu a využívání této technologie, a to zejména pro návrh a analýzu komplexních systémů, kam patří mimo jiné i zdravotnické informační systémy [46].

4.4 Příklad – elektronický recept řešený pomocí OPM

Podklady pro přípravu návrhu legislativy a softwarového řešení eReceptu jsme původně připravovali prostřednictvím metodologie BORM v nástroji Craft.CASE [10–14], protože pro OPM v té době ještě nebyl žádný vývojový nástroj ani standard. Jako ilustraci možností využití Objektově-procesní metodologie (OPM) pro návrh architektury si na závěr ukážeme příklad v prostředí OPCloud. Jedná se o softwarové řešení izraelské firmy OPCloud Ltd., určené pro tvorbu modelů podle standardu ISO 19450:2024 (*www.opcloud.tech*).

Struktura modelů vytvářených v OPM připomíná ruské matrjošky (obr. 13). Na nejvyšší úrovni pracujeme s jedním základním procesem „Předpis léčivého přípravku“, se kterým je propojeno šest objektů. V OPM pracujeme s entitami (things), což mohou být buď objekty, nebo procesy, a s vazbami (links) mezi nimi.



Obrázek 13 – Hierarchická struktura modelů vytvářených v OPM – při modelování eReceptu se pohybujeme ve třech hierarchických úrovních.

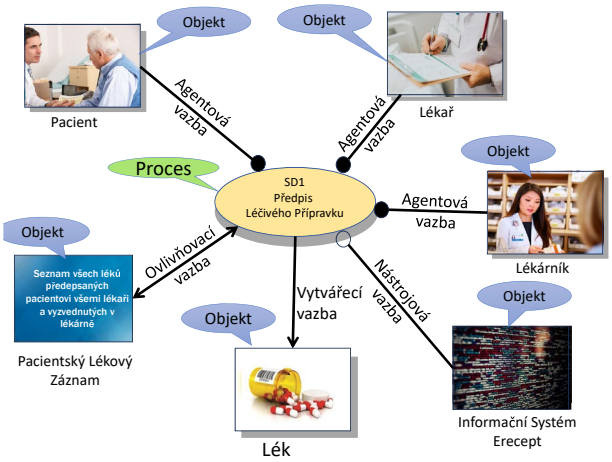
Objekty jsou entity, které existují, vznikají nebo zanikají po celou uvažovanou dobu; v diagramu jsou označeny obdélníkem. Rozlišuje se, zda jsou hmotné (tangible) nebo nehmotné. Hmotné objekty (např. Pacient, Lékař, Lékárník, Lék) mají za obdélníkem stín. Informační systém eRecept nebo Pacientský lékový záznam jsou objekty nehmotné.

Procesy jsou vždy dočasné a přechodné, označují se elipsou. Stejně jako objekty se mohou odehrávat ve hmotném světě (např. proces ‚Předpis léčivého přípravku‘ má podstíněnou elipsu), nebo jsou nehmotné, probíhající v informačním světě (např. vyhledávání údaje v databázi).

Mezi objekty a procesy existují vazby vyjadřující jejich vztahy. Ty jsou buď strukturální, nebo procedurální. V tomto článku se

soustředíme na **procedurální vazby**, které znázorňují **interakci mezi objekty a procesy**:

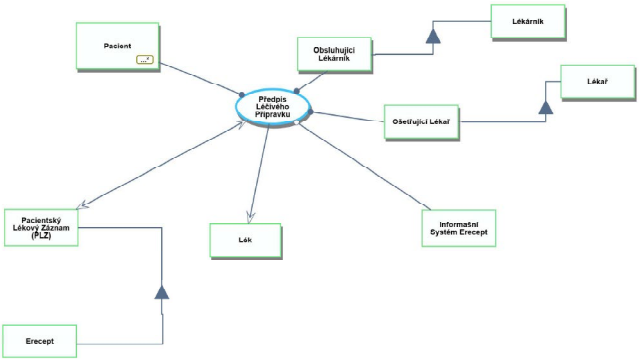
1. **Transformační vazby:** Vyjadřují, že objekty mohou procesy vytvářet, spotřebovávat nebo měnit jejich stav.
   a. Proces „Předpis léčivého přípravku“ (obr. 14) vytváří nový objekt „Lék“ pomocí **vytvářecí vazby** *(result link)*, zobrazené šipkou od procesu k objektu.
   b. Proces „Činnost lékaře“ vytváří objekt „eRecept“, který je spotřebován procesem „Vyzvednutí léku v lékárně“ (obr. 16) pomocí **spotřebovací vazby** *(consumption link)*.
   c. Změna stavu objektu (např. „Pacientský lékový záznam“) se zobrazuje **ovlivňovací vazbou** *(effect link)* v podobě obousměrné šipky.
2. **Umožňující vazby** *(enabling links)*: Vyjadřují, že proces ke své činnosti vyžaduje objekt, který jej aktivuje, ale sám jím není ovlivněn (např. „Lékař“, „Pacient“ či „IS eRecept“). Tyto vazby mají na straně objektu malé kolečko:
   a. Agentová vazba: Pokud je umožňujícím objektem živá osoba („Lékař“, „Pacient“), kolečko je plné.
   b. Nástrojová vazba *(instrument link)*: Pokud jde o neživý nástroj („IS eRecept“), kolečko je prázdné (viz obr. 14).



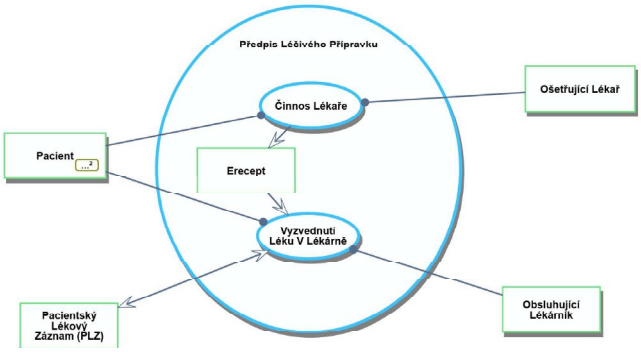
Obrázek 14 – Obrázkové schéma nejvyšší úrovně modelu eReceptu. Proces „Předpis Léčivého Přípravku“ je propojen s pěti objekty.

Procedurální vazby mohou vycházet i od jednotlivých stavů objektu. Například u objektu „Pacient“ (obr. 18) mohou nastat dva stavy: buď si lék vyzvedne sám (tím aktivuje proces „Vyzvednutí léku“), nebo vyzvednutím pověří jinou osobu.

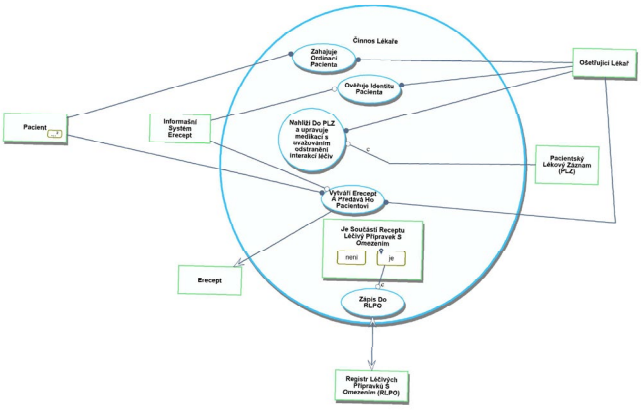
Na obr. 15 – obr. 18 jsou objektově-procesní diagramy, znázorňující zjednodušený model architektury eReceptu v OPM. Pro jednoduchost jsou zde vynechány procesy definující uživatelské nastavování stavů některých objektů při modelování.



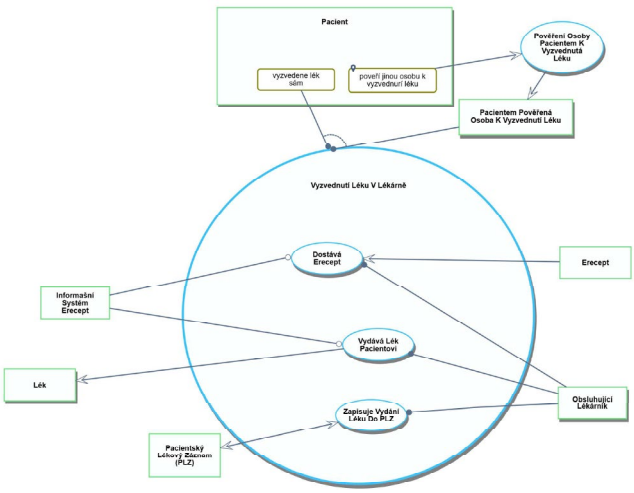
Obrázek 15 – Objektově procesní diagram (OPD) nejvyšší úrovně vytvořený v nástroji OPCloud



Obrázek 16 – OPD procesu „Předpis Léčivého Přípravku“ v sobě obsahuje dva procesy „Činnost Lékaře“ a „Vyzvednutí Léku v Lékárně“. První proces vytváří objekt „Erecept“, který je spotřebován procesem „Vyzvednutí Léku V Lékárně“. Tento proces transformační vazbou ovlivňuje objekt „Pacientský Lékový Záznam“.



Obrázek 17 – OPD procesu „Činnost lékaře“



Obrázek 18 – OPD procesu „Vyzvednutí léku v lékárně“

DOI: 10.35191/medsoft_2025_1_37_vojtech

Model Name: Předpis lékového přípravku

SD
1. Pacient is stateful.
2. Pacientský Lékový Záznam consists of Erecept.
3. Lékař consists of Ošetřující Lékař.
4. Lékárník consists of Obsluhující Lékárník.
5. Obsluhující Lékárník,Ošetřující Lékař, and Pacient handle Předpis Léčivého Přípravku.
6. Předpis Léčivého Přípravku requires Informační Systém Erecept.
7. Předpis Léčivého Přípravku affects Pacientský Lékový Záznam.
8. Předpis Léčivého Přípravku yields Lék.

SD1: Předpis Léčivého Přípravku in-zoomed
1. Předpis Léčivého Přípravku from SD zooms in SD1 into Činnos Lékaře, and Vyzvednutí Léku V Lékárně, which occur in that time sequence, as well as Erecept.
2. Pacient is stateful.
3. Ošetřující Lékař and Pacient handle Činnos Lékaře.
4. Činnos Lékaře yields Erecept.
5. Obsluhující Lékárník and Pacient handle Vyzvednutí Léku V Lékárně.
6. Vyzvednutí Léku V Lékárně affects Pacientský Lékový Záznam.
7. Vyzvednutí Léku V Lékárně consumes Erecept.

Obrázek 19 – Pro ilustraci podoby jazyka OPL, uvádíme jazykovou podobu objektově procesních diagramů (OPD) zobrazených na obrázcích 15 a 16

Složitost modelu je v OPM řízena pomocí hierarchické abstrakce. Celý model je provázán od výchozího diagramu (úroveň SD) až po detailní úrovně (SD 1.1, SD 1.2). Každému diagramu (OPD) odpovídá textová reprezentace v jazyce **OPL** (*Object-Process Language*), viz ukázka na obr. 19.

Chování modelu lze otestovat **simulací**. Na obr. 20 je naznačeno, jak simulace v nástroji OPCloud probíhá. Reálný průběh simulace je k dispozici na videu: https://www.youtube.com/watch?v=Hd9aS6tJraw

4.5 OPM jako validační metavrstva proti halucinacím velkých jazykových modelů (LLM)

LLM dávají „lidsky znějící“ textové popisy procesů, ale bez ukotvení v nějakém člověkem kontrolovaném modelu mohou ztrácet skutečné porozumění. Tak např. Klievtsova a spol, [49] ukazují, že LLM jsou schopny produkovat popisy procesů podobné lidským na základě předdefinovaných vzorů (např. procesní modely v jazyce v BPMN), ale vytvářené popisy zjevně postrádají skutečné pochopení procesních modelů, a pro jejich využití je nutný hybridní přístup zahrnující práci lidského analytika.

A právě OPM dnes přináší dva přístupy, které se výborně doplňují (viz obr. 12):

- **Obrovská výpočetní kapacita LLM** využitelná jako jazykový extraktor, který nám umožňuje analyzovat haldy textového materiálu a shrnout znalosti do nově vytvářeného textu.
- **OPM jako normalizátor, korektor a validátor** shrnující text do normalizované formy standardizovaného jazyka OPL, jenž je propojen s grafickým systémem hierarchicky uspořádaných (a potenciálně simulovatelných) diagramů OPD.

Perspektivně by dobře nastavený model LLM využívající OPM by v budoucnu měl být schopen z textu (zápisů z workshopů, z metodiky, popisu stávajících procesů a legislativy, i z dalších textových zdrojů.) rychle vytěžit potřebné znalosti a převést je do přesné syntaxe OPL výrazů jako:

- **objekty** (např. Pacient, eRecept, Očkovací záznam, Souhlas);
- **stavy objektů** (např. eRecept: vystaven / zneplatněn / vydán; Souhlas: udělen / odmítnut);
- **procesy** (např. Předepsání léku, Výdej léku, Očkování, Záznam očkování, Podepsání souhlasu);
- a příslušné **strukturální a procedurální vazby** mezi nimi.

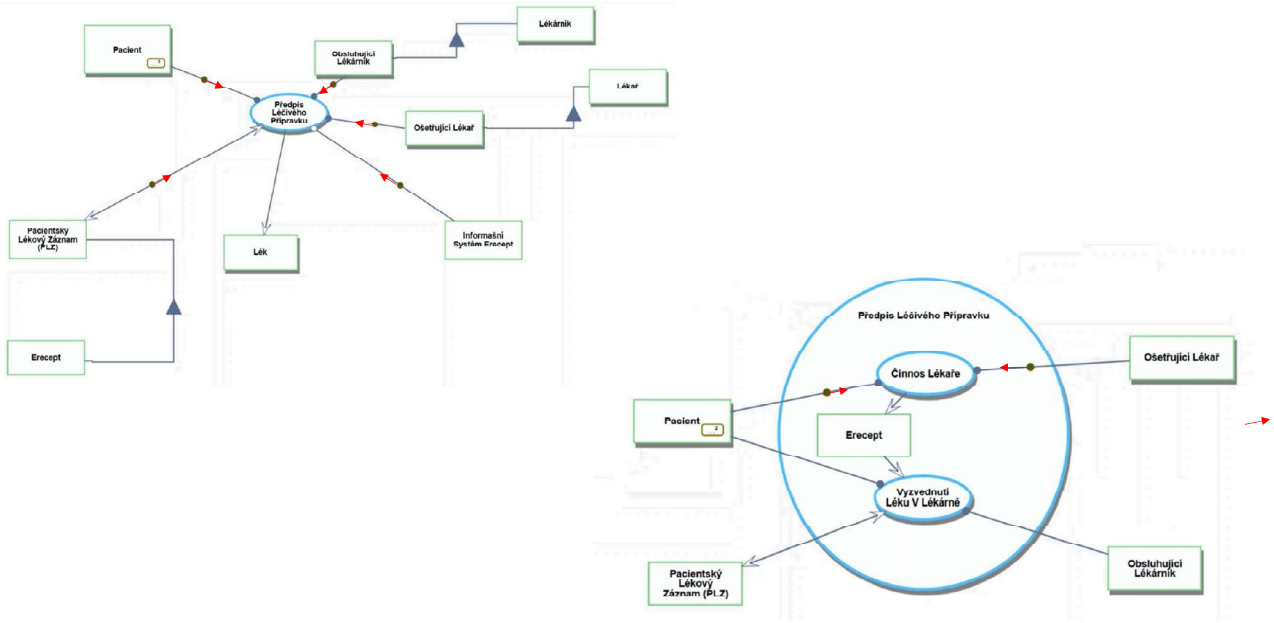
Síla LLM modelů se dá v OPM propojit s validací pomocí simulace v modelovacím nástroji. Vytvořené OPL výrazy potom analytik ihned vidí v diagramu OPD, spustí simulaci a model verifikuje a validuje.

LLM jsou náchylné k halucinacím, protože mohou generovat věrohodně znějící tvrzení bez záruky pravdivosti. Takové výstupy z LLM, které považujeme za halucinace, jsou:

1. **Logicky neřešitelné konstrukce** známé jako Russellův paradox [50]
2. **Syntakticky sice správné, ale sémanticky prázdné věty** známé jako Wittgensteinovský „nonsense“ [51]

LLM mají obrovskou výhodu v tom, že mohou rychle navrhovat, ale samy sebe nemohou kontrolovat. OPM + lidský analytik k nim však přidávají filtr, který snižuje riziko, že halucinace projdou do specifikace, což je v teoretické rovině v souladu s Tarskiho závěrem, že sémantickou pravdivost nelze spolehlivě uzavírat uvnitř téhož jazykového systému bez přidaných vrstev metajazyka [52]

OPM, který je právě takovou přidanou metavrstvou k jazykovým modelům, dovoluje model vygenerovaný od LLM v grafické podobě OPD pro lidského analytika spustit, simulovat a ověřit. To je v návrhu zdravotnických informačních systémů extrémně užitečné, protože takto dokážeme odhalit:



Obrázek 20 –Simulace chování objektově-procesního modelu eReceptu v prostředí OPCloud.Animace znázorňuje dynamiku systému: symbolické „kuličky“ (tokeny) se pohybují mezi objekty a procesy, přičemž se postupně otevírají aktivované diagramy na nižších úrovních hierarchie. To umožňuje detailně pozorovat a validovat chování modelu v reálném čase.

- **časový nesmysl** (např. „souhlas je udělen až po provedení výkonu"),
- **chybějící aktér** (proces se „děje sám"),
- **neexistující objekt** (LLM si „vymyslí" artefakt),
- nebo **skrytý konflikt stavů** (objekt je současně v neslučitelných stavech).

LLM je vhodné používat jako generátory kandidátních vět v OPL, ale nikoli jako autoritu rozhodující o správnosti. Praxe ukazuje, že LLM sice umí vytvářet věrohodné lidské popisy, ale nedokážou plně chápat procesní strukturu a konzistenci modelu. Výstupy je proto nutné vždy porovnat se strukturou OPM a ověřit lidským analytikem. Simulace v OPM modelu pak může sloužit jako povinná „protihalucinační brána" před implementací systému, protože právě simulace funguje jako efektivní detektor chyb a validátor.

To také odpovídá zkušenostem kolegů z Technické univerzity v Mnichově (TUM), kteří v rámci svého benchmarku porovnávali výstupy stejného LLM při práci jen se zadáním v přirozeném jazyku versus při doplnění OPM modelu ve formě OPL do LLM. Ukázali, že tato integrace díky vnesené ontologii a formální struktuře vede k vyšší kvalitě generovaných výstupů a k výraznému omezení halucinací technických specifikací a požadavků na subsystémy kosmických misí [53]

5 Závěr

Modelování procesů hraje klíčovou roli při překlenování komunikačních mezer a rozdílných pohledů mezi zdravotnickými profesionály, informatiky a legislativci. Komplexnost zdravotnické legislativy a odlišné odborné zázemí těchto skupin často vedou k bariérám v porozumění a k nejednotným interpretacím právních norem. Použití nástrojů pro formalizovaný popis složitých procesů a jejich grafická vizualizace tyto bariéry účinně odstraňuje. Simulace navíc umožňuje názorně otestovat chování navržené architektury a včas odhalit chyby v návrhu.

Na příkladu vytváření eReceptu v české legislativě jsme se v minulosti přesvědčili, že formalizace návrhu softwarové i legislativní architektury pomocí metodologie BORM a nástroje Craft. CASE podstatně zkrátila celý legislativní proces přípravy zákona.

Nová standardizovaná **Objektově-procesní metodologie (OPM)** lokalizovaná do českého jazyka a moderní nástroje, které tuto technologii implementují, tyto možnosti dále rozšiřují:

- **Vytvoření společného standardizovaného jazyka:** OPM poskytuje vizuální notaci srozumitelnou pro zdravotníky, technické specialisty i tvůrce legislativy. Tento společný jazyk pro popis procesů a právních požadavků předchází komunikačním mezerám a podporuje sdílené porozumění.
- **Vizualizace komplexních interakcí:** Modelování umožňuje graficky reprezentovat chování vytvářené procesní architektury, což pomáhá zúčastněným stranám lépe pochopit operace a identifikovat oblasti pro zlepšení.
- **Překonávání rozdílů v chápání legislativy:** Modelování pomáhá transformovat právní expertízu do akceschopných znalostí pro neprávníky a naopak, čímž podporuje transdisciplinární přístup k právu ve veřejném zdraví. Model architektury je stejně pochopitelný pro IT profesionály jako pro legislativce či odborníky z praxe.
- **Podpora mezioborové spolupráce:** Zlepšení komunikace je zásadní pro efektivní spolupráci, která je nezbytná pro poskytování bezpečné péče. Modelování identifikuje klíčové jevy a vztahy, což vede k optimalizaci procesů. Multidisciplinární porozumění je klíčem k dalšímu rozvoji zdravotnických informačních systémů.
- **Zajištění konzistence a snížení chyb:** Díky jasné vizualizaci lze snížit riziko nejednoznačností v legislativním návrhu. To vede k robustnější legislativě, která je lépe přizpůsobena realitě zdravotnické praxe.
- **Využití umělé inteligence:** Obousměrná vazba mezi grafickou podobou (OPD – umožňující simulaci) a textovou podobou (OPL – usnadňující napojení na velké jazykové modely) otevírá nové možnosti, zejména v automatizovaném propojování stávající a nově vytvářené legislativy.

Formalizace a modelování legislativních procesů postupně posouvá tvorbu práva od intuitivního ‚umění' založeného na zkušenostech k exaktnímu pojetí, které se opírá o algoritmy a moderní modelovací nástroje.

"FROM ART TO ALGORITHM" – FORMALIZATION AND MODELING OF LEGISLATIVE PROCESSES WITH A FOCUS ON HEALTHCARE

Abstract

The article deals with the formalization and modeling of legislative processes in healthcare. It underscores the importance of cooperation among architects, implementers, and users of information systems from the very beginning of their development. It draws attention to the pitfalls of creating information systems for public administration, requiring the embedding of an accurate description of the processes used in legislation. Using the example of our own experience with solving the issue of ePrescription and creating legislation for the digitization of vaccination records, it shows the importance of using methods and visual simulation tools to facilitate mutual understanding between healthcare professionals, legislators, and software developers, for early detection of errors and for significantly shortening the legislative process. The article also discusses current methods of formalizing legislative creation. It describes the promising, internationally standardized object process methodology (OPM) and our experience with the OPCloud tool, which implements it. Finally, recommendations are formulated for future strategy and the implementation of formalization approaches that facilitate interdisciplinary collaboration and the development of hybrid approaches combining formal process analysis methods with AI language models.

Keywords

AI, eHealth, Formalization, Modeling, Legislation, LLM, OPM, Software

Literatura

[1.] Knott RP, Merunka V, Polak J. Process modeling for object oriented analysis using BORM Object Behavioral Analysis. Proceedings Fourth International Conference on Requirements Engineering ICRE 2000 (Cat No98TB100219). IEEE; 2000. pp. 7–16. doi:10.1109/ICRE.2000.855566

[2.] Knott R, Merunka V, Polak J. The BORM methodology: a third-generation fully object-oriented methodology. Knowledge-Based Systems. 2003;16: 77–89. doi:10.1016/S0950–7051(02)00075–8

[3.] Merunka V, Brozek J, Sebek M, Polak J. BORM-business object relation modeling. 2009. Available: https://aisel.aisnet.org/amcis2009/788/

[4.] Brožek J, Merunka V, Merunková I. Organization modeling and simulation using BORM approach. Lecture Notes in Business Information Processing. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. pp. 27–40. doi:10.1007/978-3-642-15723-3_3

[5.] Molhanec M, Merunka V. BORM: Agile Modelling for Business Intelligence. Business Intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications. IGI Global; 2012. pp. 120–131. doi:10.4018/978-1-61350-050-7.ch006

[6.] Merunka V, Pergl R, Tůma J. BORM-II and UML as accessibility process in knowledge and business modelling. Lecture Notes in Electrical Engineering. Cham: Springer International Publishing; 2015. pp. 1–6. doi:10.1007/978-3-319-06764-3_1

[7.] Pavlicek J, Rod M, Pavlickova P. Usability evaluation of business process modeling standards – BPMN and BORM case study. Lecture Notes in Business Information Processing. Cham: Springer International Publishing; 2021. pp. 93–104. doi:10.1007/978-3-030-79022-6_9

[8.] Craft Case. Business Process Analysis. In: Visualize, define and simulate your processes. Locate inefficiencies, performance deviations or errors was never easier. [Internet]. Available: https://craftcase.com/

[9.] Berger J. Method and system for automated request modelling. US Patent. 7904431, 2011. Available: https://patentimages.storage.googleapis.com/4f/3b/91/efd8c98d224638/US7904431.pdf

[10.] Kofránek J, Berger J, Polák J, Vojtěch A. Modelování ehealth procesů pomocí hierarchických stavových automatů (statechartse. Medsoft. 2018;30: 35–56. Available: http://www.creativeconnections.cz/medsoft/2018/Medsoft_2018_Kofranek1.pdf

[11.] Kofránek J, Berger J, Štěpánek P, Vrubel F, Vojtěch A. Modelování procesů e-preskripce. Medsoft. 2019;31: 57–65. Available: https://www.medsoft.website/sbornik/2019/Medsoft_2019_Kofranek_Print.pdf

[12.] Bruthans J, Kofránek J, Vojtěch A. Concept and practice of electronic prescription. Medsoft. 2021;33: 89–92. Available: https://scholar.google.com/citations?view_op=view_citation&hl=en&citation_for_view=Qv8pd7MAAAAJ:0N-VGjzr574C

[13.] Berger J, Bruthans J, Vojtěch A, Kofránek J. Using process model to define the legislative framework of electronic prescription in the Czech Republic. Health Informatics J. 2024;30: 1–12. doi:10.1177/14604582241270902

[14.] Berger J, Bruthans J, Kofránek J. Improving Implementation of an Electronic Prescription System for COVID-19 Vaccination in the Czech Republic: Process Modeling Approach. JMIR Formative Research. 2023;7: e41575. Available: https://scholar.google.com/citations?view_op=view_citation&hl=en&citation_for_view=Qv8pd7MAAAAJ:_axFR9aDTf0C

[15.] Atkinson K. 2.1 – representation of legal information from A - information representation, preprocessing, and document assembly. In: Katz DM, Dolin R, Bommarito M, editors. Legal informatics. Cambridge University Press; 2021. pp. 35–40. doi:10.1017/9781316529683.004

[16.] Bench-Capon T, Araszkiewicz M, Ashley K, Atkinson K, Bex F, Borges F, et al. A history of AI and Law in 50 papers: 25 years of the international conference on AI and Law. Artif Intell Law. 2012;20: 215–319. doi:10.1007/s10506-012-9131-x

[17.] Michálek J. Modelování právních norem pomocí počítačových programů. 2021. Available: https://dspace.cuni.cz/handle/20.500.11956/170531

[18.] Šavelka J, Myška M, Ptašnik A, Spáčilová D. Právní informační systémy. Brno: Tribun EU; 2011. Available: https://scholar.google.com/citations?user=02Dvk88AAAAJ&hl=en&oi=sra

[19.] Waddington M. Rules as code: Drawing out the logic of legislation for drafters and computers. SSRN Electron J. 2022. doi:10.2139/ssrn.4299375

[20.] Xscential. How to Construct RFPs and Tenders for Legislative Modernization. In: LegisPro Standards® [Internet]. Available: https://xcential.com/legis-standards

[21.] Lucke JV, Fitsilis F, Etscheid J. Using artificial intelligence for legislation – thinking about and selecting realistic topics. EGOV-CeDEM-ePart-*. 2022. Available: https://dgsociety.org/wp-content/uploads/2022/09/CEUR-proceedings-2022.pdf#page=46

[22.] Hugo A. López, Søren Debois, Tijs Slaats & Thomas T. Hildebrandt. Business Process Compliance Using Reference Models of Law. 1st ed. In: Wehrheim H, Cabot J, editors. Lecture Notes in Computer Science. 1st ed. Cham, Switzerland: Springer Nature; 2020. doi:10.1007/978-3-030-45234-6_19

[23.] Epifanova TV, Vovchenko NG, Toporov DA, Pozdnyshov AN. Development of legal education and machine-readable law in the conditions of economy digitization. Digital Economy: Complexity and Variety vs Rationality. Cham: Springer International Publishing; 2020. pp. 971–979. doi:10.1007/978-3-030-29586-8_110

[24.] Flatt A, Langner A, Leps O. Model-driven development of akoma ntoso application profiles: A conceptual framework for model-based generation of XML subschemas. 1st ed. Cham, Switzerland: Springer International Publishing; 2023. doi:10.1007/978-3-031-14132-4

[25.] OASIS. In: Akoma Ntoso Version 1.0 – OASIS Open [Internet]. 2018. Available: https://www.oasis-open.org/standard/akn-v1–0/

[26.] MinBZK/leos: Legislation Editing Open Software – GitHub. Available: https://github.com/MinBZK/leos

[27.] Yahya M, Breslin JG, Ali MI. Semantic Web and Knowledge Graphs for Industry 4.0. Appl Sci (Basel). 2021;11: 5110. doi:10.3390/app11115110

[28.] Pan JZ. Resource Description Framework. Handbook on Ontologies. Berlin, Heidelberg: Springer Berlin Heidelberg; 2009. pp. 71–90. doi:10.1007/978-3-540-92673-3_3

[29.] Antoniou G, van Harmelen F. Web ontology language: OWL. Handbook on Ontologies. Berlin, Heidelberg: Springer Berlin Heidelberg; 2009. pp. 91–110. doi:10.1007/978-3-540-92673-3_4

[30.] Boella G, Caro LD, Humphreys L, Robaldo L, Rossi P, van der Torre L. Eunomos, a legal document and knowledge management system for the Web to provide relevant, reliable and up-to-date information on the law. Artif Intell Law. 2016;24: 245–283. doi:10.1007/s10506-016-9184-3

[31.] Cvrček F. Právnický elektronický slovník (PES). Právník Roč. 2015;154: 247–260. Available: https://www.ilaw.cas.cz/upload/web/files/pravnik/issues/2015/3/4.INF._Cvrcek_3_2015.pdf

[32.] Unhelkar B. Software engineering with UML. London, England: CRC Press; 2020. doi:10.1201/9781351235181

[33.] Friedenthal S, Moore A, Steiner R. A practical guide to SysML: the systems modeling language. Morgan Kaufmann; 2014. Available: https://www.academia.edu/download/73088918/c158c95430a561305e3e050e9ee74011e55b.pdf

[34.] Sparxs System. In: Enterprise Architect [Internet]. Available: https://sparxsystems.com/

[35.] Archi. In: The Open Source modelling toolkit for creating ArchiMate models and sketches. Used by Enterprise Architects everywhere. [Internet]. Available: https://www.archimatetool.com/

[36.] Procesní modelování agend veřejné správy. In: Digitální a Informační Agentura [Internet]. Available: https://www.dia.gov.cz/cs/nase-cinnosti/projekty/projekty-fondy-eu/procesni-modelovani-agend-verejne-spravy

[37.] Dori D, Tombre K. From engineering drawings to 3D cad models: are we ready now? Comput Aided Des. 1995;27: 243–254. doi:10.1016/0010–4485(95)91134–7

[38.] Dori D. Documenting system specifications through opm web-based graphics/text equivalence: the case of the free flight system. J Logic Comput. 1995;5: 227–249. Available: https://dovdori.technion.ac.il/wp-content/uploads/2022/04/DocumentingSystemSpecificationsThroughOPM.pdf

[39.] Dori D. Document analysis systems development and representation through the object-process methodology. Document Analysis Systems: Theory and Practice. Berlin, Heidelberg: Springer Berlin Heidelberg; 1999. pp. 271–282. doi:10.1007/3-540-48172-9_22

[40.] Dori D. Object-process methodology: A holistic systems paradigm. Berlin, Heidelberg: Springer Science & Business Media; 2002. doi:10.1007/978-3-642-56209-9

[41.] Dori D. Model-based systems engineering with OPM and SysML. 1st ed. New York, NY: Springer; 2016. doi:10.1007/978-1-4939-3295-5

[42.] Kang X, Shteingardt V, Wang Y, Dori D. Neuro-Conceptual Artificial Intelligence: Integrating OPM with deep learning to enhance question answering quality. arXiv [cs.CL]. 2025. Available: http://arxiv.org/abs/2502.09658

[43.] Dori D. Object-Process Methodology. Encyclopedia of Knowledge Management, Second Edition. IGI Global; 2010. pp. 1208–1220. doi:10.4018/978-1-59904-931-1.ch116

[44.] Dori D, Kohen H, Jbara A, Wengrowicz N, Lavi R, Soskin NL, et al. OPCloud : An OPM integrated conceptual-executable modeling environment for industry 4.0. Systems Engineering in the Fourth Industrial Revolution. Wiley; 2019. pp. 243–271. doi:10.1002/9781119513957.ch11

[45.] Casebolt JM, Jbara A, Dori D. Business process improvement using Object-Process Methodology. Syst Eng. 2020;23: 36–48. doi:10.1002/sys.21499

[46.] Vizitiu C, Dominey K, Nistorescu A, Dinculescu A, Marin M. MHealth case study presenting design SynMeth, a rapid prototyping MBSE methodology, by advancing specific OPM-to-SysML mapping. IEEE Access. 2025;13: 53531–53545. doi:10.1109/access.2025.3553945

[47.] Dori D. Model-based standards authoring: ISO 15288 as a case in point. Syst Eng. 2024;27: 302–314. doi:10.1002/sys.21721

[48.] ISO (the International Organization for Standardization). Automation systems and integration – Object-Process Methodology. 2024 Jan. Report No.: ISO 19450:2024. Available: https://drive.google.com/file/d/1mgrRDQJ7BUsDPcGX6mdqWp8nnYLfXWcm/view?usp=sharing

[49.] Klievtsova N, Mangler J, Kampik T, Rinderle-Ma S. Utilizing process models in the requirements engineering process through Model2Text transformation. 2024 IEEE 32nd International Requirements Engineering Conference (RE). IEEE; 2024. pp. 205–217. doi:10.1109/re59067.2024.00028

[50.] Deutsch H., Marshall O., Irwine A. D. Russell's paradox. In: Zalta and Uri Nodelman E, editor. Stanford Encyclopedia of Philosophy (Winter 25 Edition). Standford: The Metaphysics Research Lab, Department of Philosophy, Stanford University, Stanford, CA 94305; 2025. Available: https://plato.stanford.edu/archives/win2024/entries/russell-paradox/

[51.] Biletzki A, Matar A. Ludwig wittgenstein. In: Zalta and Uri Nodelman E, editor. The Standford Encyclopedia of Philosophy Fall 23 Edidion. Standford: The Metaphysics Research Lab, Department of Philosophy, Stanford University, Stanford, CA 94305; 2023. Available: https://plato.stanford.edu/archives/fall2023/entries/wittgenstein/

[52.] Hodges W. Tarski's truth definitions. In: Zalta and Uri Nodelman E, editor. The Standford Encyclopedia of Philosophy Winter 22 Edidion. Standford: The Metaphysics Research Lab, Department of Philosophy, Stanford University, Stanford, CA 94305; 2022. Available: https://plato.stanford.edu/archives/win2022/entries/tarski-truth/

[53.] García Alarcia RM, Russo P, Renga A, Golkar A. Bringing systems engineering models to large language models: An integration of OPM with an LLM for design assistants. Proceedings of the 12th International Conference on Model-Based Software and Systems Engineering. SCITEPRESS – Science and Technology Publications; 2024. doi:10.5220/0012621900003645

Kontakt

Mgr. et Mgr. Adam Vojtěch
Laboratoř Biokybernetiky
Ústav patologické fyziologie 1. LF UK
U nemocnice 5
128 53 Praha 2
adam.vojtech@lf1.cuni.cz

doc MUDr. Jiří Kofránek, CSc
Laboratoř Biokybernetiky
Ústav patologické fyziologie 1. LF UK
U nemocnice 5
128 53 Praha 2
jiri.kofranek@lf1@cuni.cz
+420 777 686 868

doc. Ing. Vojtěch Merunka, Ph.D.
Katedra informačního inženýrství
Provozně ekonomická fakulta ČZU v Praze
Kamýcká 129
165 00 Praha-Suchdol
merunka@pef.czu.cz
https://vojtech.merunka.eu/
+420 603 802 220